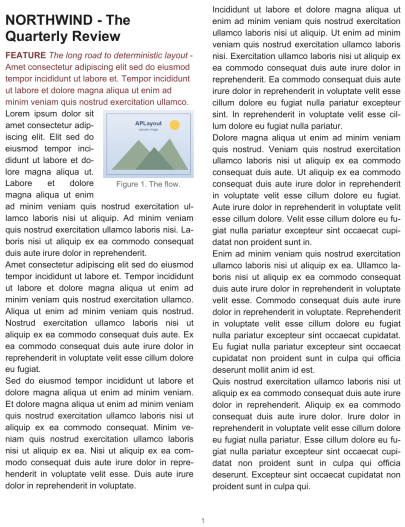
appligent
DOCUMENT SOLUTIONS



NORTHWIND - The Quarterly Review

FEATURE *The long road to deterministic layout* - Amet consectetur adipiscing elit sed do eiusmod tempor incididunt ut labore et. Tempor incididunt ut labore et dolore magna aliqua ut enim ad minim veniam quis nostrud exercitation ullamco.

Lorem ipsum dolor sit amet consectetur adipiscing elit. Elit sed do eiusmod tempor incididunt ut labore et dolore magna aliqua ut. Labore et dolore magna aliqua ut enim ad minim veniam quis nostrud exercitation ullamco laboris nisi ut aliquip. Ad minim veniam quis nostrud exercitation ullamco laboris nisi. Laboris nisi ut aliquip ex ea commodo consequat duis aute irure dolor in reprehenderit.

Figure 1. The flow.

Amet consectetur adipiscing elit sed do eiusmod tempor incididunt ut labore et. Tempor incididunt ut labore et dolore magna aliqua ut enim ad minim veniam quis nostrud exercitation ullamco. Aliqua ut enim ad minim veniam quis nostrud. Nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat duis aute. Ex ea commodo consequat duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat.

Sed do eiusmod tempor incididunt ut labore et dolore magna aliqua ut enim ad minim veniam. Et dolore magna aliqua ut enim ad minim veniam quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Minim veniam quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea. Nisi ut aliquip ex ea commodo consequat duis aute irure dolor in reprehenderit in voluptate velit esse. Duis aute irure dolor in reprehenderit in voluptate.

Incididunt ut labore et dolore magna aliqua ut enim ad minim veniam quis nostrud exercitation ullamco laboris nisi ut aliquip. Ut enim ad minim veniam quis nostrud exercitation ullamco laboris nisi. Exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat duis aute irure dolor in reprehenderit. Ea commodo consequat duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur excepteur sint. In reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.

Dolore magna aliqua ut enim ad minim veniam quis nostrud. Veniam quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat duis aute. Ut aliquip ex ea commodo consequat duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat. Aute irure dolor in reprehenderit in voluptate velit esse cillum dolore. Velit esse cillum dolore eu fugiat nulla pariatur excepteur sint occaecat cupidatat non proident sunt in.

Enim ad minim veniam quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea. Ullamco laboris nisi ut aliquip ex ea commodo consequat duis aute irure dolor in reprehenderit in voluptate velit esse. Commodo consequat duis aute irure dolor in reprehenderit in voluptate. Reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur excepteur sint occaecat cupidatat. Eu fugiat nulla pariatur excepteur sint occaecat cupidatat non proident sunt in culpa qui officia deserunt mollit anim id est.

Quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat duis aute irure dolor in reprehenderit. Aliquip ex ea commodo consequat duis aute irure dolor. Irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Esse cillum dolore eu fugiat nulla pariatur excepteur sint occaecat cupidatat non proident sunt in culpa qui officia deserunt. Excepteur sint occaecat cupidatat non proident sunt in culpa qui.

1

PageWright™

JSON Spec Language Reference

The complete reference to the document description language.

Appligent, Inc.
22 East Baltimore Avenue
Lansdowne, PA 19050 U.S.A.

(610) 284-4006
sales@appligent.com
https://www.appligent.com

Version 1.0
October 2026

© 2026 Appligent, Inc. All rights reserved. PageWright is a trademark of Appligent, Inc.

About this manual

This is the **PageWright™ JSON Spec Language Reference**. PageWright is a PDF page-layout engine: you describe a document as a **JSON spec** - its pages, text, tables, images, charts, forms and structure - and the engine flows it onto pages and writes a tagged, standards-compliant PDF.

Every page of this manual was produced by the engine itself from a JSON spec (`sample/guide-spec-reference.json`), so it doubles as a worked example: the cover, the table of contents, the running heads and folios, the code boxes, the reference tables, and the index and glossary at the back are all ordinary engine features described in these pages.

> **Note**
>
> Conventions. Monospace text in a red tint marks a JSON key, value, or command - for example `"type": "paragraph"` or `pagewrightapp`. Shaded boxes are complete, valid snippets you can paste into a spec. **Bold** marks the first mention of a defined term; the Glossary collects them.

Contents

About this manual ii
Part I — Introduction 1
1. What a spec is 1
2. The shape of a document 1
3. How to read this reference 2
Part II — Keys every block accepts 4
4. Common block keys 4
Part III — The elements 7
5. Text 7
6. Structure and spacing 35
7. Tables 64
8. Pictures and drawings 74
9. Placing existing PDFs 122
10. Navigation and apparatus 134
11. Notes and asides 158
12. Forms 164
13. Data-driven documents 170
14. Story threading 185
15. Meters 191
Part IV — Shared vocabularies 197
16. Shared definitions 197
Part V — The document object 223
17. Document-level keys 223
Appendices 235
A. Every element at a glance 235
B. Enumerations 236
Glossary 241
Index 242

Part I — Introduction

1. What a spec is

A PageWright document is a single **JSON** file. It is not a program and there is no scripting: you describe *what the document contains* - its pages, text, tables, pictures, charts, forms and structure - and the engine decides where everything lands, flows it onto pages, and writes a tagged, standards-compliant PDF.

That distinction matters. You never position a line of text or ask for a page break in the middle of a paragraph. You say "here is a heading, here are four paragraphs, here is a table", and the engine paginates, wraps, justifies, hyphenates, keeps headings with their text, repeats table headers across page breaks, and numbers everything.

The smallest useful document is three lines:

```
{
  "blocks": [
    {
      "type": "paragraph",
      "text": "Hello."
    }
  ]
}
```

Render it from the command line:

```
pagewrightapp hello.json hello.pdf
```

2. The shape of a document

Every spec is a JSON **object**. At the top sit document-wide settings - metadata, page setup, default styling - and one of two content keys:

- `blocks` - a flat list of elements, for a document with one page setup throughout.
- `sections` - a list of sections, each with its own page setup and its own `blocks`, for a document that changes orientation, margins, column count or folio style partway through.

A document uses one or the other, never both. Everything else is optional.

```
{
  "title": "Quarterly Report",
  "author": "Appligent, Inc.",
  "tagged": true,
  "lang": "en-US",
  "pageSetup": {
    "size": "Letter",
    "margins": 64,
    "footer": {
      "center": "{{page}}"
    }
  },
  "defaultStyle": {
    "fontFamily": "Helvetica",
    "fontSize": 11
  },
  "blocks": [
    {
      "type": "heading",
      "level": 1,
      "text": "Summary"
    },
    {
      "type": "paragraph",
      "text": "Revenue rose nine per cent."
    }
  ]
}
```

Note

`tagged: true` is what makes the output an accessible PDF: headings become real heading structure, tables get row and cell structure, and pictures carry their alternate text. Set it on anything that will be read by a person rather than only by a machine.

3. How to read this reference

Every element gets the same treatment: what it is for, a worked example you can paste into a spec, and a table of its keys. In those tables:

- **Key** is the JSON property name, spelled exactly as the engine expects it.
- **Type** is `string`, `number`, `integer`, `boolean`, `object`, `array of X`, or the name of a shared definition documented in Part IV.
- **Default** is what the engine uses when you leave the key out. An empty cell means the key has no default - leaving it out simply means the feature is off.
- **Description** says what the key does. **Required** marks a key the element cannot be built without.

Keys that every block accepts - `style`, `id`, `lang`, `position` and the rest - are documented once, in their own chapter, and left out of the per-element tables so that what you see there is what is specific to that element.

> **Tip**
>
> This reference is generated from the engine's own JSON Schema, so the key tables cannot drift from the engine. The same schema ships with the engine and can validate your specs directly.

Part II — Keys every block accepts

4. Common block keys

The keys below may appear on **any** element. They are listed here once and omitted from the individual element tables.

Key	Type	Default	Description
`align`	`string`		Horizontal alignment of the block's content within its column. One of: `left`, `center`, `right`, `justify`, `start`, `end`.
`fitWidth`	`boolean`		Heading: solve the font size so the (single-line) text fills the column.
`id`	`string`		Cross-reference anchor: a run's "ref" renders this block's page number and links here.
`keepWithNext`	`boolean`		Pagination: keep this block on the same page as the start of the next block (avoids a stranded line).
`lang`	`string`		BCP-47 language tag for this block, emitted as /Lang so a screen reader switches pronunciation. It also decides how the block's words divide when `hyphenate` is on: by this language's rules rather than the document's, and not at all when the engine embeds no patterns for it.
`letterSpacing`	`number`		Extra points between glyphs (inherited).
`position`	`position`		Place the block at fixed coordinates instead of in the flow, measured from the trim page's top-left -- or, when the nearest enclosing container that writes "childOrigin" says "container", from that container's box (see childOrigin). A positioned block reserves NO vertical space in the flow unless it also sets "textWrap".
`style`	`style`		Style for this block. Text properties cascade to nested content; box properties (padding, border, spacing) apply to this block only.
`textWrap`	`string`		Text wrap around an absolutely-positioned block (ignored without "position"). "none" (default): the block leaves the flow entirely and reserves no space -- what a letterhead, background panel or form-field overlay needs. "topAndBottom": the flow keeps clear of the block's whole vertical band, edge to edge, so flow content AFTER it in document order drops below it. "around": the flow keeps clear only of the block's OWN RECTANGLE, so a panel narrower than its column gets text beside it and full width below -- use this rather than topAndBottom when a pinned block should not split the page across. A block pinned strictly INSIDE its column (touching neither edge) leaves the strip on one side unused; pin it to a column edge to use both. (Named textWrap because `pre` already owns a boolean `wrap`.) One of: `none`, `topAndBottom`, `around`.
`tracking`	`number`		Alias for letterSpacing.
`unless`	`string`		Data-driven conditional: drop this block when the predicate is true.
`when`	`string`		Data-driven conditional: keep this block only when the predicate is true (a path, or "path OP value").
`wordSpacing`	`number`		Extra points between words. A BLOCK property: it is read from the block being laid out, so setting it on a RUN parses and resolves but changes nothing. Inherited by nested blocks.

Note

`when` and `unless` are the data-driven conditionals: give either one a predicate and the block survives into the document only when the predicate holds. They are the per-block form of the `if` element.

Part III — The elements

5. Text

Elements that put words on the page.

The paragraph element

Type name: `paragraph`.
The workhorse. A paragraph is a run of prose that the engine wraps, justifies and paginates for you: give it `text` for a single unstyled string, or `runs` when parts of it need their own styling, a link, a footnote or a cross-reference. Whitespace is collapsed - every run of spaces, tabs and newlines becomes one space - so you can format the JSON however you like without changing the output. A paragraph splits across pages and columns at line boundaries, with widow and orphan control.

Example

```
{
  "type": "paragraph",
  "runs": [
    {
      "text": "Tighten the retaining bolt to "
    },
    {
      "text": "18 N·m",
      "style": {
        "weight": "bold"
      }
    },
    {
      "text": " before flight."
    }
  ]
}
```

Figures aligned on a decimal tab, with a dot leader

```
{
  "type": "paragraph",
  "style": {
    "tabs": [
      {
        "at": 60,
        "align": "left"
      },
      {
        "at": 300,
        "align": "decimal",
        "leader": "dot"
      }
    ]
  },
  "text": "Retaining bolt\t18.00"
}
```

Keys

Key	Type	Default	Description
`anchor`	`anchor`		-
`angle`	`number | string`		Turn the block, in degrees counter-clockwise, about the CENTRE of its "position" box. A CONTAINER in the flow may also carry this: it becomes a GROUP that turns about its own measured box and reserves the bounding box of the turned result, and its children inherit the transform. Any other flow block ignores it with a warning -- wrap it in a container. The block lays out upright inside its box and the finished result is turned, so a "position.height" that truncates content truncates the same text it would unrotated, and the turned result is NOT clipped to its box: it may extend past it, exactly as an unrotated positioned block may. Not to be confused with the image block's "rotate", which is an integer QUARTER TURN CLOCKWISE applied to the picture INSIDE its box (and re-sizes that box to the turned orientation). The two compose -- an image can stand a sideways scan upright with "rotate" and tilt the whole frame with "angle". Same meaning as the watermark's "angle". A STRING is a data token, so each record of a template turns its own sticker: "{{s.angle}}" takes the record's number (or text of one, such as a CSV field); a record without it leaves the block upright, silently, and a value that is not a number is reported with the block's pointer (and the -merge record) and drawn upright. A string with no token is read as it stands.
`bleed`	`number`		Step and Repeat: how far this placed block's box reaches past its TRIM on every side, in points -- the box IS the bleed box, and the trim is the box inset by this much. A design tool sizes a card by it (the trim plus twice the bleed) and draws the trim line inside each copy; the engine draws nothing from it, and keeps the bleed only through the block's own `clip`. Only a block with `position` has a trim: on any other block the key is refused with a notice naming the block.

Key	Type	Default	Description
`blend`	`string`		How this block's OWN ink combines with the page under it -- the sixteen PDF blend modes (ISO 32000-1 11.3.5), written as /BM. On a container that means its background, border and stripe, never its children; for the subtree as one object see "groupBlend". Case and separators are folded, so "colorDodge", "color-dodge" and "COLORDODGE" are one mode. An unrecognized name is refused with a warning and the block paints normally. See the blend-modes sample. One of: `normal`, `multiply`, `screen`, `overlay`, `darken`, `lighten`, `colorDodge`, `colorBurn`, `hardLight`, `softLight`, `difference`, `exclusion`, `hue`, `saturation`, ...
`boxHeight`	`number`		Vertical writing only: the height in points of the box the columns are set in. Omit to fill the height the paragraph is offered: in a horizontal page from where it stands down to the page foot (at most the text area's height); in a table cell, a `columns` cell, a cardGrid card or a container down to the page foot too, so a row holding it stands a page tall; in a positioned paragraph or container its position.height (without one, the room from where the text begins down to the flow's foot, above any footnote band - or, beginning in the bottom margin or within a character of that foot, down to the page's foot - at most the text area's full height, as a flowing paragraph's; it took the text area's full height from where the box was placed, running past the page foot, until 2026-10-15); in a cell of a positioned `columns` block, table or cardGrid with no height - or a container holding one - every such text in it (a list item's too) shares the box's room: each is given the same box, the tallest (at most the text area's height) with which the whole box ends at that foot - or, where the box's horizontal content runs past the foot however short that box is, with which every such text does - every row still drawn, a card grid's rows still uniform, and columns past a cell's width cut and said, naming the box they are set in (it took the text area's full height there, running past the page foot, until 2026-10-15). Not bounded, as before: such text in a positioned box with a position.height (a table drops its rows past that height, said; a `columns` block's or cardGrid's stands a text area tall past it), or positioned in a vertical section - set boxHeight there. The foot is read when the box is placed: a footnote that its own cells, or a block placed after it, add on that page is not above it. Set it for a smaller box, which a vertical block in a cell, a card or a container must do. A box shorter than one character is raised to one (a column holds at least one), and the engine says so. When the box does not fit in the room left the paragraph moves to the next page (a fresh page shrinks the box to the room there). Columns that overrun the box's width continue on the next page; in a table cell, a `columns` cell, a cardGrid card or a positioned box, which do not continue, they are not drawn, and the engine says how many characters were lost and that the columns ran past the width (a taller boxHeight, a wider box or shorter text holds them).
`breakBefore`	`boolean`		Pagination: open a new page before this block - a part divider, an appendix. Ignored when the page is already empty, so it never leaves a blank page behind.

Key	Type	Default	Description
`changed`	`boolean`		Revision / change bar: draw a vertical bar in the page margin beside this block's content to flag it as changed. A document-level changeBars object customizes the bar (width / color / side / distance); a condition tag declared with changeBar:true sets this automatically for tagged content.
`clip`	`boolean`		Clip this block's children (on a container) or its own ink to its box, with the style's corner "radius". Only bites where content is BIGGER than the box: a flow container grows to fit its children vertically, so give it "position" with a height, or a "minHeight", before expecting a vertical crop. Container-scoped, like "groupOpacity": on any other flow block it is refused with a warning. For a crop to a SHAPE rather than to the box, see "clipPath"; giving both INTERSECTS them.
`clipPath`	`object`		Crop this block to an arbitrary SHAPE stated inside its box -- a design tool's FRAME. Same vocabulary as the "shape" block and read by the same outline builder, so a shape and a clipPath given identical parameters describe the identical outline. Scoped exactly as "clip" is: a positioned block of any type (it needs a "position.height", because the shape is stated inside the box) or a container in the flow; anywhere else it is refused with a warning. Naming "clip" as well INTERSECTS the two. The shape turns, scales and skews with the block that carries it.
`columnGap`	`number`		Vertical writing only: extra space in points between columns, on top of the one-em column body. Omit for the conventional half-em gap.
`condition`	`string`		Conditional text: render this block only when its condition-tag expression is shown under the document's conditions map. Whitespace/comma tag terms, each optionally '!'-negated, AND-combined; an unlisted tag defaults to shown. Empty means always shown.
`counter`	`string | object`		Named autonumber stream (defined in numbering.counters): bump it and prefix this paragraph its value ("Step 1. "). A string names the stream; the object form restarts it.
`counterRestart`	`boolean`		Restart the named counter stream before this block (shorthand alongside a string "counter").
`dropCap`	`integer`		Drop cap: set the first character as an enlarged initial dropped over this many lines; the body wraps around it.
`float`	`string`		Float this block out of the stacked flow to one side of its column; content that FOLLOWS it wraps into the space beside it. Works on ANY block type -- a table, a container, a chart, a code listing, a figure. "center" straddles the page content width, so text in every column wraps around it. The float box takes the block's own "width"; a block that declares none takes HALF the column, so a float always leaves room beside it. Ignored on a block that also sets "position" or "anchor" -- those have already chosen their placement. Not to be confused with "anchor": { "float": true }, a BOOLEAN about migrating an anchored frame to the next column that can hold it. One of: `left`, `right`, `center`.

Key	Type	Default	Description
`groupBlend`	`string`		Composite this block AND its children once, then blend the RESULT with the page. NOT the same key as "blend", and the difference is larger than it is for opacity: two stacked children that each multiply the backdrop multiply it TWICE, which is darker than the group multiplying once, and no per-child setting reproduces it. That is what a design tool's blend-mode control on a group means. Container-scoped: on any other flow block it is refused with a warning. Also the way to blend an element together with its drop shadow, which is drawn as stacked alpha layers and so keeps its own normal compositing. See the blend-modes sample. One of: `normal`, `multiply`, `screen`, `overlay`, `darken`, `lighten`, `colorDodge`, `colorBurn`, `hardLight`, `softLight`, `difference`, `exclusion`, `hue`, `saturation`, ...
`groupOpacity`	`number`		Composite this block AND its children once, then fade the result -- true GROUP opacity. NOT the same key as "opacity", which fades each element as it paints: two overlapping opaque children at opacity 0.5 each show a seam where they cross, and under groupOpacity 0.5 they do not. The two compose. Container-scoped: on any other flow block it is refused with a warning. See the clipped-groups sample.
`minFontSize`	`number`	6	shrinkToFit's floor, in points: shrinking never sets any run of the text below this, and text written smaller keeps its own size.
`mirrorH`	`boolean | string`		Mirror the whole block left-to-right about the centre of its "position" box, under the same restriction as "angle". Identical to scale x -1, and the two compose. NOT "flipH": that key belongs to the image block, where it mirrors the PICTURE INSIDE its frame. The two are separate words so an image can do both. A STRING is a data token, so each record of a template mirrors its own sticker: "{{s.flip}}" takes the record's boolean, or text "true" / "false" in any case, or 1 / 0 (a number or text); a record without it draws the block unmirrored, silently, and any other value ("yes" included) is reported with the block's pointer (and the -merge record) and drawn unmirrored. A string with no token is read as it stands.
`mirrorV`	`boolean | string`		Mirror the whole block top-to-bottom about the centre of its "position" box, under the same restriction as "angle". Identical to scale y -1. NOT "flipV" -- see "mirrorH". A STRING is a data token, so each record of a template mirrors its own sticker: "{{s.flip}}" takes the record's boolean, or text "true" / "false" in any case, or 1 / 0 (a number or text); a record without it draws the block unmirrored, silently, and any other value ("yes" included) is reported with the block's pointer (and the -merge record) and drawn unmirrored. A string with no token is read as it stands.

Key	Type	Default	Description
`opacity`	`number`		Constant opacity for this block: 0 (invisible) to 1 (opaque, the default). REAL alpha -- a /CA /ca ExtGState in the PDF -- so a translucent panel over a photograph shows the photograph through it. Not to be confused with "style.background" {color, alpha}, which composites the tint toward WHITE at parse time and so only looks right over a white page. PER-ELEMENT, NOT GROUP: on a container it paints that container's own decoration -- background, border, accent bar, stripe -- and never its children, which draw at their own opacity (1 unless they set one). Two translucent shapes that overlap therefore composite twice, which is what per-element opacity means; group opacity, where a whole subtree is composited once and then faded, is a separate feature. Where a draw already carries its own alpha (an SVG fill-opacity, a gradient's alpha) the two MULTIPLY.
`repeatAt`	`array of object`		Imposition: draw this block ONCE and place the drawing again at each of these offsets, in points from the block's own position, y DOWNWARD as every `position` is. One entry = one extra copy, so a list of 7 puts 8 cards on a sheet. The drawing is captured into a Form XObject and each offset is a `cm` + `Do` of that one stream, so n copies cost one copy of the content. Translation only -- `angle` / `scale` / `skew` are not applied per placement. A REPEATED FORM MAY NOT CARRY TAGGED CONTENT: ISO 14289-1 7.20 allows a form's content ONE semantic parent, so repeating tagged text (or a shape with `alt`) fails PDF/UA -- veraPDF catches it, ap_addstruct's audit does not. The engine warns and draws it anyway. Repeat decoration: a shape or svg with no `alt`, or put the block in a master's `frames`, which is drawn as a page artifact. Only a block with `position` is drawn again: on a block without one (in the flow, or an anchored frame), or on a repeat / if / include block, the key is refused with a notice naming the block. Each copy takes the block's `groupOpacity` and `groupBlend`. The layout map (-map) reports the copies as `repeats` on the block's fragment, so a design tool can select and outline them.
`runs`	`runs`		The block's text as styled inline runs, instead of "text". A run can carry a link, footnote, badge, inline maths or a cross-reference.
`scale`	`number | array of number | object | string`		Scale the block about the same pivot as "angle" (the centre of its "position" box), and under the same restriction. A CONTAINER in the flow may also carry this: it becomes a GROUP that turns about its own measured box and reserves the bounding box of the turned result, and its children inherit the transform. Any other flow block ignores it with a warning -- wrap it in a container. A number scales both axes; [x, y] or {"x":, "y":} scales them separately. 1 = natural size. Applied AFTER layout, so a scaled block does NOT re-wrap its text at the scaled measure -- this is a design tool's scale handle, not a change of font size. A STRING is a data token, so each record of a template scales its own sticker: "{{s.scale}}" takes the record's number, [x, y] or {"x":, "y":} (members may be text of numbers), or text of one or two plain numbers such as a CSV field ("1.5", "1.5 0.8", "[1.5, 0.8]"); a record without it draws the block at its natural size, silently, and a value that is not a number or a pair is reported with the block's pointer (and the -merge record) and drawn at natural size. A string with no token is read as it stands.

Key	Type	Default	Description
`shrinkToFit`	`boolean`		With a position height: shrink the text until its lines fit the box, in 0.25 pt steps of its largest size -- every size scaled by one ratio, but no run set below minFontSize (a run written smaller keeps its own size). Solved against the box's own height, so the size is the same wherever the page breaks. A block whose own keys ask for it without a position height is told so and keeps its size; a dropCap, textPath or vertical paragraph, or one a run-in heading is drawn into, is not shrunk. Text that still does not fit at minFontSize, or that its padding leaves no room for, is clipped and reported. May also come from the block's style.
`skew`	`number | array of number | object | string`		Shear the block, in DEGREES, about the same pivot as "angle", under the same restriction. [x, y] or {"x":, "y":}; a bare number skews x only. Positive x leans the top to the right and positive y lifts the right-hand side -- both counter-clockwise, matching "angle". Clamped to +/-89.9: a 90-degree skew is a singular matrix and draws nothing. A STRING is a data token, so each record of a template skews its own sticker: "{{s.skew}}" takes the record's number (x only), [x, y] or {"x":, "y":}, or text of one or two plain numbers; clamped as the written one is. A record without it draws the block unskewed, silently, and a value that is not a number or a pair is reported with the block's pointer (and the -merge record) and drawn unskewed. A string with no token is read as it stands.
`straddle`	`boolean`		Straddle across columns: when true and the body is multi-column, this block spans the FULL content width and text in every column wraps around it (above and below) - a spanning float generalized from center-floated images to any block. No effect in a single-column body, on a positioned / anchored block, or on an image/pdf already using float:"center" (the same mechanism). Does not balance the columns above it or act as a section divider.
`text`	`string`		The block's text, when it needs no inline styling.
`textPath`	`textPath`		-
`verticalAlign`	`string`		Where the content sits in its "position" box when it is shorter than the box's height: "top" (the default), "middle" ("center" is the same) or "bottom". A paragraph, heading or pre moves as a whole (its own box with it); a container keeps its box where it is and moves its children inside it (a child with its own "position" stays put). The offset is the box height less the content's height as it is actually placed -- the same lines, shrinkToFit size and padding that are then drawn -- halved for middle. Content as tall as the box or taller does not move: it is clipped and reported exactly as without the key. Needs a position with a height: on a block without one, on any other kind of block, on text set along a textPath "d" curve (stated in fractions of the whole box, so drawn across all of it; the "arc" form aligns), and on vertical text whose columns fill the box (columns run the full height unless a paragraph's or a heading's boxHeight is shorter), it is refused with a notice and ignored. The layout map keeps reporting the box. One of: `top`, `middle`, `bottom`, `center`.

Key	Type	Default	Description
`writingMode`	`string`		Writing mode for this paragraph. "vertical-rl" sets it in 直排 (tategaki) columns - glyphs advance downward, columns advance right to left; "vertical-lr" runs the columns left to right. The block owns a fixed box inside the ordinary horizontal flow, so the surrounding layout is unchanged. Omit (or "inherit") to take the section's pageSetup.writingMode, itself horizontal by default. Vertical text needs an embedded font (pageSetup.embedFont or a style fontFamily) - the base-14 faces carry no vertical CMap and no Han glyphs. One of: `horizontal-tb`, `horizontal`, `lr-tb`, `yokogaki`, 横排, `vertical-rl`, `vertical`, `tb-rl`, `tategaki`, 直排, 縦書き, `vertical-lr`, `tb-lr`, `inherit`.
`z`	`number`		CANVAS MODE ONLY (see the top-level "pages"): where this element sits in the PAINT order of its page. Lower draws first, further back; unset means "wherever I am in the list", and ties keep author order. It moves the PAINT and not the READING: the structure tree keeps the document order, so a caption written first because it is read first can still be painted last because it sits on top. That separation is what a flattened scene graph cannot express. Ignored outside canvas mode, where the flow already fixes both orders.

The heading element

Type name: `heading`.

A titled break in the flow, at `level` 1 to 6. Headings do more than look large: each one is collected into the table of contents, becomes a PDF bookmark, and is tagged `H1`-`H6` for accessibility. Give a heading an `id` and other blocks can cross-reference it. With document numbering enabled the engine writes the number for you, so `"1.2 Assembly"` is something you never type.

Example

```
{
  "type": "heading",
  "level": 2,
  "id": "assembly",
  "text": "Assembly"
}
```

An appendix heading, lettered by document numbering

```
{
  "type": "heading",
  "level": 1,
  "appendix": true,
  "id": "app-data",
  "text": "Data tables"
}
```

Keys

Key	Type	Default	Description
`anchor`	`anchor`		-
`angle`	`number` | `string`		Turn the block, in degrees counter-clockwise, about the CENTRE of its "position" box. A CONTAINER in the flow may also carry this: it becomes a GROUP that turns about its own measured box and reserves the bounding box of the turned result, and its children inherit the transform. Any other flow block ignores it with a warning -- wrap it in a container. The block lays out upright inside its box and the finished result is turned, so a "position.height" that truncates content truncates the same text it would unrotated, and the turned result is NOT clipped to its box: it may extend past it, exactly as an unrotated positioned block may. Not to be confused with the image block's "rotate", which is an integer QUARTER TURN CLOCKWISE applied to the picture INSIDE its box (and re-sizes that box to the turned orientation). The two compose -- an image can stand a sideways scan upright with "rotate" and tilt the whole frame with "angle". Same meaning as the watermark's "angle". A STRING is a data token, so each record of a template turns its own sticker: "{{s.angle}}" takes the record's number (or text of one, such as a CSV field); a record without it leaves the block upright, silently, and a value that is not a number is reported with the block's pointer (and the -merge record) and drawn upright. A string with no token is read as it stands.
`appendix`	`boolean`	false	On a level-1 heading, opens the APPENDICES: the chapter counter restarts and switches to letters from here on, so this becomes A, the next B. Chapter-relative captions follow ("Figure A-1"), and a cross-reference reads "Appendix A".
`bleed`	`number`		Step and Repeat: how far this placed block's box reaches past its TRIM on every side, in points -- the box IS the bleed box, and the trim is the box inset by this much. A design tool sizes a card by it (the trim plus twice the bleed) and draws the trim line inside each copy; the engine draws nothing from it, and keeps the bleed only through the block's own `clip`. Only a block with `position` has a trim: on any other block the key is refused with a notice naming the block.
`blend`	`string`		How this block's OWN ink combines with the page under it -- the sixteen PDF blend modes (ISO 32000-1 11.3.5), written as /BM. On a container that means its background, border and stripe, never its children; for the subtree as one object see "groupBlend". Case and separators are folded, so "colorDodge", "color-dodge" and "COLORDODGE" are one mode. An unrecognized name is refused with a warning and the block paints normally. See the blend-modes sample. One of: `normal`, `multiply`, `screen`, `overlay`, `darken`, `lighten`, `colorDodge`, `colorBurn`, `hardLight`, `softLight`, `difference`, `exclusion`, `hue`, `saturation`, ...

Key	Type	Default	Description
`boxHeight`	`number`		Vertical writing only (a heading whose writingMode, its own or its section's, is vertical): the height in points of the box its columns are set in - a paragraph's boxHeight, same meaning and default (since rev 2026-10-14.2). Set it for a bounded heading: in a table cell, a `columns` cell, a cardGrid card or a container the row or box is sized to it rather than standing a page tall, and in a horizontal page the block after it follows below it, so keepWithNext can be met. A box shorter than one character is raised to one, and the engine says so. When the box does not fit in the room left the heading moves to the next page (a fresh page shrinks the box to the room there), and a heading before it with keepWithNext moves with it, whether or not this one keeps with its own next. When its columns are wider than the width left it is split where it stands in a page's column or a positioned box (since 2026-10-15: the next column or page is no wider, so a box of the same height is just as wide there; moving it only split it there and left a heading before it alone), the rest continuing on the next page, still the heading (<Hn>, listed once) - a heading before it with keepWithNext keeps with its first part, and its own keepWithNext holds its last part to the block after it; where the next slot may be wider - a vertical section, a box narrowed by its padding or beside a float, a story's frame - it moves on whole and is split only there if it must be. In a table cell, a `columns` cell, a cardGrid card or a positioned box, which do not continue, columns past the width are not drawn (its <Hn> says what is drawn; its bookmark and contents row keep the title), and the engine says how many characters were lost and that the columns ran past the width. Omit to fill the height the heading is offered, as a vertical paragraph without boxHeight does: in a horizontal page from where it stands down to the page foot (at most the text area's height) - and since a heading keeps with its next block by default and nothing can follow it on that page, it moves to the next page unless it already stands at a page top, its body after it on the page after (keepWithNext: false keeps it where it stands; a vertical paragraph given keepWithNext moves the same way); in a table cell, a `columns` cell, a cardGrid card or a container down to the page foot, a row holding it a page tall - and in a cell of a positioned `columns` block, table or cardGrid with no height, its share of that box's room, as a vertical paragraph's (since 2026-10-15; not in a box with a position.height, nor in a vertical section).
`breakBefore`	`boolean`		Pagination: open a new page before this block - a part divider, an appendix. Ignored when the page is already empty, so it never leaves a blank page behind.
`changed`	`boolean`		Revision / change bar: draw a vertical bar in the page margin beside this block's content to flag it as changed. A document-level changeBars object customizes the bar (width / color / side / distance); a condition tag declared with changeBar:true sets this automatically for tagged content.
`clip`	`boolean`		Clip this block's children (on a container) or its own ink to its box, with the style's corner "radius". Only bites where content is BIGGER than the box: a flow container grows to fit its children vertically, so give it "position" with a height, or a "minHeight", before expecting a vertical crop. Container-scoped, like "groupOpacity": on any other flow block it is refused with a warning. For a crop to a SHAPE rather than to the box, see "clipPath"; giving both INTERSECTS them.

Key	Type	Default	Description
`clipPath`	`object`		Crop this block to an arbitrary SHAPE stated inside its box -- a design tool's FRAME. Same vocabulary as the "shape" block and read by the same outline builder, so a shape and a clipPath given identical parameters describe the identical outline. Scoped exactly as "clip" is: a positioned block of any type (it needs a "position.height", because the shape is stated inside the box) or a container in the flow; anywhere else it is refused with a warning. Naming "clip" as well INTERSECTS the two. The shape turns, scales and skews with the block that carries it.
`columnGap`	`number`		Vertical writing only: extra space in points between the heading's columns, on top of the one-em column body - a paragraph's columnGap, same meaning (since rev 2026-10-14.2). Omit for the conventional half-em gap, half the heading's own font size.
`condition`	`string`		Conditional text: render this block only when its condition-tag expression is shown under the document's conditions map. Whitespace/comma tag terms, each optionally '!'-negated, AND-combined; an unlisted tag defaults to shown. Empty means always shown.
`float`	`string`		Float this block out of the stacked flow to one side of its column; content that FOLLOWS it wraps into the space beside it. Works on ANY block type -- a table, a container, a chart, a code listing, a figure. "center" straddles the page content width, so text in every column wraps around it. The float box takes the block's own "width"; a block that declares none takes HALF the column, so a float always leaves room beside it. Ignored on a block that also sets "position" or "anchor" -- those have already chosen their placement. Not to be confused with "anchor": { "float": true }, a BOOLEAN about migrating an anchored frame to the next column that can hold it. One of: `left`, `right`, `center`.
`groupBlend`	`string`		Composite this block AND its children once, then blend the RESULT with the page. NOT the same key as "blend", and the difference is larger than it is for opacity: two stacked children that each multiply the backdrop multiply it TWICE, which is darker than the group multiplying once, and no per-child setting reproduces it. That is what a design tool's blend-mode control on a group means. Container-scoped: on any other flow block it is refused with a warning. Also the way to blend an element together with its drop shadow, which is drawn as stacked alpha layers and so keeps its own normal compositing. See the blend-modes sample. One of: `normal`, `multiply`, `screen`, `overlay`, `darken`, `lighten`, `colorDodge`, `colorBurn`, `hardLight`, `softLight`, `difference`, `exclusion`, `hue`, `saturation`, ...
`groupOpacity`	`number`		Composite this block AND its children once, then fade the result -- true GROUP opacity. NOT the same key as "opacity", which fades each element as it paints: two overlapping opaque children at opacity 0.5 each show a seam where they cross, and under groupOpacity 0.5 they do not. The two compose. Container-scoped: on any other flow block it is refused with a warning. See the clipped-groups sample.

Key	Type	Default	Description
`level`	`integer`		Heading level, 1 (most prominent) to 6. Drives the tagged structure (H1-H6), the outline hierarchy and automatic heading numbering. THOSE THREE ARE SEPARABLE: `outline: false` keeps the H1-H6 tag while leaving the heading out of the bookmark panel and the contents list, and `numbered: false` drops it from the numbering stream. Use `outline: false` for a card title or panel label -- a heading to a reader, noise beside the document's real chapters -- rather than styling a paragraph bold, which produces a <P> and is announced as body text.
`minFontSize`	`number`	6	shrinkToFit's floor, in points: shrinking never sets any run of the text below this, and text written smaller keeps its own size.
`mirrorH`	`boolean | string`		Mirror the whole block left-to-right about the centre of its "position" box, under the same restriction as "angle". Identical to scale x -1, and the two compose. NOT "flipH": that key belongs to the image block, where it mirrors the PICTURE INSIDE its frame. The two are separate words so an image can do both. A STRING is a data token, so each record of a template mirrors its own sticker: "{{s.flip}}" takes the record's boolean, or text "true" / "false" in any case, or 1 / 0 (a number or text); a record without it draws the block unmirrored, silently, and any other value ("yes" included) is reported with the block's pointer (and the -merge record) and drawn unmirrored. A string with no token is read as it stands.
`mirrorV`	`boolean | string`		Mirror the whole block top-to-bottom about the centre of its "position" box, under the same restriction as "angle". Identical to scale y -1. NOT "flipV" -- see "mirrorH". A STRING is a data token, so each record of a template mirrors its own sticker: "{{s.flip}}" takes the record's boolean, or text "true" / "false" in any case, or 1 / 0 (a number or text); a record without it draws the block unmirrored, silently, and any other value ("yes" included) is reported with the block's pointer (and the -merge record) and drawn unmirrored. A string with no token is read as it stands.
`number`	`string`		Pin a literal number/label for this block. Does not advance the counter.
`numbered`	`boolean`	true	Take part in automatic heading numbering (when the document enables it). false leaves a title / front-matter heading unnumbered and does not advance the counter.
`opacity`	`number`		Constant opacity for this block: 0 (invisible) to 1 (opaque, the default). REAL alpha -- a /CA /ca ExtGState in the PDF -- so a translucent panel over a photograph shows the photograph through it. Not to be confused with "style.background" {color, alpha}, which composites the tint toward WHITE at parse time and so only looks right over a white page. PER-ELEMENT, NOT GROUP: on a container it paints that container's own decoration -- background, border, accent bar, stripe -- and never its children, which draw at their own opacity (1 unless they set one). Two translucent shapes that overlap therefore composite twice, which is what per-element opacity means; group opacity, where a whole subtree is composited once and then faded, is a separate feature. Where a draw already carries its own alpha (an SVG fill-opacity, a gradient's alpha) the two MULTIPLY.

Key	Type	Default	Description
`outline`	`boolean`	true	Whether this heading enters the document OUTLINE -- the bookmark panel and, because both are fed by the same registration, the table of contents and the running heads. false keeps everything else about the heading, including its <Hn> tagged structure, its numbering and its style. Intended for a heading inside a card, panel or callout: real structure for assistive technology, absent from navigation.
`repeatAt`	`array of object`		Imposition: draw this block ONCE and place the drawing again at each of these offsets, in points from the block's own position, y DOWNWARD as every `position` is. One entry = one extra copy, so a list of 7 puts 8 cards on a sheet. The drawing is captured into a Form XObject and each offset is a `cm` + `Do` of that one stream, so n copies cost one copy of the content. Translation only -- `angle` / `scale` / `skew` are not applied per placement. A REPEATED FORM MAY NOT CARRY TAGGED CONTENT: ISO 14289-1 7.20 allows a form's content ONE semantic parent, so repeating tagged text (or a shape with `alt`) fails PDF/UA -- veraPDF catches it, ap_addstruct's audit does not. The engine warns and draws it anyway. Repeat decoration: a shape or svg with no `alt`, or put the block in a master's `frames`, which is drawn as a page artifact. Only a block with `position` is drawn again: on a block without one (in the flow, or an anchored frame), or on a repeat / if / include block, the key is refused with a notice naming the block. Each copy takes the block's `groupOpacity` and `groupBlend`. The layout map (-map) reports the copies as `repeats` on the block's fragment, so a design tool can select and outline them.
`runIn`	`boolean`	false	Draw this heading INLINE at the start of the following paragraph's first line (a bold lead-in), the body flowing on the same line beside it. The heading stays a distinct <Hn> structure element (outline / TOC / bookmark). No effect unless a plain paragraph follows; falls back to a normal stacked heading under a baseline grid or when the next block is not a plain paragraph. Ignored on a heading set on a curve (textPath) or in vertical (直排) columns, which are shapes of their own: the heading is set on its curve or in its columns, on its own, and the engine says so once (at /pointer).
`runs`	`runs`		The block's text as styled inline runs, instead of "text". A run can carry a link, footnote, badge, inline maths or a cross-reference.

Key	Type	Default	Description
scale	number | array of number | object | string		Scale the block about the same pivot as "angle" (the centre of its "position" box), and under the same restriction. A CONTAINER in the flow may also carry this: it becomes a GROUP that turns about its own measured box and reserves the bounding box of the turned result, and its children inherit the transform. Any other flow block ignores it with a warning -- wrap it in a container. A number scales both axes; [x, y] or {"x":, "y":} scales them separately. 1 = natural size. Applied AFTER layout, so a scaled block does NOT re-wrap its text at the scaled measure -- this is a design tool's scale handle, not a change of font size. A STRING is a data token, so each record of a template scales its own sticker: "{{s.scale}}" takes the record's number, [x, y] or {"x":, "y":} (members may be text of numbers), or text of one or two plain numbers such as a CSV field ("1.5", "1.5 0.8", "[1.5, 0.8]"); a record without it draws the block at its natural size, silently, and a value that is not a number or a pair is reported with the block's pointer (and the -merge record) and drawn at natural size. A string with no token is read as it stands.
shrinkToFit	boolean		With a position height: shrink the text until its lines fit the box, in 0.25 pt steps of its largest size -- every size scaled by one ratio, but no run set below minFontSize (a run written smaller keeps its own size). Solved against the box's own height, so the size is the same wherever the page breaks. A block whose own keys ask for it without a position height is told so and keeps its size; a textPath or vertical heading, or a side head, is not shrunk. Text that still does not fit at minFontSize, or that its padding leaves no room for, is clipped and reported. May also come from the block's style.
sidehead	boolean	false	Draw this heading in the reserved side-head band (see pageSetup.sideHeadWidth) instead of the body flow, with its top aligned to the body that follows. No effect unless the section defines a side-head area; then the heading consumes no vertical flow and the body sits beside it.
skew	number | array of number | object | string		Shear the block, in DEGREES, about the same pivot as "angle", under the same restriction. [x, y] or {"x":, "y":}; a bare number skews x only. Positive x leans the top to the right and positive y lifts the right-hand side -- both counter-clockwise, matching "angle". Clamped to +/-89.9: a 90-degree skew is a singular matrix and draws nothing. A STRING is a data token, so each record of a template skews its own sticker: "{{s.skew}}" takes the record's number (x only), [x, y] or {"x":, "y":}, or text of one or two plain numbers; clamped as the written one is. A record without it draws the block unskewed, silently, and a value that is not a number or a pair is reported with the block's pointer (and the -merge record) and drawn unskewed. A string with no token is read as it stands.
straddle	boolean		Straddle across columns: when true and the body is multi-column, this block spans the FULL content width and text in every column wraps around it (above and below) - a spanning float generalized from center-floated images to any block. No effect in a single-column body, on a positioned / anchored block, or on an image/pdf already using float:"center" (the same mechanism). Does not balance the columns above it or act as a section divider.
text	string		The block's text, when it needs no inline styling.
textPath	textPath		-

Key	Type	Default	Description
`verticalAlign`	`string`		Where the content sits in its "position" box when it is shorter than the box's height: "top" (the default), "middle" ("center" is the same) or "bottom". A paragraph, heading or pre moves as a whole (its own box with it); a container keeps its box where it is and moves its children inside it (a child with its own "position" stays put). The offset is the box height less the content's height as it is actually placed -- the same lines, shrinkToFit size and padding that are then drawn -- halved for middle. Content as tall as the box or taller does not move: it is clipped and reported exactly as without the key. Needs a position with a height: on a block without one, on any other kind of block, on text set along a textPath "d" curve (stated in fractions of the whole box, so drawn across all of it; the "arc" form aligns), and on vertical text whose columns fill the box (columns run the full height unless a paragraph's or a heading's boxHeight is shorter), it is refused with a notice and ignored. The layout map keeps reporting the box. One of: `top`, `middle`, `bottom`, `center`.
`writingMode`	`string`		Writing mode for this heading. "vertical-rl" sets it in 直排 (tategaki) columns at the heading's own size - glyphs advance downward, columns advance right to left; "vertical-lr" runs the columns left to right. It stays the heading: tagged <Hn>, with one bookmark and one contents row. It moves to the next page when its box is taller than the room left, and when its columns are wider than the width left where moving can help - without a boxHeight (its box grows to a fresh page's height), in a vertical section (the width left shrinks as columns fill), in a box narrowed by its padding or beside a float, or in a story's frame (the next slot may be wider); a heading in its own boxHeight given a page's whole column, or in a positioned box, whose box comes back no wider in the next column or page, is split where it stands instead (since 2026-10-15), as is any heading at a page top. The rest continues as the heading (<Hn>, at its size, listed once), never as body text. Its columns are boxHeight tall and columnGap apart - the paragraph's two keys, with the same meaning and defaults (since rev 2026-10-14.2). Without boxHeight it fills the height it is offered, exactly as a vertical paragraph without one does: in a horizontal page from where it stands down to the page foot (at most the text area's height), and in a table cell, a `columns` cell or a container down to the page foot too, so a table or columns row holding it stands a page tall; with boxHeight it is bounded there, its row or box sized to it, and in a horizontal page its body follows below it. In a vertical section it is a column group in the page's own flow. Its endnotes and citations are drawn in the column and linked both ways; a footnote or a margin note has no page foot or margin band beside a column, so it is not drawn and takes no number, and the engine says so. runIn is ignored on a vertical heading - a column is a shape of its own - and the engine says so once (at /pointer). Omit (or "inherit") to take the section's pageSetup.writingMode, itself horizontal by default. Vertical text needs an embedded font (pageSetup.embedFont or a style fontFamily) - the base-14 faces carry no vertical CMap and no Han glyphs. One of: `horizontal-tb`, `horizontal`, `lr-tb`, `yokogaki`, `横排`, `vertical-rl`, `vertical`, `tb-rl`, `tategaki`, `直排`, `縦書き`, `vertical-lr`, `tb-lr`, `inherit`.

Key	Type	Default	Description
`z`	`number`		CANVAS MODE ONLY (see the top-level "pages"): where this element sits in the PAINT order of its page. Lower draws first, further back; unset means "wherever I am in the list", and ties keep author order. It moves the PAINT and not the READING: the structure tree keeps the document order, so a caption written first because it is read first can still be painted last because it sits on top. That separation is what a flattened scene graph cannot express. Ignored outside canvas mode, where the flow already fixes both orders.

The list element

Type name: `list`.

A bulleted or numbered list. Each item is itself a list of blocks, so an item can hold several paragraphs, a nested list, an image or a table - not just one line of text. Set `ordered` for numbers, and nest lists to whatever depth the material needs.

Example

```
{
  "type": "list",
  "ordered": true,
  "items": [
    {
      "blocks": [
        {
          "type": "paragraph",
          "text": "Seat the motor."
        }
      ]
    },
    {
      "blocks": [
        {
          "type": "paragraph",
          "text": "Attach the shock cord."
        }
      ]
    }
  ]
}
```

A multi-level numbered outline (1, 1.1, 1.1.1)

```
{
  "type": "list",
  "ordered": true,
  "multiLevel": true,
  "items": [
    {
      "blocks": [
        {
          "type": "paragraph",
          "text": "Preparation"
        }
      ],
      "items": [
        {
          "blocks": [
            {
              "type": "paragraph",
              "text": "Inspect the airframe."
            }
          ]
        },
        {
          "blocks": [
            {
              "type": "paragraph",
              "text": "Check the motor mount."
            }
          ]
        }
      ]
    },
    {
      "blocks": [
        {
          "type": "paragraph",
          "text": "Assembly"
        }
      ]
    }
  ]
}
```

Keys

Key	Type	Default	Description
anchor	anchor		-

Key	Type	Default	Description
angle	number | string		Turn the block, in degrees counter-clockwise, about the CENTRE of its "position" box. A CONTAINER in the flow may also carry this: it becomes a GROUP that turns about its own measured box and reserves the bounding box of the turned result, and its children inherit the transform. Any other flow block ignores it with a warning -- wrap it in a container. The block lays out upright inside its box and the finished result is turned, so a "position.height" that truncates content truncates the same text it would unrotated, and the turned result is NOT clipped to its box: it may extend past it, exactly as an unrotated positioned block may. Not to be confused with the image block's "rotate", which is an integer QUARTER TURN CLOCKWISE applied to the picture INSIDE its box (and re-sizes that box to the turned orientation). The two compose -- an image can stand a sideways scan upright with "rotate" and tilt the whole frame with "angle". Same meaning as the watermark's "angle". A STRING is a data token, so each record of a template turns its own sticker: "{{s.angle}}" takes the record's number (or text of one, such as a CSV field); a record without it leaves the block upright, silently, and a value that is not a number is reported with the block's pointer (and the -merge record) and drawn upright. A string with no token is read as it stands.
bleed	number		Step and Repeat: how far this placed block's box reaches past its TRIM on every side, in points -- the box IS the bleed box, and the trim is the box inset by this much. A design tool sizes a card by it (the trim plus twice the bleed) and draws the trim line inside each copy; the engine draws nothing from it, and keeps the bleed only through the block's own clip. Only a block with position has a trim: on any other block the key is refused with a notice naming the block.
blend	string		How this block's OWN ink combines with the page under it -- the sixteen PDF blend modes (ISO 32000-1 11.3.5), written as /BM. On a container that means its background, border and stripe, never its children; for the subtree as one object see "groupBlend". Case and separators are folded, so "colorDodge", "color-dodge" and "COLORDODGE" are one mode. An unrecognized name is refused with a warning and the block paints normally. See the blend-modes sample. One of: normal, multiply, screen, overlay, darken, lighten, colorDodge, colorBurn, hardLight, softLight, difference, exclusion, hue, saturation, ...
breakBefore	boolean		Pagination: open a new page before this block - a part divider, an appendix. Ignored when the page is already empty, so it never leaves a blank page behind.
changed	boolean		Revision / change bar: draw a vertical bar in the page margin beside this block's content to flag it as changed. A document-level changeBars object customizes the bar (width / color / side / distance); a condition tag declared with changeBar:true sets this automatically for tagged content.

Key	Type	Default	Description
`clip`	`boolean`		Clip this block's children (on a container) or its own ink to its box, with the style's corner "radius". Only bites where content is BIGGER than the box: a flow container grows to fit its children vertically, so give it "position" with a height, or a "minHeight", before expecting a vertical crop. Container-scoped, like "groupOpacity": on any other flow block it is refused with a warning. For a crop to a SHAPE rather than to the box, see "clipPath"; giving both INTERSECTS them.
`clipPath`	`object`		Crop this block to an arbitrary SHAPE stated inside its box -- a design tool's FRAME. Same vocabulary as the "shape" block and read by the same outline builder, so a shape and a clipPath given identical parameters describe the identical outline. Scoped exactly as "clip" is: a positioned block of any type (it needs a "position.height", because the shape is stated inside the box) or a container in the flow; anywhere else it is refused with a warning. Naming "clip" as well INTERSECTS the two. The shape turns, scales and skews with the block that carries it.
`condition`	`string`		Conditional text: render this block only when its condition-tag expression is shown under the document's conditions map. Whitespace/comma tag terms, each optionally '!'-negated, AND-combined; an unlisted tag defaults to shown. Empty means always shown.
`float`	`string`		Float this block out of the stacked flow to one side of its column; content that FOLLOWS it wraps into the space beside it. Works on ANY block type -- a table, a container, a chart, a code listing, a figure. "center" straddles the page content width, so text in every column wraps around it. The float box takes the block's own "width"; a block that declares none takes HALF the column, so a float always leaves room beside it. Ignored on a block that also sets "position" or "anchor" -- those have already chosen their placement. Not to be confused with "anchor": { "float": true }, a BOOLEAN about migrating an anchored frame to the next column that can hold it. One of: `left`, `right`, `center`.
`groupBlend`	`string`		Composite this block AND its children once, then blend the RESULT with the page. NOT the same key as "blend", and the difference is larger than it is for opacity: two stacked children that each multiply the backdrop multiply it TWICE, which is darker than the group multiplying once, and no per-child setting reproduces it. That is what a design tool's blend-mode control on a group means. Container-scoped: on any other flow block it is refused with a warning. Also the way to blend an element together with its drop shadow, which is drawn as stacked alpha layers and so keeps its own normal compositing. See the blend-modes sample. One of: `normal`, `multiply`, `screen`, `overlay`, `darken`, `lighten`, `colorDodge`, `colorBurn`, `hardLight`, `softLight`, `difference`, `exclusion`, `hue`, `saturation`, ...
`groupOpacity`	`number`		Composite this block AND its children once, then fade the result -- true GROUP opacity. NOT the same key as "opacity", which fades each element as it paints: two overlapping opaque children at opacity 0.5 each show a seam where they cross, and under groupOpacity 0.5 they do not. The two compose. Container-scoped: on any other flow block it is refused with a warning. See the clipped-groups sample.

Key	Type	Default	Description
`items`	`array of listItem`		The list items. A bare string is plain text; an object can carry runs, its own style, or nested blocks.
`levels`	`array of string`		Multi-level: per-level numeral formats, outermost first (["upper-roman","upper-alpha","1"] => I, I.A, I.A.1). Inherited by nested lists; levels past the end reuse the last.
`marker`	`string`		Explicit marker style. An ordered style (decimal / alpha / roman) implies ordered:true. Overrides the ordered default (decimal or bullet). One of: `bullet`, `dash`, `decimal`, `1`, `lower-alpha`, `a`, `upper-alpha`, `A`, `lower-roman`, `i`, `upper-roman`, `I`.
`markerSuffix`	`string`	.	Multi-level: text drawn after the whole number ("1.1."). Set "" for "1.1".
`mirrorH`	`boolean | string`		Mirror the whole block left-to-right about the centre of its "position" box, under the same restriction as "angle". Identical to scale x -1, and the two compose. NOT "flipH": that key belongs to the image block, where it mirrors the PICTURE INSIDE its frame. The two are separate words so an image can do both. A STRING is a data token, so each record of a template mirrors its own sticker: "{{s.flip}}" takes the record's boolean, or text "true" / "false" in any case, or 1 / 0 (a number or text); a record without it draws the block unmirrored, silently, and any other value ("yes" included) is reported with the block's pointer (and the -merge record) and drawn unmirrored. A string with no token is read as it stands.
`mirrorV`	`boolean | string`		Mirror the whole block top-to-bottom about the centre of its "position" box, under the same restriction as "angle". Identical to scale y -1. NOT "flipV" -- see "mirrorH". A STRING is a data token, so each record of a template mirrors its own sticker: "{{s.flip}}" takes the record's boolean, or text "true" / "false" in any case, or 1 / 0 (a number or text); a record without it draws the block unmirrored, silently, and any other value ("yes" included) is reported with the block's pointer (and the -merge record) and drawn unmirrored. A string with no token is read as it stands.
`multiLevel`	`boolean`		Number nested ordered lists with their full ancestor path - 1, 1.1, 1.1.1 - instead of restarting per level (the legal / outline convention). Opt-in; nested ordered lists inherit it.
`opacity`	`number`		Constant opacity for this block: 0 (invisible) to 1 (opaque, the default). REAL alpha -- a /CA /ca ExtGState in the PDF -- so a translucent panel over a photograph shows the photograph through it. Not to be confused with "style.background" {color, alpha}, which composites the tint toward WHITE at parse time and so only looks right over a white page. PER-ELEMENT, NOT GROUP: on a container it paints that container's own decoration -- background, border, accent bar, stripe -- and never its children, which draw at their own opacity (1 unless they set one). Two translucent shapes that overlap therefore composite twice, which is what per-element opacity means; group opacity, where a whole subtree is composited once and then faded, is a separate feature. Where a draw already carries its own alpha (an SVG fill-opacity, a gradient's alpha) the two MULTIPLY.
`ordered`	`boolean`		Number the items (1. 2. 3.) instead of bulleting them.

Key	Type	Default	Description
repeatAt	array of object		Imposition: draw this block ONCE and place the drawing again at each of these offsets, in points from the block's own position, y DOWNWARD as every position is. One entry = one extra copy, so a list of 7 puts 8 cards on a sheet. The drawing is captured into a Form XObject and each offset is a cm + Do of that one stream, so n copies cost one copy of the content. Translation only -- angle / scale / skew are not applied per placement. A REPEATED FORM MAY NOT CARRY TAGGED CONTENT: ISO 14289-1 7.20 allows a form's content ONE semantic parent, so repeating tagged text (or a shape with alt) fails PDF/UA -- veraPDF catches it, ap_addstruct's audit does not. The engine warns and draws it anyway. Repeat decoration: a shape or svg with no alt, or put the block in a master's frames, which is drawn as a page artifact. Only a block with position is drawn again: on a block without one (in the flow, or an anchored frame), or on a repeat / if / include block, the key is refused with a notice naming the block. Each copy takes the block's groupOpacity and groupBlend. The layout map (-map) reports the copies as repeats on the block's fragment, so a design tool can select and outline them.
scale	number | array of number | object | string		Scale the block about the same pivot as "angle" (the centre of its "position" box), and under the same restriction. A CONTAINER in the flow may also carry this: it becomes a GROUP that turns about its own measured box and reserves the bounding box of the turned result, and its children inherit the transform. Any other flow block ignores it with a warning -- wrap it in a container. A number scales both axes; [x, y] or {"x":, "y":} scales them separately. 1 = natural size. Applied AFTER layout, so a scaled block does NOT re-wrap its text at the scaled measure -- this is a design tool's scale handle, not a change of font size. A STRING is a data token, so each record of a template scales its own sticker: "{{s.scale}}" takes the record's number, [x, y] or {"x":, "y":} (members may be text of numbers), or text of one or two plain numbers such as a CSV field ("1.5", "1.5 0.8", "[1.5, 0.8]"); a record without it draws the block at its natural size, silently, and a value that is not a number or a pair is reported with the block's pointer (and the -merge record) and drawn at natural size. A string with no token is read as it stands.
separator	string	.	Multi-level: separator between levels ("1.1").
skew	number | array of number | object | string		Shear the block, in DEGREES, about the same pivot as "angle", under the same restriction. [x, y] or {"x":, "y":}; a bare number skews x only. Positive x leans the top to the right and positive y lifts the right-hand side -- both counter-clockwise, matching "angle". Clamped to +/-89.9: a 90-degree skew is a singular matrix and draws nothing. A STRING is a data token, so each record of a template skews its own sticker: "{{s.skew}}" takes the record's number (x only), [x, y] or {"x":, "y":}, or text of one or two plain numbers; clamped as the written one is. A record without it draws the block unskewed, silently, and a value that is not a number or a pair is reported with the block's pointer (and the -merge record) and drawn unskewed. A string with no token is read as it stands.
start	integer	1	Ordinal of the first item.

Key	Type	Default	Description
straddle	boolean		Straddle across columns: when true and the body is multi-column, this block spans the FULL content width and text in every column wraps around it (above and below) - a spanning float generalized from center-floated images to any block. No effect in a single-column body, on a positioned / anchored block, or on an image/pdf already using float:"center" (the same mechanism). Does not balance the columns above it or act as a section divider.
verticalAlign	string		Where the content sits in its "position" box when it is shorter than the box's height: "top" (the default), "middle" ("center" is the same) or "bottom". A paragraph, heading or pre moves as a whole (its own box with it); a container keeps its box where it is and moves its children inside it (a child with its own "position" stays put). The offset is the box height less the content's height as it is actually placed -- the same lines, shrinkToFit size and padding that are then drawn -- halved for middle. Content as tall as the box or taller does not move: it is clipped and reported exactly as without the key. Needs a position with a height: on a block without one, on any other kind of block, on text set along a textPath "d" curve (stated in fractions of the whole box, so drawn across all of it; the "arc" form aligns), and on vertical text whose columns fill the box (columns run the full height unless a paragraph's or a heading's boxHeight is shorter), it is refused with a notice and ignored. The layout map keeps reporting the box. One of: top, middle, bottom, center.
z	number		CANVAS MODE ONLY (see the top-level "pages"): where this element sits in the PAINT order of its page. Lower draws first, further back; unset means "wherever I am in the list", and ties keep author order. It moves the PAINT and not the READING: the structure tree keeps the document order, so a caption written first because it is read first can still be painted last because it sits on top. That separation is what a flattened scene graph cannot express. Ignored outside canvas mode, where the flow already fixes both orders.

The pre element

Type name: pre. Accepted aliases: code.

Verbatim text, drawn line for line in a monospace face with every space preserved: a console session, a configuration file, a code listing. Unlike a paragraph it never reflows. fit shrinks an over-wide line to the column; wrap breaks it onto an indented continuation instead. Supplying tokens instead of text gives syntax coloring, and a caption promotes the block to a numbered listing.

Example

```
{
  "type": "code",
  "language": "python",
  "lineNumbers": true,
  "caption": "Reading a spec file.",
  "text": "def load(path):\n    return json.load(open(path))"
}
```

Syntax-colored tokens, with one line highlighted

```
{
  "type": "code",
  "language": "python",
  "lineNumbers": true,
  "highlightLines": [
    1
  ],
  "tokens": [
    [
      {
        "text": "def ",
        "color": "#CF222E"
      },
      {
        "text": "load",
        "color": "#8250DF"
      },
      {
        "text": "(path):",
        "color": "#1F2328"
      }
    ],
    [
      {
        "text": "    return ",
        "color": "#CF222E"
      },
      {
        "text": "json",
        "color": "#0550AE"
      },
      {
        "text": ".load(open(path))",
        "color": "#1F2328"
      }
    ]
  ]
}
```

Keys

Key	Type	Default	Description
anchor	anchor		-

Key	Type	Default	Description
`angle`	`number` | `string`		Turn the block, in degrees counter-clockwise, about the CENTRE of its "position" box. A CONTAINER in the flow may also carry this: it becomes a GROUP that turns about its own measured box and reserves the bounding box of the turned result, and its children inherit the transform. Any other flow block ignores it with a warning -- wrap it in a container. The block lays out upright inside its box and the finished result is turned, so a "position.height" that truncates content truncates the same text it would unrotated, and the turned result is NOT clipped to its box: it may extend past it, exactly as an unrotated positioned block may. Not to be confused with the image block's "rotate", which is an integer QUARTER TURN CLOCKWISE applied to the picture INSIDE its box (and re-sizes that box to the turned orientation). The two compose -- an image can stand a sideways scan upright with "rotate" and tilt the whole frame with "angle". Same meaning as the watermark's "angle". A STRING is a data token, so each record of a template turns its own sticker: "{{s.angle}}" takes the record's number (or text of one, such as a CSV field); a record without it leaves the block upright, silently, and a value that is not a number is reported with the block's pointer (and the -merge record) and drawn upright. A string with no token is read as it stands.
`bleed`	`number`		Step and Repeat: how far this placed block's box reaches past its TRIM on every side, in points -- the box IS the bleed box, and the trim is the box inset by this much. A design tool sizes a card by it (the trim plus twice the bleed) and draws the trim line inside each copy; the engine draws nothing from it, and keeps the bleed only through the block's own `clip`. Only a block with `position` has a trim: on any other block the key is refused with a notice naming the block.
`blend`	`string`		How this block's OWN ink combines with the page under it -- the sixteen PDF blend modes (ISO 32000-1 11.3.5), written as /BM. On a container that means its background, border and stripe, never its children; for the subtree as one object see "groupBlend". Case and separators are folded, so "colorDodge", "color-dodge" and "COLORDODGE" are one mode. An unrecognized name is refused with a warning and the block paints normally. See the blend-modes sample. One of: `normal`, `multiply`, `screen`, `overlay`, `darken`, `lighten`, `colorDodge`, `colorBurn`, `hardLight`, `softLight`, `difference`, `exclusion`, `hue`, `saturation`, ...
`breakBefore`	`boolean`		Pagination: open a new page before this block - a part divider, an appendix. Ignored when the page is already empty, so it never leaves a blank page behind.
`caption`	`string` | `runs`		A caption makes this a LISTING: the text is drawn above the code (in the surrounding body face) and collected for the list of listings ("lol"). Numbered when the document enables "numbering": { "listings": ... }. Accepts a plain string, an ARRAY OF RUNS, or a SINGLE RUN OBJECT - the same run forms a paragraph's "runs" takes - so a caption can carry per-span styling, and above all a per-span "fontFamily": a caption that mixes scripts sets each span in the face that suits it instead of flattening to one. The flat text is always derived from the runs and stays the entry of record: it is what reaches the list of figures, the /Alt and the structure tree. A number ("Figure 3-2.") is prepended as the first run, unstyled, so it takes the block's own face.

Key	Type	Default	Description
`changed`	`boolean`		Revision / change bar: draw a vertical bar in the page margin beside this block's content to flag it as changed. A document-level changeBars object customizes the bar (width / color / side / distance); a condition tag declared with changeBar:true sets this automatically for tagged content.
`clip`	`boolean`		Clip this block's children (on a container) or its own ink to its box, with the style's corner "radius". Only bites where content is BIGGER than the box: a flow container grows to fit its children vertically, so give it "position" with a height, or a "minHeight", before expecting a vertical crop. Container-scoped, like "groupOpacity": on any other flow block it is refused with a warning. For a crop to a SHAPE rather than to the box, see "clipPath"; giving both INTERSECTS them.
`clipPath`	`object`		Crop this block to an arbitrary SHAPE stated inside its box -- a design tool's FRAME. Same vocabulary as the "shape" block and read by the same outline builder, so a shape and a clipPath given identical parameters describe the identical outline. Scoped exactly as "clip" is: a positioned block of any type (it needs a "position.height", because the shape is stated inside the box) or a container in the flow; anywhere else it is refused with a warning. Naming "clip" as well INTERSECTS the two. The shape turns, scales and skews with the block that carries it.
`condition`	`string`		Conditional text: render this block only when its condition-tag expression is shown under the document's conditions map. Whitespace/comma tag terms, each optionally '!'-negated, AND-combined; an unlisted tag defaults to shown. Empty means always shown.
`fit`	`boolean`		Shrink the monospace font uniformly so the widest line fits the column width (default true); an over-wide line scales down instead of spilling past the margin.
`float`	`string`		Float this block out of the stacked flow to one side of its column; content that FOLLOWS it wraps into the space beside it. Works on ANY block type -- a table, a container, a chart, a code listing, a figure. "center" straddles the page content width, so text in every column wraps around it. The float box takes the block's own "width"; a block that declares none takes HALF the column, so a float always leaves room beside it. Ignored on a block that also sets "position" or "anchor" -- those have already chosen their placement. Not to be confused with "anchor": { "float": true }, a BOOLEAN about migrating an anchored frame to the next column that can hold it. One of: `left`, `right`, `center`.

Key	Type	Default	Description
groupBlend	string		Composite this block AND its children once, then blend the RESULT with the page. NOT the same key as "blend", and the difference is larger than it is for opacity: two stacked children that each multiply the backdrop multiply it TWICE, which is darker than the group multiplying once, and no per-child setting reproduces it. That is what a design tool's blend-mode control on a group means. Container-scoped: on any other flow block it is refused with a warning. Also the way to blend an element together with its drop shadow, which is drawn as stacked alpha layers and so keeps its own normal compositing. See the blend-modes sample. One of: normal, multiply, screen, overlay, darken, lighten, colorDodge, colorBurn, hardLight, softLight, difference, exclusion, hue, saturation, ...
groupOpacity	number		Composite this block AND its children once, then fade the result -- true GROUP opacity. NOT the same key as "opacity", which fades each element as it paints: two overlapping opaque children at opacity 0.5 each show a seam where they cross, and under groupOpacity 0.5 they do not. The two compose. Container-scoped: on any other flow block it is refused with a warning. See the clipped-groups sample.
highlightColor	color		Emphasis band color for highlightLines (default soft blue).
highlightLines	array of integer		1-based source lines to emphasize with a background band.
language	string		Informational language label for a code block (e.g. "python"); does not affect layout.
lineNumbers	boolean		Draw an aligned monospace line-number gutter to the left of the code (default false).
mirrorH	boolean | string		Mirror the whole block left-to-right about the centre of its "position" box, under the same restriction as "angle". Identical to scale x -1, and the two compose. NOT "flipH": that key belongs to the image block, where it mirrors the PICTURE INSIDE its frame. The two are separate words so an image can do both. A STRING is a data token, so each record of a template mirrors its own sticker: "{{s.flip}}" takes the record's boolean, or text "true" / "false" in any case, or 1 / 0 (a number or text); a record without it draws the block unmirrored, silently, and any other value ("yes" included) is reported with the block's pointer (and the -merge record) and drawn unmirrored. A string with no token is read as it stands.
mirrorV	boolean | string		Mirror the whole block top-to-bottom about the centre of its "position" box, under the same restriction as "angle". Identical to scale y -1. NOT "flipV" -- see "mirrorH". A STRING is a data token, so each record of a template mirrors its own sticker: "{{s.flip}}" takes the record's boolean, or text "true" / "false" in any case, or 1 / 0 (a number or text); a record without it draws the block unmirrored, silently, and any other value ("yes" included) is reported with the block's pointer (and the -merge record) and drawn unmirrored. A string with no token is read as it stands.
number	string		Pin a literal number/label for this block. Does not advance the counter.

Key	Type	Default	Description
numbered	boolean	true	Take part in automatic caption numbering when the document enables it. false leaves this one unnumbered and does not advance the counter.
opacity	number		Constant opacity for this block: 0 (invisible) to 1 (opaque, the default). REAL alpha -- a /CA /ca ExtGState in the PDF -- so a translucent panel over a photograph shows the photograph through it. Not to be confused with "style.background" {color, alpha}, which composites the tint toward WHITE at parse time and so only looks right over a white page. PER-ELEMENT, NOT GROUP: on a container it paints that container's own decoration -- background, border, accent bar, stripe -- and never its children, which draw at their own opacity (1 unless they set one). Two translucent shapes that overlap therefore composite twice, which is what per-element opacity means; group opacity, where a whole subtree is composited once and then faded, is a separate feature. Where a draw already carries its own alpha (an SVG fill-opacity, a gradient's alpha) the two MULTIPLY.
panelColor	color		Panel fill behind a tokenized code block (default light gray).
repeatAt	array of object		Imposition: draw this block ONCE and place the drawing again at each of these offsets, in points from the block's own position, y DOWNWARD as every position is. One entry = one extra copy, so a list of 7 puts 8 cards on a sheet. The drawing is captured into a Form XObject and each offset is a cm + Do of that one stream, so n copies cost one copy of the content. Translation only -- angle / scale / skew are not applied per placement. A REPEATED FORM MAY NOT CARRY TAGGED CONTENT: ISO 14289-1 7.20 allows a form's content ONE semantic parent, so repeating tagged text (or a shape with alt) fails PDF/UA -- veraPDF catches it, ap_addstruct's audit does not. The engine warns and draws it anyway. Repeat decoration: a shape or svg with no alt, or put the block in a master's frames, which is drawn as a page artifact. Only a block with position is drawn again: on a block without one (in the flow, or an anchored frame), or on a repeat / if / include block, the key is refused with a notice naming the block. Each copy takes the block's groupOpacity and groupBlend. The layout map (-map) reports the copies as repeats on the block's fragment, so a design tool can select and outline them.

Key	Type	Default	Description
`scale`	`number | array of number | object | string`		Scale the block about the same pivot as "angle" (the centre of its "position" box), and under the same restriction. A CONTAINER in the flow may also carry this: it becomes a GROUP that turns about its own measured box and reserves the bounding box of the turned result, and its children inherit the transform. Any other flow block ignores it with a warning -- wrap it in a container. A number scales both axes; [x, y] or {"x":, "y":} scales them separately. 1 = natural size. Applied AFTER layout, so a scaled block does NOT re-wrap its text at the scaled measure -- this is a design tool's scale handle, not a change of font size. A STRING is a data token, so each record of a template scales its own sticker: "{{s.scale}}" takes the record's number, [x, y] or {"x":, "y":} (members may be text of numbers), or text of one or two plain numbers such as a CSV field ("1.5", "1.5 0.8", "[1.5, 0.8]"); a record without it draws the block at its natural size, silently, and a value that is not a number or a pair is reported with the block's pointer (and the -merge record) and drawn at natural size. A string with no token is read as it stands.
`skew`	`number | array of number | object | string`		Shear the block, in DEGREES, about the same pivot as "angle", under the same restriction. [x, y] or {"x":, "y":}; a bare number skews x only. Positive x leans the top to the right and positive y lifts the right-hand side -- both counter-clockwise, matching "angle". Clamped to +/-89.9: a 90-degree skew is a singular matrix and draws nothing. A STRING is a data token, so each record of a template skews its own sticker: "{{s.skew}}" takes the record's number (x only), [x, y] or {"x":, "y":}, or text of one or two plain numbers; clamped as the written one is. A record without it draws the block unskewed, silently, and a value that is not a number or a pair is reported with the block's pointer (and the -merge record) and drawn unskewed. A string with no token is read as it stands.
`straddle`	`boolean`		Straddle across columns: when true and the body is multi-column, this block spans the FULL content width and text in every column wraps around it (above and below) - a spanning float generalized from center-floated images to any block. No effect in a single-column body, on a positioned / anchored block, or on an image/pdf already using float:"center" (the same mechanism). Does not balance the columns above it or act as a section divider.
`text`	`string`		Verbatim preformatted text (a Unix-man-page Quick Reference, a code listing): drawn line-for-line (\n = a hard line break) with every space preserved, in a monospace font - Courier by default, but only when nobody chose a face: a `fontFamily` on the block wins, and so does one inherited from a container or table cell, so a `pre` inside a styled box sets in that box's monospace rather than falling back to Base-14 - no whitespace collapsing, wrapping or reflow. Ignored when "tokens" is present.
`tokens`	`array of array of string | object`		Syntax highlighting: a pre-tokenized color stream that supersedes "text". One inner array per source line; each token is a colored span whose whitespace is preserved so the source line reconstructs exactly. A tokenized code block paints a panel behind every line.

Key	Type	Default	Description
`verticalAlign`	`string`		Where the content sits in its "position" box when it is shorter than the box's height: "top" (the default), "middle" ("center" is the same) or "bottom". A paragraph, heading or pre moves as a whole (its own box with it); a container keeps its box where it is and moves its children inside it (a child with its own "position" stays put). The offset is the box height less the content's height as it is actually placed -- the same lines, shrinkToFit size and padding that are then drawn -- halved for middle. Content as tall as the box or taller does not move: it is clipped and reported exactly as without the key. Needs a position with a height: on a block without one, on any other kind of block, on text set along a textPath "d" curve (stated in fractions of the whole box, so drawn across all of it; the "arc" form aligns), and on vertical text whose columns fill the box (columns run the full height unless a paragraph's or a heading's boxHeight is shorter), it is refused with a notice and ignored. The layout map keeps reporting the box. One of: `top`, `middle`, `bottom`, `center`.
`wrap`	`boolean`		Soft-wrap over-wide lines onto indented continuation lines instead of shrinking (default false; takes precedence over fit). Short lines keep full size; a long line breaks at a space (or hard-breaks an unbreakable token) so it never spills past the margin.
`z`	`number`		CANVAS MODE ONLY (see the top-level "pages"): where this element sits in the PAINT order of its page. Lower draws first, further back; unset means "wherever I am in the list", and ties keep author order. It moves the PAINT and not the READING: the structure tree keeps the document order, so a caption written first because it is read first can still be painted last because it sits on top. That separation is what a flattened scene graph cannot express. Ignored outside canvas mode, where the flow already fixes both orders.

6. Structure and spacing

Elements that group other elements or shape the page.

The container element

Type name: `container`.

A group of blocks treated as one. A container is how you give several blocks a shared background, border, padding or margin - a boxed sidebar, a tinted panel, a card. It is also the unit of `keepTogether`, which asks the engine not to split the group across a page break.

Example

```
{
  "type": "container",
  "keepTogether": true,
  "style": {
    "background": "#eef2f8",
    "padding": 10,
    "border": {
      "width": 0.75,
      "color": "#2a4d7a"
    }
  },
  "blocks": [
    {
      "type": "paragraph",
      "text": "A boxed aside."
    }
  ]
}
```

Keys

Key	Type	Default	Description
`anchor`	`anchor`		-
`angle`	`number | string`		Turn the block, in degrees counter-clockwise, about the CENTRE of its "position" box. A CONTAINER in the flow may also carry this: it becomes a GROUP that turns about its own measured box and reserves the bounding box of the turned result, and its children inherit the transform. Any other flow block ignores it with a warning -- wrap it in a container. The block lays out upright inside its box and the finished result is turned, so a "position.height" that truncates content truncates the same text it would unrotated, and the turned result is NOT clipped to its box: it may extend past it, exactly as an unrotated positioned block may. Not to be confused with the image block's "rotate", which is an integer QUARTER TURN CLOCKWISE applied to the picture INSIDE its box (and re-sizes that box to the turned orientation). The two compose -- an image can stand a sideways scan upright with "rotate" and tilt the whole frame with "angle". Same meaning as the watermark's "angle". A STRING is a data token, so each record of a template turns its own sticker: "{{s.angle}}" takes the record's number (or text of one, such as a CSV field); a record without it leaves the block upright, silently, and a value that is not a number is reported with the block's pointer (and the -merge record) and drawn upright. A string with no token is read as it stands.
`bleed`	`number`		Step and Repeat: how far this placed block's box reaches past its TRIM on every side, in points -- the box IS the bleed box, and the trim is the box inset by this much. A design tool sizes a card by it (the trim plus twice the bleed) and draws the trim line inside each copy; the engine draws nothing from it, and keeps the bleed only through the block's own `clip`. Only a block with `position` has a trim: on any other block the key is refused with a notice naming the block.

Key	Type	Default	Description
`blend`	`string`		How this block's OWN ink combines with the page under it -- the sixteen PDF blend modes (ISO 32000-1 11.3.5), written as /BM. On a container that means its background, border and stripe, never its children; for the subtree as one object see "groupBlend". Case and separators are folded, so "colorDodge", "color-dodge" and "COLORDODGE" are one mode. An unrecognized name is refused with a warning and the block paints normally. See the blend-modes sample. One of: `normal`, `multiply`, `screen`, `overlay`, `darken`, `lighten`, `colorDodge`, `colorBurn`, `hardLight`, `softLight`, `difference`, `exclusion`, `hue`, `saturation`, ...
`blocks`	`array of block`		The blocks held by this container, in reading order. A container is also how you give a run of content one shared style or background.
`boxAlign`	`string`	left	Horizontal placement of the box when width makes it narrower than the space it is placed in (default "left"). Physical edges: "left" is the left edge in a right-to-left document too. It moves the box, not the text inside it -- that is align. No effect on a floated container, which float places. A box keeps its place on every page it splits onto. One of: `left`, `center`, `right`.
`breakBefore`	`boolean`		Pagination: open a new page before this block - a part divider, an appendix. Ignored when the page is already empty, so it never leaves a blank page behind.
`changed`	`boolean`		Revision / change bar: draw a vertical bar in the page margin beside this block's content to flag it as changed. A document-level changeBars object customizes the bar (width / color / side / distance); a condition tag declared with changeBar:true sets this automatically for tagged content.
`childOrigin`	`string`	page	Where the position.x / position.y of this container's POSITIONED descendants are measured from. "page" (what an unwritten key means, unless an enclosing container set "container"): the trim page's top-left, as everywhere else. "container": this container's own box's top-left -- the BORDER box, not the content box inside the padding: for a positioned container its position rect (its fragment in the layout map), for one in the flow the box it paints where it landed, after pagination and in whatever column (its `visual` rect in the map, recorded even when it paints nothing; on a page a split container continues onto, the part placed there). So a sticker at {x: 10, y: 10} sits 10 pt in from the card's corner wherever the card is. The nearest container that WRITES the key decides: a nested "page" resets its subtree to page coordinates, and a nested container that does not write it inherits the rule (its descendants measure from the ancestor that said "container", not from it). Width and height are unaffected. The container's own angle / scale / skew / mirror, clip and groupOpacity apply to its children as before -- they are laid out in the container's upright box and turn with it; verticalAlign on the container does not move them. Everything that reports geometry reports the resulting PAGE coordinates: the layout map, TextShop line records, link rectangles, textWrap bands, clip notices. On any block other than a container, and for any other value, it is refused with a notice and ignored. One of: `page`, `container`.

Key	Type	Default	Description
`clip`	`boolean`		Clip this block's children (on a container) or its own ink to its box, with the style's corner "radius". Only bites where content is BIGGER than the box: a flow container grows to fit its children vertically, so give it "position" with a height, or a "minHeight", before expecting a vertical crop. Container-scoped, like "groupOpacity": on any other flow block it is refused with a warning. For a crop to a SHAPE rather than to the box, see "clipPath"; giving both INTERSECTS them.
`clipPath`	`object`		Crop this block to an arbitrary SHAPE stated inside its box -- a design tool's FRAME. Same vocabulary as the "shape" block and read by the same outline builder, so a shape and a clipPath given identical parameters describe the identical outline. Scoped exactly as "clip" is: a positioned block of any type (it needs a "position.height", because the shape is stated inside the box) or a container in the flow; anywhere else it is refused with a warning. Naming "clip" as well INTERSECTS the two. The shape turns, scales and skews with the block that carries it.
`condition`	`string`		Conditional text: render this block only when its condition-tag expression is shown under the document's conditions map. Whitespace/comma tag terms, each optionally '!'-negated, AND-combined; an unlisted tag defaults to shown. Empty means always shown.
`float`	`string`		Float this block out of the stacked flow to one side of its column; content that FOLLOWS it wraps into the space beside it. Works on ANY block type -- a table, a container, a chart, a code listing, a figure. "center" straddles the page content width, so text in every column wraps around it. The float box takes the block's own "width"; a block that declares none takes HALF the column, so a float always leaves room beside it. Ignored on a block that also sets "position" or "anchor" -- those have already chosen their placement. Not to be confused with "anchor": { "float": true }, a BOOLEAN about migrating an anchored frame to the next column that can hold it. One of: `left`, `right`, `center`.
`groupBlend`	`string`		Composite this block AND its children once, then blend the RESULT with the page. NOT the same key as "blend", and the difference is larger than it is for opacity: two stacked children that each multiply the backdrop multiply it TWICE, which is darker than the group multiplying once, and no per-child setting reproduces it. That is what a design tool's blend-mode control on a group means. Container-scoped: on any other flow block it is refused with a warning. Also the way to blend an element together with its drop shadow, which is drawn as stacked alpha layers and so keeps its own normal compositing. See the blend-modes sample. One of: `normal`, `multiply`, `screen`, `overlay`, `darken`, `lighten`, `colorDodge`, `colorBurn`, `hardLight`, `softLight`, `difference`, `exclusion`, `hue`, `saturation`, ...
`groupOpacity`	`number`		Composite this block AND its children once, then fade the result -- true GROUP opacity. NOT the same key as "opacity", which fades each element as it paints: two overlapping opaque children at opacity 0.5 each show a seam where they cross, and under groupOpacity 0.5 they do not. The two compose. Container-scoped: on any other flow block it is refused with a warning. See the clipped-groups sample.

Key	Type	Default	Description
keepTogether	boolean		Keep the whole group on one page when it fits a fresh one: it moves to the next page whole rather than split. Unwritten, it is true for a container that paints a box (a style background, border, accentBar or stripe), as for a callout, and false for one that paints nothing. Write false to let a box split. A container that is split -- because it is taller than a page, or keepTogether is false -- keeps its side padding, width and boxAlign on every page it reaches, and takes its bottom padding and spaceAfter where it ends. One that paints a box is drawn on every page, open where the page cuts it, and repeats its top padding below each cut; one that paints nothing does not.
mirrorH	boolean | string		Mirror the whole block left-to-right about the centre of its "position" box, under the same restriction as "angle". Identical to scale x -1, and the two compose. NOT "flipH": that key belongs to the image block, where it mirrors the PICTURE INSIDE its frame. The two are separate words so an image can do both. A STRING is a data token, so each record of a template mirrors its own sticker: "{{s.flip}}" takes the record's boolean, or text "true" / "false" in any case, or 1 / 0 (a number or text); a record without it draws the block unmirrored, silently, and any other value ("yes" included) is reported with the block's pointer (and the -merge record) and drawn unmirrored. A string with no token is read as it stands.
mirrorV	boolean | string		Mirror the whole block top-to-bottom about the centre of its "position" box, under the same restriction as "angle". Identical to scale y -1. NOT "flipV" -- see "mirrorH". A STRING is a data token, so each record of a template mirrors its own sticker: "{{s.flip}}" takes the record's boolean, or text "true" / "false" in any case, or 1 / 0 (a number or text); a record without it draws the block unmirrored, silently, and any other value ("yes" included) is reported with the block's pointer (and the -merge record) and drawn unmirrored. A string with no token is read as it stands.
opacity	number		Constant opacity for this block: 0 (invisible) to 1 (opaque, the default). REAL alpha -- a /CA /ca ExtGState in the PDF -- so a translucent panel over a photograph shows the photograph through it. Not to be confused with "style.background" {color, alpha}, which composites the tint toward WHITE at parse time and so only looks right over a white page. PER-ELEMENT, NOT GROUP: on a container it paints that container's own decoration -- background, border, accent bar, stripe -- and never its children, which draw at their own opacity (1 unless they set one). Two translucent shapes that overlap therefore composite twice, which is what per-element opacity means; group opacity, where a whole subtree is composited once and then faded, is a separate feature. Where a draw already carries its own alpha (an SVG fill-opacity, a gradient's alpha) the two MULTIPLY.

Key	Type	Default	Description
repeatAt	array of object		Imposition: draw this block ONCE and place the drawing again at each of these offsets, in points from the block's own position, y DOWNWARD as every position is. One entry = one extra copy, so a list of 7 puts 8 cards on a sheet. The drawing is captured into a Form XObject and each offset is a cm + Do of that one stream, so n copies cost one copy of the content. Translation only -- angle / scale / skew are not applied per placement. A REPEATED FORM MAY NOT CARRY TAGGED CONTENT: ISO 14289-1 7.20 allows a form's content ONE semantic parent, so repeating tagged text (or a shape with alt) fails PDF/UA -- veraPDF catches it, ap_addstruct's audit does not. The engine warns and draws it anyway. Repeat decoration: a shape or svg with no alt, or put the block in a master's frames, which is drawn as a page artifact. Only a block with position is drawn again: on a block without one (in the flow, or an anchored frame), or on a repeat / if / include block, the key is refused with a notice naming the block. Each copy takes the block's groupOpacity and groupBlend. The layout map (-map) reports the copies as repeats on the block's fragment, so a design tool can select and outline them.
role	string	none	The STRUCTURE TYPE this box announces itself as, for assistive technology. ONLY ON A container BLOCK -- "role" is an ordinary word and appears freely in a document's own DATA (a {name, role, years} record bound by a repeat, say); nothing outside a container block is touched by this key. Default "none" = no structure element at all, which is what every container was before this key existed and what a purely visual grouping (a tile, a swatch, a specimen box) should stay. "aside" is content outside the main narrative -- a sidebar, a pull quote, a closing commentary band -- and is the type a callout sets for itself; it is written as <Aside> in the PDF 2.0 namespace under PDF/UA-2, and elsewhere as <Aside> plus a /RoleMap to Div, so a pre-PDF-2.0 reader still gets a standard type. "section" is <Sect>, "div" a generic block-level grouping, "blockquote" an extended quotation. An unknown value REFUSES the document rather than being ignored -- a role that silently does nothing is precisely the defect this key exists to fix. Not offered: Figure (needs /Alt and has its own block), Note (PDF 1.7 reserves it for content REFERRED TO from the body, which is the footnote path), Caption, TOC. One of: aside, section, div, blockquote, none.

Key	Type	Default	Description
`scale`	`number | array of number | object | string`		Scale the block about the same pivot as "angle" (the centre of its "position" box), and under the same restriction. A CONTAINER in the flow may also carry this: it becomes a GROUP that turns about its own measured box and reserves the bounding box of the turned result, and its children inherit the transform. Any other flow block ignores it with a warning -- wrap it in a container. A number scales both axes; [x, y] or {"x":, "y":} scales them separately. 1 = natural size. Applied AFTER layout, so a scaled block does NOT re-wrap its text at the scaled measure -- this is a design tool's scale handle, not a change of font size. A STRING is a data token, so each record of a template scales its own sticker: "{{s.scale}}" takes the record's number, [x, y] or {"x":, "y":} (members may be text of numbers), or text of one or two plain numbers such as a CSV field ("1.5", "1.5 0.8", "[1.5, 0.8]"); a record without it draws the block at its natural size, silently, and a value that is not a number or a pair is reported with the block's pointer (and the -merge record) and drawn at natural size. A string with no token is read as it stands.
`skew`	`number | array of number | object | string`		Shear the block, in DEGREES, about the same pivot as "angle", under the same restriction. [x, y] or {"x":, "y":}; a bare number skews x only. Positive x leans the top to the right and positive y lifts the right-hand side -- both counter-clockwise, matching "angle". Clamped to +/-89.9: a 90-degree skew is a singular matrix and draws nothing. A STRING is a data token, so each record of a template skews its own sticker: "{{s.skew}}" takes the record's number (x only), [x, y] or {"x":, "y":}, or text of one or two plain numbers; clamped as the written one is. A record without it draws the block unskewed, silently, and a value that is not a number or a pair is reported with the block's pointer (and the -merge record) and drawn unskewed. A string with no token is read as it stands.
`straddle`	`boolean`		Straddle across columns: when true and the body is multi-column, this block spans the FULL content width and text in every column wraps around it (above and below) - a spanning float generalized from center-floated images to any block. No effect in a single-column body, on a positioned / anchored block, or on an image/pdf already using float:"center" (the same mechanism). Does not balance the columns above it or act as a section divider.
`verticalAlign`	`string`		Where the content sits in its "position" box when it is shorter than the box's height: "top" (the default), "middle" ("center" is the same) or "bottom". A paragraph, heading or pre moves as a whole (its own box with it); a container keeps its box where it is and moves its children inside it (a child with its own "position" stays put). The offset is the box height less the content's height as it is actually placed -- the same lines, shrinkToFit size and padding that are then drawn -- halved for middle. Content as tall as the box or taller does not move: it is clipped and reported exactly as without the key. Needs a position with a height: on a block without one, on any other kind of block, on text set along a textPath "d" curve (stated in fractions of the whole box, so drawn across all of it; the "arc" form aligns), and on vertical text whose columns fill the box (columns run the full height unless a paragraph's or a heading's boxHeight is shorter), it is refused with a notice and ignored. The layout map keeps reporting the box. One of: `top`, `middle`, `bottom`, `center`.

Key	Type	Default	Description
`width`	`dimension`		Explicit box width (default "auto" = the full width): points, or "NN%" of the width the container is placed in -- its column, or the cell, card or box that holds it. The box takes that width, its blocks wrap inside its padding, and boxAlign places it; the room beside a narrower box stays empty (to run text beside a box, float it). A width wider than the space takes the full width, as does zero or less. A box keeps its width and boxAlign on every page it splits onto. On a floated container it sizes the float box instead -- points or a percentage of the column; unwritten, half the column -- and the box fills it.
`z`	`number`		CANVAS MODE ONLY (see the top-level "pages"): where this element sits in the PAINT order of its page. Lower draws first, further back; unset means "wherever I am in the list", and ties keep author order. It moves the PAINT and not the READING: the structure tree keeps the document order, so a caption written first because it is read first can still be painted last because it sits on top. That separation is what a flattened scene graph cannot express. Ignored outside canvas mode, where the flow already fixes both orders.

The columns element

Type name: `columns`.

A row of side-by-side columns, each holding its own block. Use it for a two-up comparison, a label-and-value pair, or a picture beside its explanation. This is different from `pageSetup.columns`, which pours a single stream of text down multiple newspaper columns; `columns` places distinct content side by side.

Example

```
{
  "type": "columns",
  "gap": 18,
  "columns": [
    {
      "width": "50%",
      "block": {
        "type": "paragraph",
        "text": "Left."
      }
    },
    {
      "width": "50%",
      "block": {
        "type": "paragraph",
        "text": "Right."
      }
    }
  ]
}
```

Keys

Key	Type	Default	Description
anchor	anchor		-
angle	number | string		Turn the block, in degrees counter-clockwise, about the CENTRE of its "position" box. A CONTAINER in the flow may also carry this: it becomes a GROUP that turns about its own measured box and reserves the bounding box of the turned result, and its children inherit the transform. Any other flow block ignores it with a warning -- wrap it in a container. The block lays out upright inside its box and the finished result is turned, so a "position.height" that truncates content truncates the same text it would unrotated, and the turned result is NOT clipped to its box: it may extend past it, exactly as an unrotated positioned block may. Not to be confused with the image block's "rotate", which is an integer QUARTER TURN CLOCKWISE applied to the picture INSIDE its box (and re-sizes that box to the turned orientation). The two compose -- an image can stand a sideways scan upright with "rotate" and tilt the whole frame with "angle". Same meaning as the watermark's "angle". A STRING is a data token, so each record of a template turns its own sticker: "{{s.angle}}" takes the record's number (or text of one, such as a CSV field); a record without it leaves the block upright, silently, and a value that is not a number is reported with the block's pointer (and the -merge record) and drawn upright. A string with no token is read as it stands.
bleed	number		Step and Repeat: how far this placed block's box reaches past its TRIM on every side, in points -- the box IS the bleed box, and the trim is the box inset by this much. A design tool sizes a card by it (the trim plus twice the bleed) and draws the trim line inside each copy; the engine draws nothing from it, and keeps the bleed only through the block's own clip. Only a block with position has a trim: on any other block the key is refused with a notice naming the block.
blend	string		How this block's OWN ink combines with the page under it -- the sixteen PDF blend modes (ISO 32000-1 11.3.5), written as /BM. On a container that means its background, border and stripe, never its children; for the subtree as one object see "groupBlend". Case and separators are folded, so "colorDodge", "color-dodge" and "COLORDODGE" are one mode. An unrecognized name is refused with a warning and the block paints normally. See the blend-modes sample. One of: normal, multiply, screen, overlay, darken, lighten, colorDodge, colorBurn, hardLight, softLight, difference, exclusion, hue, saturation, ...
breakBefore	boolean		Pagination: open a new page before this block - a part divider, an appendix. Ignored when the page is already empty, so it never leaves a blank page behind.
changed	boolean		Revision / change bar: draw a vertical bar in the page margin beside this block's content to flag it as changed. A document-level changeBars object customizes the bar (width / color / side / distance); a condition tag declared with changeBar:true sets this automatically for tagged content.

Key	Type	Default	Description
`clip`	`boolean`		Clip this block's children (on a container) or its own ink to its box, with the style's corner "radius". Only bites where content is BIGGER than the box: a flow container grows to fit its children vertically, so give it "position" with a height, or a "minHeight", before expecting a vertical crop. Container-scoped, like "groupOpacity": on any other flow block it is refused with a warning. For a crop to a SHAPE rather than to the box, see "clipPath"; giving both INTERSECTS them.
`clipPath`	`object`		Crop this block to an arbitrary SHAPE stated inside its box -- a design tool's FRAME. Same vocabulary as the "shape" block and read by the same outline builder, so a shape and a clipPath given identical parameters describe the identical outline. Scoped exactly as "clip" is: a positioned block of any type (it needs a "position.height", because the shape is stated inside the box) or a container in the flow; anywhere else it is refused with a warning. Naming "clip" as well INTERSECTS the two. The shape turns, scales and skews with the block that carries it.
`columns`	`array of object`		The columns themselves - each holds its own blocks, and may set a width.
`condition`	`string`		Conditional text: render this block only when its condition-tag expression is shown under the document's conditions map. Whitespace/comma tag terms, each optionally '!'-negated, AND-combined; an unlisted tag defaults to shown. Empty means always shown.
`float`	`string`		Float this block out of the stacked flow to one side of its column; content that FOLLOWS it wraps into the space beside it. Works on ANY block type -- a table, a container, a chart, a code listing, a figure. "center" straddles the page content width, so text in every column wraps around it. The float box takes the block's own "width"; a block that declares none takes HALF the column, so a float always leaves room beside it. Ignored on a block that also sets "position" or "anchor" -- those have already chosen their placement. Not to be confused with "anchor": { "float": true }, a BOOLEAN about migrating an anchored frame to the next column that can hold it. One of: `left`, `right`, `center`.
`gap`	`number`		Gutter between the columns, in points.
`groupBlend`	`string`		Composite this block AND its children once, then blend the RESULT with the page. NOT the same key as "blend", and the difference is larger than it is for opacity: two stacked children that each multiply the backdrop multiply it TWICE, which is darker than the group multiplying once, and no per-child setting reproduces it. That is what a design tool's blend-mode control on a group means. Container-scoped: on any other flow block it is refused with a warning. Also the way to blend an element together with its drop shadow, which is drawn as stacked alpha layers and so keeps its own normal compositing. See the blend-modes sample. One of: `normal`, `multiply`, `screen`, `overlay`, `darken`, `lighten`, `colorDodge`, `colorBurn`, `hardLight`, `softLight`, `difference`, `exclusion`, `hue`, `saturation`, ...

Key	Type	Default	Description
groupOpacity	number		Composite this block AND its children once, then fade the result -- true GROUP opacity. NOT the same key as "opacity", which fades each element as it paints: two overlapping opaque children at opacity 0.5 each show a seam where they cross, and under groupOpacity 0.5 they do not. The two compose. Container-scoped: on any other flow block it is refused with a warning. See the clipped-groups sample.
mirrorH	boolean | string		Mirror the whole block left-to-right about the centre of its "position" box, under the same restriction as "angle". Identical to scale x -1, and the two compose. NOT "flipH": that key belongs to the image block, where it mirrors the PICTURE INSIDE its frame. The two are separate words so an image can do both. A STRING is a data token, so each record of a template mirrors its own sticker: "{{s.flip}}" takes the record's boolean, or text "true" / "false" in any case, or 1 / 0 (a number or text); a record without it draws the block unmirrored, silently, and any other value ("yes" included) is reported with the block's pointer (and the -merge record) and drawn unmirrored. A string with no token is read as it stands.
mirrorV	boolean | string		Mirror the whole block top-to-bottom about the centre of its "position" box, under the same restriction as "angle". Identical to scale y -1. NOT "flipV" -- see "mirrorH". A STRING is a data token, so each record of a template mirrors its own sticker: "{{s.flip}}" takes the record's boolean, or text "true" / "false" in any case, or 1 / 0 (a number or text); a record without it draws the block unmirrored, silently, and any other value ("yes" included) is reported with the block's pointer (and the -merge record) and drawn unmirrored. A string with no token is read as it stands.
opacity	number		Constant opacity for this block: 0 (invisible) to 1 (opaque, the default). REAL alpha -- a /CA /ca ExtGState in the PDF -- so a translucent panel over a photograph shows the photograph through it. Not to be confused with "style.background" {color, alpha}, which composites the tint toward WHITE at parse time and so only looks right over a white page. PER-ELEMENT, NOT GROUP: on a container it paints that container's own decoration -- background, border, accent bar, stripe -- and never its children, which draw at their own opacity (1 unless they set one). Two translucent shapes that overlap therefore composite twice, which is what per-element opacity means; group opacity, where a whole subtree is composited once and then faded, is a separate feature. Where a draw already carries its own alpha (an SVG fill-opacity, a gradient's alpha) the two MULTIPLY.

Key	Type	Default	Description
`repeatAt`	`array of object`		Imposition: draw this block ONCE and place the drawing again at each of these offsets, in points from the block's own position, y DOWNWARD as every `position` is. One entry = one extra copy, so a list of 7 puts 8 cards on a sheet. The drawing is captured into a Form XObject and each offset is a `cm` + `Do` of that one stream, so n copies cost one copy of the content. Translation only -- `angle` / `scale` / `skew` are not applied per placement. A REPEATED FORM MAY NOT CARRY TAGGED CONTENT: ISO 14289-1 7.20 allows a form's content ONE semantic parent, so repeating tagged text (or a shape with `alt`) fails PDF/UA -- veraPDF catches it, ap_addstruct's audit does not. The engine warns and draws it anyway. Repeat decoration: a shape or svg with no `alt`, or put the block in a master's `frames`, which is drawn as a page artifact. Only a block with `position` is drawn again: on a block without one (in the flow, or an anchored frame), or on a repeat / if / include block, the key is refused with a notice naming the block. Each copy takes the block's `groupOpacity` and `groupBlend`. The layout map (-map) reports the copies as `repeats` on the block's fragment, so a design tool can select and outline them.
`scale`	`number | array of number | object | string`		Scale the block about the same pivot as "angle" (the centre of its "position" box), and under the same restriction. A CONTAINER in the flow may also carry this: it becomes a GROUP that turns about its own measured box and reserves the bounding box of the turned result, and its children inherit the transform. Any other flow block ignores it with a warning -- wrap it in a container. A number scales both axes; [x, y] or {"x":, "y":} scales them separately. 1 = natural size. Applied AFTER layout, so a scaled block does NOT re-wrap its text at the scaled measure -- this is a design tool's scale handle, not a change of font size. A STRING is a data token, so each record of a template scales its own sticker: "{{s.scale}}" takes the record's number, [x, y] or {"x":, "y":} (members may be text of numbers), or text of one or two plain numbers such as a CSV field ("1.5", "1.5 0.8", "[1.5, 0.8]"); a record without it draws the block at its natural size, silently, and a value that is not a number or a pair is reported with the block's pointer (and the -merge record) and drawn at natural size. A string with no token is read as it stands.
`skew`	`number | array of number | object | string`		Shear the block, in DEGREES, about the same pivot as "angle", under the same restriction. [x, y] or {"x":, "y":}; a bare number skews x only. Positive x leans the top to the right and positive y lifts the right-hand side -- both counter-clockwise, matching "angle". Clamped to +/-89.9: a 90-degree skew is a singular matrix and draws nothing. A STRING is a data token, so each record of a template skews its own sticker: "{{s.skew}}" takes the record's number (x only), [x, y] or {"x":, "y":}, or text of one or two plain numbers; clamped as the written one is. A record without it draws the block unskewed, silently, and a value that is not a number or a pair is reported with the block's pointer (and the -merge record) and drawn unskewed. A string with no token is read as it stands.

Key	Type	Default	Description
straddle	boolean		Straddle across columns: when true and the body is multi-column, this block spans the FULL content width and text in every column wraps around it (above and below) - a spanning float generalized from center-floated images to any block. No effect in a single-column body, on a positioned / anchored block, or on an image/pdf already using float:"center" (the same mechanism). Does not balance the columns above it or act as a section divider.
valign	string		How the columns align to each other when they differ in height. "stretch" makes them equal. One of: top, center, bottom, stretch.
verticalAlign	string		Where the content sits in its "position" box when it is shorter than the box's height: "top" (the default), "middle" ("center" is the same) or "bottom". A paragraph, heading or pre moves as a whole (its own box with it); a container keeps its box where it is and moves its children inside it (a child with its own "position" stays put). The offset is the box height less the content's height as it is actually placed -- the same lines, shrinkToFit size and padding that are then drawn -- halved for middle. Content as tall as the box or taller does not move: it is clipped and reported exactly as without the key. Needs a position with a height: on a block without one, on any other kind of block, on text set along a textPath "d" curve (stated in fractions of the whole box, so drawn across all of it; the "arc" form aligns), and on vertical text whose columns fill the box (columns run the full height unless a paragraph's or a heading's boxHeight is shorter), it is refused with a notice and ignored. The layout map keeps reporting the box. One of: top, middle, bottom, center.
z	number		CANVAS MODE ONLY (see the top-level "pages"): where this element sits in the PAINT order of its page. Lower draws first, further back; unset means "wherever I am in the list", and ties keep author order. It moves the PAINT and not the READING: the structure tree keeps the document order, so a caption written first because it is read first can still be painted last because it sits on top. That separation is what a flattened scene graph cannot express. Ignored outside canvas mode, where the flow already fixes both orders.

The cardGrid element

Type name: `cardGrid`. Accepted aliases: `cardgrid`.

A grid of cards that wraps: you supply the items, the engine decides how many fit per row and flows the rest onto the next. The signature layout of a dashboard or a product summary page.

Example

```
{
  "type": "cardGrid",
  "minColWidth": 150,
  "gap": 12,
  "items": [
    {
      "type": "container",
      "blocks": [
        {
          "type": "paragraph",
          "text": "One"
        }
      ]
    },
    {
      "type": "container",
      "blocks": [
        {
          "type": "paragraph",
          "text": "Two"
        }
      ]
    }
  ]
}
```

Keys

Key	Type	Default	Description
`anchor`	`anchor`		-

Key	Type	Default	Description
angle	number | string		Turn the block, in degrees counter-clockwise, about the CENTRE of its "position" box. A CONTAINER in the flow may also carry this: it becomes a GROUP that turns about its own measured box and reserves the bounding box of the turned result, and its children inherit the transform. Any other flow block ignores it with a warning -- wrap it in a container. The block lays out upright inside its box and the finished result is turned, so a "position.height" that truncates content truncates the same text it would unrotated, and the turned result is NOT clipped to its box: it may extend past it, exactly as an unrotated positioned block may. Not to be confused with the image block's "rotate", which is an integer QUARTER TURN CLOCKWISE applied to the picture INSIDE its box (and re-sizes that box to the turned orientation). The two compose -- an image can stand a sideways scan upright with "rotate" and tilt the whole frame with "angle". Same meaning as the watermark's "angle". A STRING is a data token, so each record of a template turns its own sticker: "{{s.angle}}" takes the record's number (or text of one, such as a CSV field); a record without it leaves the block upright, silently, and a value that is not a number is reported with the block's pointer (and the -merge record) and drawn upright. A string with no token is read as it stands.
bleed	number		Step and Repeat: how far this placed block's box reaches past its TRIM on every side, in points -- the box IS the bleed box, and the trim is the box inset by this much. A design tool sizes a card by it (the trim plus twice the bleed) and draws the trim line inside each copy; the engine draws nothing from it, and keeps the bleed only through the block's own clip. Only a block with position has a trim: on any other block the key is refused with a notice naming the block.
blend	string		How this block's OWN ink combines with the page under it -- the sixteen PDF blend modes (ISO 32000-1 11.3.5), written as /BM. On a container that means its background, border and stripe, never its children; for the subtree as one object see "groupBlend". Case and separators are folded, so "colorDodge", "color-dodge" and "COLORDODGE" are one mode. An unrecognized name is refused with a warning and the block paints normally. See the blend-modes sample. One of: normal, multiply, screen, overlay, darken, lighten, colorDodge, colorBurn, hardLight, softLight, difference, exclusion, hue, saturation, ...
breakBefore	boolean		Pagination: open a new page before this block - a part divider, an appendix. Ignored when the page is already empty, so it never leaves a blank page behind.
changed	boolean		Revision / change bar: draw a vertical bar in the page margin beside this block's content to flag it as changed. A document-level changeBars object customizes the bar (width / color / side / distance); a condition tag declared with changeBar:true sets this automatically for tagged content.

Key	Type	Default	Description
clip	boolean		Clip this block's children (on a container) or its own ink to its box, with the style's corner "radius". Only bites where content is BIGGER than the box: a flow container grows to fit its children vertically, so give it "position" with a height, or a "minHeight", before expecting a vertical crop. Container-scoped, like "groupOpacity": on any other flow block it is refused with a warning. For a crop to a SHAPE rather than to the box, see "clipPath"; giving both INTERSECTS them.
clipPath	object		Crop this block to an arbitrary SHAPE stated inside its box -- a design tool's FRAME. Same vocabulary as the "shape" block and read by the same outline builder, so a shape and a clipPath given identical parameters describe the identical outline. Scoped exactly as "clip" is: a positioned block of any type (it needs a "position.height", because the shape is stated inside the box) or a container in the flow; anywhere else it is refused with a warning. Naming "clip" as well INTERSECTS the two. The shape turns, scales and skews with the block that carries it.
condition	string		Conditional text: render this block only when its condition-tag expression is shown under the document's conditions map. Whitespace/comma tag terms, each optionally '!'-negated, AND-combined; an unlisted tag defaults to shown. Empty means always shown.
float	string		Float this block out of the stacked flow to one side of its column; content that FOLLOWS it wraps into the space beside it. Works on ANY block type -- a table, a container, a chart, a code listing, a figure. "center" straddles the page content width, so text in every column wraps around it. The float box takes the block's own "width"; a block that declares none takes HALF the column, so a float always leaves room beside it. Ignored on a block that also sets "position" or "anchor" -- those have already chosen their placement. Not to be confused with "anchor": { "float": true }, a BOOLEAN about migrating an anchored frame to the next column that can hold it. One of: left, right, center.
gap	number		Gap between cards, both directions. "gapCol" and "gapRow" override it per axis.
gapCol	number		Horizontal gap between cards, overriding "gap".
gapRow	number		Vertical gap between cards, overriding "gap".
groupBlend	string		Composite this block AND its children once, then blend the RESULT with the page. NOT the same key as "blend", and the difference is larger than it is for opacity: two stacked children that each multiply the backdrop multiply it TWICE, which is darker than the group multiplying once, and no per-child setting reproduces it. That is what a design tool's blend-mode control on a group means. Container-scoped: on any other flow block it is refused with a warning. Also the way to blend an element together with its drop shadow, which is drawn as stacked alpha layers and so keeps its own normal compositing. See the blend-modes sample. One of: normal, multiply, screen, overlay, darken, lighten, colorDodge, colorBurn, hardLight, softLight, difference, exclusion, hue, saturation, ...

Key	Type	Default	Description
groupOpacity	number		Composite this block AND its children once, then fade the result -- true GROUP opacity. NOT the same key as "opacity", which fades each element as it paints: two overlapping opaque children at opacity 0.5 each show a seam where they cross, and under groupOpacity 0.5 they do not. The two compose. Container-scoped: on any other flow block it is refused with a warning. See the clipped-groups sample.
items	array of block		The cards. Each is a block group rendered as one cell of the grid.
minColWidth	number		Narrowest a column may get, IN POINTS, before the grid drops to fewer columns. This is what makes the grid responsive to the measure.
mirrorH	boolean | string		Mirror the whole block left-to-right about the centre of its "position" box, under the same restriction as "angle". Identical to scale x -1, and the two compose. NOT "flipH": that key belongs to the image block, where it mirrors the PICTURE INSIDE its frame. The two are separate words so an image can do both. A STRING is a data token, so each record of a template mirrors its own sticker: "{{s.flip}}" takes the record's boolean, or text "true" / "false" in any case, or 1 / 0 (a number or text); a record without it draws the block unmirrored, silently, and any other value ("yes" included) is reported with the block's pointer (and the -merge record) and drawn unmirrored. A string with no token is read as it stands.
mirrorV	boolean | string		Mirror the whole block top-to-bottom about the centre of its "position" box, under the same restriction as "angle". Identical to scale y -1. NOT "flipV" -- see "mirrorH". A STRING is a data token, so each record of a template mirrors its own sticker: "{{s.flip}}" takes the record's boolean, or text "true" / "false" in any case, or 1 / 0 (a number or text); a record without it draws the block unmirrored, silently, and any other value ("yes" included) is reported with the block's pointer (and the -merge record) and drawn unmirrored. A string with no token is read as it stands.
opacity	number		Constant opacity for this block: 0 (invisible) to 1 (opaque, the default). REAL alpha -- a /CA /ca ExtGState in the PDF -- so a translucent panel over a photograph shows the photograph through it. Not to be confused with "style.background" {color, alpha}, which composites the tint toward WHITE at parse time and so only looks right over a white page. PER-ELEMENT, NOT GROUP: on a container it paints that container's own decoration -- background, border, accent bar, stripe -- and never its children, which draw at their own opacity (1 unless they set one). Two translucent shapes that overlap therefore composite twice, which is what per-element opacity means; group opacity, where a whole subtree is composited once and then faded, is a separate feature. Where a draw already carries its own alpha (an SVG fill-opacity, a gradient's alpha) the two MULTIPLY.

Key	Type	Default	Description
`repeatAt`	`array of object`		Imposition: draw this block ONCE and place the drawing again at each of these offsets, in points from the block's own position, y DOWNWARD as every `position` is. One entry = one extra copy, so a list of 7 puts 8 cards on a sheet. The drawing is captured into a Form XObject and each offset is a `cm` + `Do` of that one stream, so n copies cost one copy of the content. Translation only -- `angle` / `scale` / `skew` are not applied per placement. A REPEATED FORM MAY NOT CARRY TAGGED CONTENT: ISO 14289-1 7.20 allows a form's content ONE semantic parent, so repeating tagged text (or a shape with `alt`) fails PDF/UA -- veraPDF catches it, ap_addstruct's audit does not. The engine warns and draws it anyway. Repeat decoration: a shape or svg with no `alt`, or put the block in a master's `frames`, which is drawn as a page artifact. Only a block with `position` is drawn again: on a block without one (in the flow, or an anchored frame), or on a repeat / if / include block, the key is refused with a notice naming the block. Each copy takes the block's `groupOpacity` and `groupBlend`. The layout map (-map) reports the copies as `repeats` on the block's fragment, so a design tool can select and outline them.
`scale`	`number | array of number | object | string`		Scale the block about the same pivot as "angle" (the centre of its "position" box), and under the same restriction. A CONTAINER in the flow may also carry this: it becomes a GROUP that turns about its own measured box and reserves the bounding box of the turned result, and its children inherit the transform. Any other flow block ignores it with a warning -- wrap it in a container. A number scales both axes; [x, y] or {"x":, "y":} scales them separately. 1 = natural size. Applied AFTER layout, so a scaled block does NOT re-wrap its text at the scaled measure -- this is a design tool's scale handle, not a change of font size. A STRING is a data token, so each record of a template scales its own sticker: "{{s.scale}}" takes the record's number, [x, y] or {"x":, "y":} (members may be text of numbers), or text of one or two plain numbers such as a CSV field ("1.5", "1.5 0.8", "[1.5, 0.8]"); a record without it draws the block at its natural size, silently, and a value that is not a number or a pair is reported with the block's pointer (and the -merge record) and drawn at natural size. A string with no token is read as it stands.
`skew`	`number | array of number | object | string`		Shear the block, in DEGREES, about the same pivot as "angle", under the same restriction. [x, y] or {"x":, "y":}; a bare number skews x only. Positive x leans the top to the right and positive y lifts the right-hand side -- both counter-clockwise, matching "angle". Clamped to +/-89.9: a 90-degree skew is a singular matrix and draws nothing. A STRING is a data token, so each record of a template skews its own sticker: "{{s.skew}}" takes the record's number (x only), [x, y] or {"x":, "y":}, or text of one or two plain numbers; clamped as the written one is. A record without it draws the block unskewed, silently, and a value that is not a number or a pair is reported with the block's pointer (and the -merge record) and drawn unskewed. A string with no token is read as it stands.

Key	Type	Default	Description
straddle	boolean		Straddle across columns: when true and the body is multi-column, this block spans the FULL content width and text in every column wraps around it (above and below) - a spanning float generalized from center-floated images to any block. No effect in a single-column body, on a positioned / anchored block, or on an image/pdf already using float:"center" (the same mechanism). Does not balance the columns above it or act as a section divider.
verticalAlign	string		Where the content sits in its "position" box when it is shorter than the box's height: "top" (the default), "middle" ("center" is the same) or "bottom". A paragraph, heading or pre moves as a whole (its own box with it); a container keeps its box where it is and moves its children inside it (a child with its own "position" stays put). The offset is the box height less the content's height as it is actually placed -- the same lines, shrinkToFit size and padding that are then drawn -- halved for middle. Content as tall as the box or taller does not move: it is clipped and reported exactly as without the key. Needs a position with a height: on a block without one, on any other kind of block, on text set along a textPath "d" curve (stated in fractions of the whole box, so drawn across all of it; the "arc" form aligns), and on vertical text whose columns fill the box (columns run the full height unless a paragraph's or a heading's boxHeight is shorter), it is refused with a notice and ignored. The layout map keeps reporting the box. One of: top, middle, bottom, center.
z	number		CANVAS MODE ONLY (see the top-level "pages"): where this element sits in the PAINT order of its page. Lower draws first, further back; unset means "wherever I am in the list", and ties keep author order. It moves the PAINT and not the READING: the structure tree keeps the document order, so a caption written first because it is read first can still be painted last because it sits on top. That separation is what a flattened scene graph cannot express. Ignored outside canvas mode, where the flow already fixes both orders.

The spacer element

Type name: spacer.

Blank vertical space, in points. The simplest way to open a gap.

Example

```
{
  "type": "spacer",
  "height": 12
}
```

Keys

Key	Type	Default	Description
anchor	anchor		-

Key	Type	Default	Description
angle	number | string		Turn the block, in degrees counter-clockwise, about the CENTRE of its "position" box. A CONTAINER in the flow may also carry this: it becomes a GROUP that turns about its own measured box and reserves the bounding box of the turned result, and its children inherit the transform. Any other flow block ignores it with a warning -- wrap it in a container. The block lays out upright inside its box and the finished result is turned, so a "position.height" that truncates content truncates the same text it would unrotated, and the turned result is NOT clipped to its box: it may extend past it, exactly as an unrotated positioned block may. Not to be confused with the image block's "rotate", which is an integer QUARTER TURN CLOCKWISE applied to the picture INSIDE its box (and re-sizes that box to the turned orientation). The two compose -- an image can stand a sideways scan upright with "rotate" and tilt the whole frame with "angle". Same meaning as the watermark's "angle". A STRING is a data token, so each record of a template turns its own sticker: "{{s.angle}}" takes the record's number (or text of one, such as a CSV field); a record without it leaves the block upright, silently, and a value that is not a number is reported with the block's pointer (and the -merge record) and drawn upright. A string with no token is read as it stands.
bleed	number		Step and Repeat: how far this placed block's box reaches past its TRIM on every side, in points -- the box IS the bleed box, and the trim is the box inset by this much. A design tool sizes a card by it (the trim plus twice the bleed) and draws the trim line inside each copy; the engine draws nothing from it, and keeps the bleed only through the block's own clip. Only a block with position has a trim: on any other block the key is refused with a notice naming the block.
blend	string		How this block's OWN ink combines with the page under it -- the sixteen PDF blend modes (ISO 32000-1 11.3.5), written as /BM. On a container that means its background, border and stripe, never its children; for the subtree as one object see "groupBlend". Case and separators are folded, so "colorDodge", "color-dodge" and "COLORDODGE" are one mode. An unrecognized name is refused with a warning and the block paints normally. See the blend-modes sample. One of: normal, multiply, screen, overlay, darken, lighten, colorDodge, colorBurn, hardLight, softLight, difference, exclusion, hue, saturation, ...
breakBefore	boolean		Pagination: open a new page before this block - a part divider, an appendix. Ignored when the page is already empty, so it never leaves a blank page behind.
changed	boolean		Revision / change bar: draw a vertical bar in the page margin beside this block's content to flag it as changed. A document-level changeBars object customizes the bar (width / color / side / distance); a condition tag declared with changeBar:true sets this automatically for tagged content.

Key	Type	Default	Description
`clip`	`boolean`		Clip this block's children (on a container) or its own ink to its box, with the style's corner "radius". Only bites where content is BIGGER than the box: a flow container grows to fit its children vertically, so give it "position" with a height, or a "minHeight", before expecting a vertical crop. Container-scoped, like "groupOpacity": on any other flow block it is refused with a warning. For a crop to a SHAPE rather than to the box, see "clipPath"; giving both INTERSECTS them.
`clipPath`	`object`		Crop this block to an arbitrary SHAPE stated inside its box -- a design tool's FRAME. Same vocabulary as the "shape" block and read by the same outline builder, so a shape and a clipPath given identical parameters describe the identical outline. Scoped exactly as "clip" is: a positioned block of any type (it needs a "position.height", because the shape is stated inside the box) or a container in the flow; anywhere else it is refused with a warning. Naming "clip" as well INTERSECTS the two. The shape turns, scales and skews with the block that carries it.
`condition`	`string`		Conditional text: render this block only when its condition-tag expression is shown under the document's conditions map. Whitespace/comma tag terms, each optionally '!'-negated, AND-combined; an unlisted tag defaults to shown. Empty means always shown.
`float`	`string`		Float this block out of the stacked flow to one side of its column; content that FOLLOWS it wraps into the space beside it. Works on ANY block type -- a table, a container, a chart, a code listing, a figure. "center" straddles the page content width, so text in every column wraps around it. The float box takes the block's own "width"; a block that declares none takes HALF the column, so a float always leaves room beside it. Ignored on a block that also sets "position" or "anchor" -- those have already chosen their placement. Not to be confused with "anchor": { "float": true }, a BOOLEAN about migrating an anchored frame to the next column that can hold it. One of: `left`, `right`, `center`.
`groupBlend`	`string`		Composite this block AND its children once, then blend the RESULT with the page. NOT the same key as "blend", and the difference is larger than it is for opacity: two stacked children that each multiply the backdrop multiply it TWICE, which is darker than the group multiplying once, and no per-child setting reproduces it. That is what a design tool's blend-mode control on a group means. Container-scoped: on any other flow block it is refused with a warning. Also the way to blend an element together with its drop shadow, which is drawn as stacked alpha layers and so keeps its own normal compositing. See the blend-modes sample. One of: `normal`, `multiply`, `screen`, `overlay`, `darken`, `lighten`, `colorDodge`, `colorBurn`, `hardLight`, `softLight`, `difference`, `exclusion`, `hue`, `saturation`, ...
`groupOpacity`	`number`		Composite this block AND its children once, then fade the result -- true GROUP opacity. NOT the same key as "opacity", which fades each element as it paints: two overlapping opaque children at opacity 0.5 each show a seam where they cross, and under groupOpacity 0.5 they do not. The two compose. Container-scoped: on any other flow block it is refused with a warning. See the clipped-groups sample.

Key	Type	Default	Description
`height`	`number`		Blank vertical space to insert, in points.
`mirrorH`	`boolean | string`		Mirror the whole block left-to-right about the centre of its "position" box, under the same restriction as "angle". Identical to scale x -1, and the two compose. NOT "flipH": that key belongs to the image block, where it mirrors the PICTURE INSIDE its frame. The two are separate words so an image can do both. A STRING is a data token, so each record of a template mirrors its own sticker: "{{s.flip}}" takes the record's boolean, or text "true" / "false" in any case, or 1 / 0 (a number or text); a record without it draws the block unmirrored, silently, and any other value ("yes" included) is reported with the block's pointer (and the -merge record) and drawn unmirrored. A string with no token is read as it stands.
`mirrorV`	`boolean | string`		Mirror the whole block top-to-bottom about the centre of its "position" box, under the same restriction as "angle". Identical to scale y -1. NOT "flipV" -- see "mirrorH". A STRING is a data token, so each record of a template mirrors its own sticker: "{{s.flip}}" takes the record's boolean, or text "true" / "false" in any case, or 1 / 0 (a number or text); a record without it draws the block unmirrored, silently, and any other value ("yes" included) is reported with the block's pointer (and the -merge record) and drawn unmirrored. A string with no token is read as it stands.
`opacity`	`number`		Constant opacity for this block: 0 (invisible) to 1 (opaque, the default). REAL alpha -- a /CA /ca ExtGState in the PDF -- so a translucent panel over a photograph shows the photograph through it. Not to be confused with "style.background" {color, alpha}, which composites the tint toward WHITE at parse time and so only looks right over a white page. PER-ELEMENT, NOT GROUP: on a container it paints that container's own decoration -- background, border, accent bar, stripe -- and never its children, which draw at their own opacity (1 unless they set one). Two translucent shapes that overlap therefore composite twice, which is what per-element opacity means; group opacity, where a whole subtree is composited once and then faded, is a separate feature. Where a draw already carries its own alpha (an SVG fill-opacity, a gradient's alpha) the two MULTIPLY.

Key	Type	Default	Description
`repeatAt`	`array of object`		Imposition: draw this block ONCE and place the drawing again at each of these offsets, in points from the block's own position, y DOWNWARD as every `position` is. One entry = one extra copy, so a list of 7 puts 8 cards on a sheet. The drawing is captured into a Form XObject and each offset is a `cm` + `Do` of that one stream, so n copies cost one copy of the content. Translation only -- `angle` / `scale` / `skew` are not applied per placement. A REPEATED FORM MAY NOT CARRY TAGGED CONTENT: ISO 14289-1 7.20 allows a form's content ONE semantic parent, so repeating tagged text (or a shape with `alt`) fails PDF/UA -- veraPDF catches it, ap_addstruct's audit does not. The engine warns and draws it anyway. Repeat decoration: a shape or svg with no `alt`, or put the block in a master's `frames`, which is drawn as a page artifact. Only a block with `position` is drawn again: on a block without one (in the flow, or an anchored frame), or on a repeat / if / include block, the key is refused with a notice naming the block. Each copy takes the block's `groupOpacity` and `groupBlend`. The layout map (-map) reports the copies as `repeats` on the block's fragment, so a design tool can select and outline them.
`scale`	`number | array of number | object | string`		Scale the block about the same pivot as "angle" (the centre of its "position" box), and under the same restriction. A CONTAINER in the flow may also carry this: it becomes a GROUP that turns about its own measured box and reserves the bounding box of the turned result, and its children inherit the transform. Any other flow block ignores it with a warning -- wrap it in a container. A number scales both axes; [x, y] or {"x":, "y":} scales them separately. 1 = natural size. Applied AFTER layout, so a scaled block does NOT re-wrap its text at the scaled measure -- this is a design tool's scale handle, not a change of font size. A STRING is a data token, so each record of a template scales its own sticker: "{{s.scale}}" takes the record's number, [x, y] or {"x":, "y":} (members may be text of numbers), or text of one or two plain numbers such as a CSV field ("1.5", "1.5 0.8", "[1.5, 0.8]"); a record without it draws the block at its natural size, silently, and a value that is not a number or a pair is reported with the block's pointer (and the -merge record) and drawn at natural size. A string with no token is read as it stands.
`skew`	`number | array of number | object | string`		Shear the block, in DEGREES, about the same pivot as "angle", under the same restriction. [x, y] or {"x":, "y":}; a bare number skews x only. Positive x leans the top to the right and positive y lifts the right-hand side -- both counter-clockwise, matching "angle". Clamped to +/-89.9: a 90-degree skew is a singular matrix and draws nothing. A STRING is a data token, so each record of a template skews its own sticker: "{{s.skew}}" takes the record's number (x only), [x, y] or {"x":, "y":}, or text of one or two plain numbers; clamped as the written one is. A record without it draws the block unskewed, silently, and a value that is not a number or a pair is reported with the block's pointer (and the -merge record) and drawn unskewed. A string with no token is read as it stands.

Key	Type	Default	Description
straddle	boolean		Straddle across columns: when true and the body is multi-column, this block spans the FULL content width and text in every column wraps around it (above and below) - a spanning float generalized from center-floated images to any block. No effect in a single-column body, on a positioned / anchored block, or on an image/pdf already using float:"center" (the same mechanism). Does not balance the columns above it or act as a section divider.
verticalAlign	string		Where the content sits in its "position" box when it is shorter than the box's height: "top" (the default), "middle" ("center" is the same) or "bottom". A paragraph, heading or pre moves as a whole (its own box with it); a container keeps its box where it is and moves its children inside it (a child with its own "position" stays put). The offset is the box height less the content's height as it is actually placed -- the same lines, shrinkToFit size and padding that are then drawn -- halved for middle. Content as tall as the box or taller does not move: it is clipped and reported exactly as without the key. Needs a position with a height: on a block without one, on any other kind of block, on text set along a textPath "d" curve (stated in fractions of the whole box, so drawn across all of it; the "arc" form aligns), and on vertical text whose columns fill the box (columns run the full height unless a paragraph's or a heading's boxHeight is shorter), it is refused with a notice and ignored. The layout map keeps reporting the box. One of: top, middle, bottom, center.
z	number		CANVAS MODE ONLY (see the top-level "pages"): where this element sits in the PAINT order of its page. Lower draws first, further back; unset means "wherever I am in the list", and ties keep author order. It moves the PAINT and not the READING: the structure tree keeps the document order, so a caption written first because it is read first can still be painted last because it sits on top. That separation is what a flattened scene graph cannot express. Ignored outside canvas mode, where the flow already fixes both orders.

The rule element

Type name: rule.

A horizontal line, with a thickness, color and inset from the column edges.

Example

```
{
  "type": "rule",
  "thickness": 0.5,
  "color": "#cccccc",
  "inset": 0
}
```

Keys

Key	Type	Default	Description
anchor	anchor		-

Key	Type	Default	Description
angle	number | string		Turn the block, in degrees counter-clockwise, about the CENTRE of its "position" box. A CONTAINER in the flow may also carry this: it becomes a GROUP that turns about its own measured box and reserves the bounding box of the turned result, and its children inherit the transform. Any other flow block ignores it with a warning -- wrap it in a container. The block lays out upright inside its box and the finished result is turned, so a "position.height" that truncates content truncates the same text it would unrotated, and the turned result is NOT clipped to its box: it may extend past it, exactly as an unrotated positioned block may. Not to be confused with the image block's "rotate", which is an integer QUARTER TURN CLOCKWISE applied to the picture INSIDE its box (and re-sizes that box to the turned orientation). The two compose -- an image can stand a sideways scan upright with "rotate" and tilt the whole frame with "angle". Same meaning as the watermark's "angle". A STRING is a data token, so each record of a template turns its own sticker: "{{s.angle}}" takes the record's number (or text of one, such as a CSV field); a record without it leaves the block upright, silently, and a value that is not a number is reported with the block's pointer (and the -merge record) and drawn upright. A string with no token is read as it stands.
bleed	number		Step and Repeat: how far this placed block's box reaches past its TRIM on every side, in points -- the box IS the bleed box, and the trim is the box inset by this much. A design tool sizes a card by it (the trim plus twice the bleed) and draws the trim line inside each copy; the engine draws nothing from it, and keeps the bleed only through the block's own clip. Only a block with position has a trim: on any other block the key is refused with a notice naming the block.
blend	string		How this block's OWN ink combines with the page under it -- the sixteen PDF blend modes (ISO 32000-1 11.3.5), written as /BM. On a container that means its background, border and stripe, never its children; for the subtree as one object see "groupBlend". Case and separators are folded, so "colorDodge", "color-dodge" and "COLORDODGE" are one mode. An unrecognized name is refused with a warning and the block paints normally. See the blend-modes sample. One of: normal, multiply, screen, overlay, darken, lighten, colorDodge, colorBurn, hardLight, softLight, difference, exclusion, hue, saturation, ...
breakBefore	boolean		Pagination: open a new page before this block - a part divider, an appendix. Ignored when the page is already empty, so it never leaves a blank page behind.
changed	boolean		Revision / change bar: draw a vertical bar in the page margin beside this block's content to flag it as changed. A document-level changeBars object customizes the bar (width / color / side / distance); a condition tag declared with changeBar:true sets this automatically for tagged content.

Key	Type	Default	Description
`clip`	`boolean`		Clip this block's children (on a container) or its own ink to its box, with the style's corner "radius". Only bites where content is BIGGER than the box: a flow container grows to fit its children vertically, so give it "position" with a height, or a "minHeight", before expecting a vertical crop. Container-scoped, like "groupOpacity": on any other flow block it is refused with a warning. For a crop to a SHAPE rather than to the box, see "clipPath"; giving both INTERSECTS them.
`clipPath`	`object`		Crop this block to an arbitrary SHAPE stated inside its box -- a design tool's FRAME. Same vocabulary as the "shape" block and read by the same outline builder, so a shape and a clipPath given identical parameters describe the identical outline. Scoped exactly as "clip" is: a positioned block of any type (it needs a "position.height", because the shape is stated inside the box) or a container in the flow; anywhere else it is refused with a warning. Naming "clip" as well INTERSECTS the two. The shape turns, scales and skews with the block that carries it.
`color`	`color`		Line color.
`condition`	`string`		Conditional text: render this block only when its condition-tag expression is shown under the document's conditions map. Whitespace/comma tag terms, each optionally '!'-negated, AND-combined; an unlisted tag defaults to shown. Empty means always shown.
`float`	`string`		Float this block out of the stacked flow to one side of its column; content that FOLLOWS it wraps into the space beside it. Works on ANY block type -- a table, a container, a chart, a code listing, a figure. "center" straddles the page content width, so text in every column wraps around it. The float box takes the block's own "width"; a block that declares none takes HALF the column, so a float always leaves room beside it. Ignored on a block that also sets "position" or "anchor" -- those have already chosen their placement. Not to be confused with "anchor": { "float": true }, a BOOLEAN about migrating an anchored frame to the next column that can hold it. One of: `left`, `right`, `center`.
`groupBlend`	`string`		Composite this block AND its children once, then blend the RESULT with the page. NOT the same key as "blend", and the difference is larger than it is for opacity: two stacked children that each multiply the backdrop multiply it TWICE, which is darker than the group multiplying once, and no per-child setting reproduces it. That is what a design tool's blend-mode control on a group means. Container-scoped: on any other flow block it is refused with a warning. Also the way to blend an element together with its drop shadow, which is drawn as stacked alpha layers and so keeps its own normal compositing. See the blend-modes sample. One of: `normal`, `multiply`, `screen`, `overlay`, `darken`, `lighten`, `colorDodge`, `colorBurn`, `hardLight`, `softLight`, `difference`, `exclusion`, `hue`, `saturation`, ...

Key	Type	Default	Description
groupOpacity	number		Composite this block AND its children once, then fade the result -- true GROUP opacity. NOT the same key as "opacity", which fades each element as it paints: two overlapping opaque children at opacity 0.5 each show a seam where they cross, and under groupOpacity 0.5 they do not. The two compose. Container-scoped: on any other flow block it is refused with a warning. See the clipped-groups sample.
inset	number		Inset from both column edges in points, so the rule is shorter than the full measure.
mirrorH	boolean | string		Mirror the whole block left-to-right about the centre of its "position" box, under the same restriction as "angle". Identical to scale x -1, and the two compose. NOT "flipH": that key belongs to the image block, where it mirrors the PICTURE INSIDE its frame. The two are separate words so an image can do both. A STRING is a data token, so each record of a template mirrors its own sticker: "{{s.flip}}" takes the record's boolean, or text "true" / "false" in any case, or 1 / 0 (a number or text); a record without it draws the block unmirrored, silently, and any other value ("yes" included) is reported with the block's pointer (and the -merge record) and drawn unmirrored. A string with no token is read as it stands.
mirrorV	boolean | string		Mirror the whole block top-to-bottom about the centre of its "position" box, under the same restriction as "angle". Identical to scale y -1. NOT "flipV" -- see "mirrorH". A STRING is a data token, so each record of a template mirrors its own sticker: "{{s.flip}}" takes the record's boolean, or text "true" / "false" in any case, or 1 / 0 (a number or text); a record without it draws the block unmirrored, silently, and any other value ("yes" included) is reported with the block's pointer (and the -merge record) and drawn unmirrored. A string with no token is read as it stands.
opacity	number		Constant opacity for this block: 0 (invisible) to 1 (opaque, the default). REAL alpha -- a /CA /ca ExtGState in the PDF -- so a translucent panel over a photograph shows the photograph through it. Not to be confused with "style.background" {color, alpha}, which composites the tint toward WHITE at parse time and so only looks right over a white page. PER-ELEMENT, NOT GROUP: on a container it paints that container's own decoration -- background, border, accent bar, stripe -- and never its children, which draw at their own opacity (1 unless they set one). Two translucent shapes that overlap therefore composite twice, which is what per-element opacity means; group opacity, where a whole subtree is composited once and then faded, is a separate feature. Where a draw already carries its own alpha (an SVG fill-opacity, a gradient's alpha) the two MULTIPLY.

Key	Type	Default	Description
repeatAt	array of object		Imposition: draw this block ONCE and place the drawing again at each of these offsets, in points from the block's own position, y DOWNWARD as every position is. One entry = one extra copy, so a list of 7 puts 8 cards on a sheet. The drawing is captured into a Form XObject and each offset is a cm + Do of that one stream, so n copies cost one copy of the content. Translation only -- angle / scale / skew are not applied per placement. A REPEATED FORM MAY NOT CARRY TAGGED CONTENT: ISO 14289-1 7.20 allows a form's content ONE semantic parent, so repeating tagged text (or a shape with alt) fails PDF/UA -- veraPDF catches it, ap_addstruct's audit does not. The engine warns and draws it anyway. Repeat decoration: a shape or svg with no alt, or put the block in a master's frames, which is drawn as a page artifact. Only a block with position is drawn again: on a block without one (in the flow, or an anchored frame), or on a repeat / if / include block, the key is refused with a notice naming the block. Each copy takes the block's groupOpacity and groupBlend. The layout map (-map) reports the copies as repeats on the block's fragment, so a design tool can select and outline them.
scale	number | array of number | object | string		Scale the block about the same pivot as "angle" (the centre of its "position" box), and under the same restriction. A CONTAINER in the flow may also carry this: it becomes a GROUP that turns about its own measured box and reserves the bounding box of the turned result, and its children inherit the transform. Any other flow block ignores it with a warning -- wrap it in a container. A number scales both axes; [x, y] or {"x":, "y":} scales them separately. 1 = natural size. Applied AFTER layout, so a scaled block does NOT re-wrap its text at the scaled measure -- this is a design tool's scale handle, not a change of font size. A STRING is a data token, so each record of a template scales its own sticker: "{{s.scale}}" takes the record's number, [x, y] or {"x":, "y":} (members may be text of numbers), or text of one or two plain numbers such as a CSV field ("1.5", "1.5 0.8", "[1.5, 0.8]"); a record without it draws the block at its natural size, silently, and a value that is not a number or a pair is reported with the block's pointer (and the -merge record) and drawn at natural size. A string with no token is read as it stands.
skew	number | array of number | object | string		Shear the block, in DEGREES, about the same pivot as "angle", under the same restriction. [x, y] or {"x":, "y":}; a bare number skews x only. Positive x leans the top to the right and positive y lifts the right-hand side -- both counter-clockwise, matching "angle". Clamped to +/-89.9: a 90-degree skew is a singular matrix and draws nothing. A STRING is a data token, so each record of a template skews its own sticker: "{{s.skew}}" takes the record's number (x only), [x, y] or {"x":, "y":}, or text of one or two plain numbers; clamped as the written one is. A record without it draws the block unskewed, silently, and a value that is not a number or a pair is reported with the block's pointer (and the -merge record) and drawn unskewed. A string with no token is read as it stands.

Key	Type	Default	Description
straddle	boolean		Straddle across columns: when true and the body is multi-column, this block spans the FULL content width and text in every column wraps around it (above and below) - a spanning float generalized from center-floated images to any block. No effect in a single-column body, on a positioned / anchored block, or on an image/pdf already using float:"center" (the same mechanism). Does not balance the columns above it or act as a section divider.
thickness	number		Line thickness in points.
verticalAlign	string		Where the content sits in its "position" box when it is shorter than the box's height: "top" (the default), "middle" ("center" is the same) or "bottom". A paragraph, heading or pre moves as a whole (its own box with it); a container keeps its box where it is and moves its children inside it (a child with its own "position" stays put). The offset is the box height less the content's height as it is actually placed -- the same lines, shrinkToFit size and padding that are then drawn -- halved for middle. Content as tall as the box or taller does not move: it is clipped and reported exactly as without the key. Needs a position with a height: on a block without one, on any other kind of block, on text set along a textPath "d" curve (stated in fractions of the whole box, so drawn across all of it; the "arc" form aligns), and on vertical text whose columns fill the box (columns run the full height unless a paragraph's or a heading's boxHeight is shorter), it is refused with a notice and ignored. The layout map keeps reporting the box. One of: top, middle, bottom, center.
z	number		CANVAS MODE ONLY (see the top-level "pages"): where this element sits in the PAINT order of its page. Lower draws first, further back; unset means "wherever I am in the list", and ties keep author order. It moves the PAINT and not the READING: the structure tree keeps the document order, so a caption written first because it is read first can still be painted last because it sits on top. That separation is what a flattened scene graph cannot express. Ignored outside canvas mode, where the flow already fixes both orders.

The pagebreak element

Type name: pagebreak. Accepted aliases: page-break.

Forces the following content onto a fresh page. Skipped when the cursor is already at the top of a page, so it never leaves a blank sheet behind.

Example

```
{
  "type": "pagebreak"
}
```

Keys

Key	Type	Default	Description
`breakBefore`	`boolean`		Pagination: open a new page before this block - a part divider, an appendix. Ignored when the page is already empty, so it never leaves a blank page behind.
`opacity`	`number`		Constant opacity for this block: 0 (invisible) to 1 (opaque, the default). REAL alpha -- a /CA /ca ExtGState in the PDF -- so a translucent panel over a photograph shows the photograph through it. Not to be confused with "style.background" {color, alpha}, which composites the tint toward WHITE at parse time and so only looks right over a white page. PER-ELEMENT, NOT GROUP: on a container it paints that container's own decoration -- background, border, accent bar, stripe -- and never its children, which draw at their own opacity (1 unless they set one). Two translucent shapes that overlap therefore composite twice, which is what per-element opacity means; group opacity, where a whole subtree is composited once and then faded, is a separate feature. Where a draw already carries its own alpha (an SVG fill-opacity, a gradient's alpha) the two MULTIPLY.

7. Tables

The tabular grid, its rows, cells and headers.

The table element

Type name: `table`.

A tabular grid with sized columns, header rows that repeat across page breaks, spanning cells, per-cell styling and full tagged-PDF structure. Cells hold blocks, so a cell can contain paragraphs, images, lists or a nested table. A long table splits across pages - and, when a single row is taller than a page, within a row.

Example

```
{
  "type": "table",
  "caption": "Quarterly totals.",
  "headerRows": 1,
  "headerRepeat": true,
  "columns": [
    {
      "width": "60%"
    },
    {
      "width": "40%",
      "align": "right"
    }
  ],
  "rows": [
    {
      "header": true,
      "cells": [
        "Quarter",
        "Units"
      ]
    },
    {
      "cells": [
        "Q1",
        "1,240"
      ]
    }
  ]
}
```

A spanning header row and one boxed total cell

```
{
  "type": "table",
  "headerRows": 1,
  "border": {
    "width": 0.5,
    "color": "#999999"
  },
  "columns": [
    {
      "width": "50%"
    },
    {
      "width": "25%",
      "align": "right"
    },
    {
      "width": "25%",
      "align": "right"
    }
  ],
  "rows": [
    {
      "header": true,
      "cells": [
        {
          "colspan": 3,
          "style": {
            "background": "#e8edf4"
          },
          "blocks": [
            {
              "type": "paragraph",
              "text": "Regional results"
            }
          ]
        }
      ]
    },
    {
      "cells": [
        {
          "blocks": [
            {
              "type": "paragraph",
              "text": "North"
            }
          ]
        },
        {
          "blocks": [
            {
              "type": "paragraph",
              "text": "112"
            }
          ]
        },
        {
```

```
          "blocks": [
            {
              "type": "paragraph",
              "text": "100"
            }
          ]
        }
      ]
    },
    {
      "cells": [
        {
          "blocks": [
            {
              "type": "paragraph",
              "text": "Total"
            }
          ]
        },
        {
          "style": {
            "border": {
              "width": 2,
              "color": "#1f4e79"
            }
          },
          "blocks": [
            {
              "type": "paragraph",
              "text": "314"
            }
          ]
        },
        {
          "blocks": [
            {
              "type": "paragraph",
              "text": "310"
            }
          ]
        }
      ]
    }
  ]
}
```

Keys

Key	Type	Default	Description
anchor	anchor		-

Key	Type	Default	Description
angle	number | string		Turn the block, in degrees counter-clockwise, about the CENTRE of its "position" box. A CONTAINER in the flow may also carry this: it becomes a GROUP that turns about its own measured box and reserves the bounding box of the turned result, and its children inherit the transform. Any other flow block ignores it with a warning -- wrap it in a container. The block lays out upright inside its box and the finished result is turned, so a "position.height" that truncates content truncates the same text it would unrotated, and the turned result is NOT clipped to its box: it may extend past it, exactly as an unrotated positioned block may. Not to be confused with the image block's "rotate", which is an integer QUARTER TURN CLOCKWISE applied to the picture INSIDE its box (and re-sizes that box to the turned orientation). The two compose -- an image can stand a sideways scan upright with "rotate" and tilt the whole frame with "angle". Same meaning as the watermark's "angle". A STRING is a data token, so each record of a template turns its own sticker: "{{s.angle}}" takes the record's number (or text of one, such as a CSV field); a record without it leaves the block upright, silently, and a value that is not a number is reported with the block's pointer (and the -merge record) and drawn upright. A string with no token is read as it stands.
bleed	number		Step and Repeat: how far this placed block's box reaches past its TRIM on every side, in points -- the box IS the bleed box, and the trim is the box inset by this much. A design tool sizes a card by it (the trim plus twice the bleed) and draws the trim line inside each copy; the engine draws nothing from it, and keeps the bleed only through the block's own clip. Only a block with position has a trim: on any other block the key is refused with a notice naming the block.
blend	string		How this block's OWN ink combines with the page under it -- the sixteen PDF blend modes (ISO 32000-1 11.3.5), written as /BM. On a container that means its background, border and stripe, never its children; for the subtree as one object see "groupBlend". Case and separators are folded, so "colorDodge", "color-dodge" and "COLORDODGE" are one mode. An unrecognized name is refused with a warning and the block paints normally. See the blend-modes sample. One of: normal, multiply, screen, overlay, darken, lighten, colorDodge, colorBurn, hardLight, softLight, difference, exclusion, hue, saturation, ...
border	border	{'width': 0.5, 'color': '#000000'}	Border around the table as a whole.
breakBefore	boolean		Pagination: open a new page before this block - a part divider, an appendix. Ignored when the page is already empty, so it never leaves a blank page behind.

Key	Type	Default	Description
caption	string | runs		A title drawn above the table and tagged <Caption>. Accepts a plain string, an ARRAY OF RUNS, or a SINGLE RUN OBJECT - the same run forms a paragraph's "runs" takes - so a caption can carry per-span styling, and above all a per-span "fontFamily": a caption that mixes scripts sets each span in the face that suits it instead of flattening to one. The flat text is always derived from the runs and stays the entry of record: it is what reaches the list of figures, the /Alt and the structure tree. A number ("Figure 3-2.") is prepended as the first run, unstyled, so it takes the block's own face.
cellPadding	insets		Padding inside every cell. A single cell can override it with its own "padding".
cellVAlign	string		Default vertical alignment of content within each row's height. One of: top, middle, center, bottom.
changed	boolean		Revision / change bar: draw a vertical bar in the page margin beside this block's content to flag it as changed. A document-level changeBars object customizes the bar (width / color / side / distance); a condition tag declared with changeBar:true sets this automatically for tagged content.
clip	boolean		Clip this block's children (on a container) or its own ink to its box, with the style's corner "radius". Only bites where content is BIGGER than the box: a flow container grows to fit its children vertically, so give it "position" with a height, or a "minHeight", before expecting a vertical crop. Container-scoped, like "groupOpacity": on any other flow block it is refused with a warning. For a crop to a SHAPE rather than to the box, see "clipPath"; giving both INTERSECTS them.
clipPath	object		Crop this block to an arbitrary SHAPE stated inside its box -- a design tool's FRAME. Same vocabulary as the "shape" block and read by the same outline builder, so a shape and a clipPath given identical parameters describe the identical outline. Scoped exactly as "clip" is: a positioned block of any type (it needs a "position.height", because the shape is stated inside the box) or a container in the flow; anywhere else it is refused with a warning. Naming "clip" as well INTERSECTS the two. The shape turns, scales and skews with the block that carries it.
colRules	border	{'width': 0.5, 'color': '#000000'}	Vertical rules drawn between columns.
columns	array of dimension | object		Column widths: dimensions, or { width } objects.
condition	string		Conditional text: render this block only when its condition-tag expression is shown under the document's conditions map. Whitespace/comma tag terms, each optionally '!'-negated, AND-combined; an unlisted tag defaults to shown. Empty means always shown.

Key	Type	Default	Description
`float`	`string`		Float this block out of the stacked flow to one side of its column; content that FOLLOWS it wraps into the space beside it. Works on ANY block type -- a table, a container, a chart, a code listing, a figure. "center" straddles the page content width, so text in every column wraps around it. The float box takes the block's own "width"; a block that declares none takes HALF the column, so a float always leaves room beside it. Ignored on a block that also sets "position" or "anchor" -- those have already chosen their placement. Not to be confused with "anchor": { "float": true }, a BOOLEAN about migrating an anchored frame to the next column that can hold it. One of: `left`, `right`, `center`.
`footerRows`	`integer`		The last N rows form the footer (tagged <TFoot>), drawn at the table end.
`frame`	`border`	{'width': 1.0, 'color': '#000000'}	Outer box drawn around the table on each page it occupies. (On a pdf-contact-sheet "frame" is a boolean instead - a different key with the same name.)
`groupBlend`	`string`		Composite this block AND its children once, then blend the RESULT with the page. NOT the same key as "blend", and the difference is larger than it is for opacity: two stacked children that each multiply the backdrop multiply it TWICE, which is darker than the group multiplying once, and no per-child setting reproduces it. That is what a design tool's blend-mode control on a group means. Container-scoped: on any other flow block it is refused with a warning. Also the way to blend an element together with its drop shadow, which is drawn as stacked alpha layers and so keeps its own normal compositing. See the blend-modes sample. One of: `normal`, `multiply`, `screen`, `overlay`, `darken`, `lighten`, `colorDodge`, `colorBurn`, `hardLight`, `softLight`, `difference`, `exclusion`, `hue`, `saturation`, ...
`groupOpacity`	`number`		Composite this block AND its children once, then fade the result -- true GROUP opacity. NOT the same key as "opacity", which fades each element as it paints: two overlapping opaque children at opacity 0.5 each show a seam where they cross, and under groupOpacity 0.5 they do not. The two compose. Container-scoped: on any other flow block it is refused with a warning. See the clipped-groups sample.
`headerCols`	`integer`		The first N columns are row headers (tagged TH with Scope=Row). With `rowGroups` this counts the AUTHORED columns that are row headers -- the synthetic group columns are added in front of them and are always headers, so `headerCols` need not account for them.
`headerRepeat`	`boolean`		Repeat the header rows after a page break (default true).
`headerRows`	`integer`		How many rows from the top are header rows. They repeat at the top of every page the table splits across, and are tagged as header cells.
`headerRule`	`border`	{'width': 1.0, 'color': '#000000'}	The rule drawn under the header rows, so it can be heavier than the body rules.

Key	Type	Default	Description
mirrorH	boolean | string		Mirror the whole block left-to-right about the centre of its "position" box, under the same restriction as "angle". Identical to scale x -1, and the two compose. NOT "flipH": that key belongs to the image block, where it mirrors the PICTURE INSIDE its frame. The two are separate words so an image can do both. A STRING is a data token, so each record of a template mirrors its own sticker: "{{s.flip}}" takes the record's boolean, or text "true" / "false" in any case, or 1 / 0 (a number or text); a record without it draws the block unmirrored, silently, and any other value ("yes" included) is reported with the block's pointer (and the -merge record) and drawn unmirrored. A string with no token is read as it stands.
mirrorV	boolean | string		Mirror the whole block top-to-bottom about the centre of its "position" box, under the same restriction as "angle". Identical to scale y -1. NOT "flipV" -- see "mirrorH". A STRING is a data token, so each record of a template mirrors its own sticker: "{{s.flip}}" takes the record's boolean, or text "true" / "false" in any case, or 1 / 0 (a number or text); a record without it draws the block unmirrored, silently, and any other value ("yes" included) is reported with the block's pointer (and the -merge record) and drawn unmirrored. A string with no token is read as it stands.
number	string		Pin a literal number/label for this block. Does not advance the counter.
numbered	boolean	true	Take part in automatic caption numbering when the document enables it. false leaves this one unnumbered and does not advance the counter.
opacity	number		Constant opacity for this block: 0 (invisible) to 1 (opaque, the default). REAL alpha -- a /CA /ca ExtGState in the PDF -- so a translucent panel over a photograph shows the photograph through it. Not to be confused with "style.background" {color, alpha}, which composites the tint toward WHITE at parse time and so only looks right over a white page. PER-ELEMENT, NOT GROUP: on a container it paints that container's own decoration -- background, border, accent bar, stripe -- and never its children, which draw at their own opacity (1 unless they set one). Two translucent shapes that overlap therefore composite twice, which is what per-element opacity means; group opacity, where a whole subtree is composited once and then faded, is a separate feature. Where a draw already carries its own alpha (an SVG fill-opacity, a gradient's alpha) the two MULTIPLY.
presentational	boolean		This table is a LAYOUT grid, not data (default false). It keeps its geometry but emits no table structure at all - no Table, THead, TBody, TR, TH or TD - and each cell's content tags itself in reading order. Use it when the grid only positions things: a form whose cells hold bare fields with no text, or a row of independent captioned fields where none heads another. Declaring headerRows/headerCols on such a table invents a relationship that is not there, which misleads a screen reader worse than saying nothing; leaving them off trips WCAG 1.3.1 forever. This is the third answer, and it exempts the table from that check.

Key	Type	Default	Description
repeatAt	array of object		Imposition: draw this block ONCE and place the drawing again at each of these offsets, in points from the block's own position, y DOWNWARD as every position is. One entry = one extra copy, so a list of 7 puts 8 cards on a sheet. The drawing is captured into a Form XObject and each offset is a cm + Do of that one stream, so n copies cost one copy of the content. Translation only -- angle / scale / skew are not applied per placement. A REPEATED FORM MAY NOT CARRY TAGGED CONTENT: ISO 14289-1 7.20 allows a form's content ONE semantic parent, so repeating tagged text (or a shape with alt) fails PDF/UA -- veraPDF catches it, ap_addstruct's audit does not. The engine warns and draws it anyway. Repeat decoration: a shape or svg with no alt, or put the block in a master's frames, which is drawn as a page artifact. Only a block with position is drawn again: on a block without one (in the flow, or an anchored frame), or on a repeat / if / include block, the key is refused with a notice naming the block. Each copy takes the block's groupOpacity and groupBlend. The layout map (-map) reports the copies as repeats on the block's fragment, so a design tool can select and outline them.
rowGroups	object		Tag a HIERARCHICAL table the way a professional 508 remediator tags one. The ground truth is a remediation of this engine's own budget-schedule sample, and with this key the engine reproduces that tree: ONE <Table> across every page the table spans (no THead/TBody; the repeated column band on later pages is an artifact), each group row drawn where it is but tagged as a rowspanned <TH Scope=Row> at the head of its group's first <TR>, the first headerCols drawn columns as <TH Scope=Row>, a total row's label widened over the synthetic columns it still sits under, and a derived /Headers on every cell (the row header immediately to its left, its column header). Mark the rows with group and total. Absent = every table tags exactly as before.
rowRules	border	{'width': 0.5, 'color': '#000000'}	Horizontal rules drawn between body rows. The header boundary is drawn by "headerRule" instead.
rows	array of array of tableCell | object		The table's rows. Each is a list of cells - a bare string for plain text, or a cell object for spans, styling or block content.

Key	Type	Default	Description
`scale`	`number | array of number | object | string`		Scale the block about the same pivot as "angle" (the centre of its "position" box), and under the same restriction. A CONTAINER in the flow may also carry this: it becomes a GROUP that turns about its own measured box and reserves the bounding box of the turned result, and its children inherit the transform. Any other flow block ignores it with a warning -- wrap it in a container. A number scales both axes; [x, y] or {"x":, "y":} scales them separately. 1 = natural size. Applied AFTER layout, so a scaled block does NOT re-wrap its text at the scaled measure -- this is a design tool's scale handle, not a change of font size. A STRING is a data token, so each record of a template scales its own sticker: "{{s.scale}}" takes the record's number, [x, y] or {"x":, "y":} (members may be text of numbers), or text of one or two plain numbers such as a CSV field ("1.5", "1.5 0.8", "[1.5, 0.8]"); a record without it draws the block at its natural size, silently, and a value that is not a number or a pair is reported with the block's pointer (and the -merge record) and drawn at natural size. A string with no token is read as it stands.
`skew`	`number | array of number | object | string`		Shear the block, in DEGREES, about the same pivot as "angle", under the same restriction. [x, y] or {"x":, "y":}; a bare number skews x only. Positive x leans the top to the right and positive y lifts the right-hand side -- both counter-clockwise, matching "angle". Clamped to +/-89.9: a 90-degree skew is a singular matrix and draws nothing. A STRING is a data token, so each record of a template skews its own sticker: "{{s.skew}}" takes the record's number (x only), [x, y] or {"x":, "y":}, or text of one or two plain numbers; clamped as the written one is. A record without it draws the block unskewed, silently, and a value that is not a number or a pair is reported with the block's pointer (and the -merge record) and drawn unskewed. A string with no token is read as it stands.
`straddle`	`boolean`		Straddle across columns: when true and the body is multi-column, this block spans the FULL content width and text in every column wraps around it (above and below) - a spanning float generalized from center-floated images to any block. No effect in a single-column body, on a positioned / anchored block, or on an image/pdf already using float:"center" (the same mechanism). Does not balance the columns above it or act as a section divider.
`stripeColor`	`color`		Zebra striping: fill alternating body rows with this color.
`tableAlign`	`string`		Horizontal placement of the table box when width is narrower than the content. One of: `left`, `center`, `right`.

Key	Type	Default	Description
`verticalAlign`	`string`		Where the content sits in its "position" box when it is shorter than the box's height: "top" (the default), "middle" ("center" is the same) or "bottom". A paragraph, heading or pre moves as a whole (its own box with it); a container keeps its box where it is and moves its children inside it (a child with its own "position" stays put). The offset is the box height less the content's height as it is actually placed -- the same lines, shrinkToFit size and padding that are then drawn -- halved for middle. Content as tall as the box or taller does not move: it is clipped and reported exactly as without the key. Needs a position with a height: on a block without one, on any other kind of block, on text set along a textPath "d" curve (stated in fractions of the whole box, so drawn across all of it; the "arc" form aligns), and on vertical text whose columns fill the box (columns run the full height unless a paragraph's or a heading's boxHeight is shorter), it is refused with a notice and ignored. The layout map keeps reporting the box. One of: `top`, `middle`, `bottom`, `center`.
`width`	`dimension`		Explicit table width (default Auto = the full content width).
`z`	`number`		CANVAS MODE ONLY (see the top-level "pages"): where this element sits in the PAINT order of its page. Lower draws first, further back; unset means "wherever I am in the list", and ties keep author order. It moves the PAINT and not the READING: the structure tree keeps the document order, so a caption written first because it is read first can still be painted last because it sits on top. That separation is what a flattened scene graph cannot express. Ignored outside canvas mode, where the flow already fixes both orders.

8. Pictures and drawings

Raster images, vector art, and generated graphics.

The image element

Type name: `image`.

A raster picture - PNG, JPEG, GIF, TIFF or WebP, or HEIC/HEIF on macOS. Size it with `width` and `height`, cap it with `maxHeight`, or let it take its natural size; the file's own orientation and ICC profile are honored. Floating an image with `float` lets text wrap beside it. A `caption` is drawn beneath and collected into the list of figures; `alt` supplies the text an assistive reader announces, and is required for a document that must pass PDF/UA.

Example

```
{
  "type": "image",
  "src": "diagram.png",
  "width": "45%",
  "float": "right",
  "caption": "The airflow around the fin root.",
  "alt": "A fin with airflow lines curving around its root fillet."
}
```

A left-floated, captioned figure with alt text

```
{
  "type": "image",
  "src": "fin.png",
  "width": "40%",
  "float": "left",
  "caption": "The fin planform.",
  "alt": "A trapezoidal fin, its root chord longer than its tip."
}
```

A landscape scan turned upright (rotate 90) into a portrait box

```
{
  "type": "image",
  "src": "scan.png",
  "width": 120,
  "rotate": 90,
  "caption": "The fold-out, turned upright.",
  "alt": "A wide schematic rotated a quarter-turn clockwise to read upright."
}
```

Keys

Key	Type	Default	Description
`alt`	`string`		Alternate text (/Alt on the tagged Figure).
`anchor`	`anchor`		-
`angle`	`number | string`		Turn the block, in degrees counter-clockwise, about the CENTRE of its "position" box. A CONTAINER in the flow may also carry this: it becomes a GROUP that turns about its own measured box and reserves the bounding box of the turned result, and its children inherit the transform. Any other flow block ignores it with a warning -- wrap it in a container. The block lays out upright inside its box and the finished result is turned, so a "position.height" that truncates content truncates the same text it would unrotated, and the turned result is NOT clipped to its box: it may extend past it, exactly as an unrotated positioned block may. Not to be confused with the image block's "rotate", which is an integer QUARTER TURN CLOCKWISE applied to the picture INSIDE its box (and re-sizes that box to the turned orientation). The two compose -- an image can stand a sideways scan upright with "rotate" and tilt the whole frame with "angle". Same meaning as the watermark's "angle". A STRING is a data token, so each record of a template turns its own sticker: "{{s.angle}}" takes the record's number (or text of one, such as a CSV field); a record without it leaves the block upright, silently, and a value that is not a number is reported with the block's pointer (and the -merge record) and drawn upright. A string with no token is read as it stands.
`autoOrient`	`boolean`	true	Honor the file's own orientation (JPEG EXIF Orientation, TIFF tag 274, HEIF irot/imir) before rotate and flip, so a photo stored sideways is drawn and sized upright. false draws the pixels as stored -- for a document that already turned the picture with rotate.

Key	Type	Default	Description
`bleed`	`number`		Step and Repeat: how far this placed block's box reaches past its TRIM on every side, in points -- the box IS the bleed box, and the trim is the box inset by this much. A design tool sizes a card by it (the trim plus twice the bleed) and draws the trim line inside each copy; the engine draws nothing from it, and keeps the bleed only through the block's own `clip`. Only a block with `position` has a trim: on any other block the key is refused with a notice naming the block.
`blend`	`string`		How this block's OWN ink combines with the page under it -- the sixteen PDF blend modes (ISO 32000-1 11.3.5), written as /BM. On a container that means its background, border and stripe, never its children; for the subtree as one object see "groupBlend". Case and separators are folded, so "colorDodge", "color-dodge" and "COLORDODGE" are one mode. An unrecognized name is refused with a warning and the block paints normally. See the blend-modes sample. One of: `normal`, `multiply`, `screen`, `overlay`, `darken`, `lighten`, `colorDodge`, `colorBurn`, `hardLight`, `softLight`, `difference`, `exclusion`, `hue`, `saturation`, ...
`blur`	`number`		Gaussian blur applied to the image's PIXELS, as a standard deviation in POINTS. The same quantity SVG's feGaussianBlur calls stdDeviation and CSS's filter:blur() takes, so an author who knows either already knows this one. 0 (the default) is a sharp image and costs nothing. STATED IN POINTS, NOT PIXELS, DELIBERATELY. The blur is applied to the raster that gets embedded, and how many pixels that raster has depends on the maxPpi downsample policy and on the box the picture was placed in. A radius in pixels would therefore soften by a visibly different amount when the same photograph moved into a smaller frame. THE PICTURE STAYS INSIDE ITS BOX. Edge pixels are replicated outward rather than fading to transparent, so a blurred photograph keeps its rectangle instead of growing a vignette; the blur does not spread beyond the box the way a CSS filter does. A soft-edged PNG keeps its softness -- the colour is premultiplied by the alpha plane before blurring, so transparent pixels contribute nothing and no dark halo appears at the edge.
`border`	`number | color | object`		Image effect: a border stroked around the (rounded) edge - a width, a color, or { width, color }.
`breakBefore`	`boolean`		Pagination: open a new page before this block - a part divider, an appendix. Ignored when the page is already empty, so it never leaves a blank page behind.

Key	Type	Default	Description
caption	string | runs		Caption printed under the image. A caption is also what enrolls the image in the Figure counter and gives a cross-reference something to resolve against. Accepts a plain string, an ARRAY OF RUNS, or a SINGLE RUN OBJECT - the same run forms a paragraph's "runs" takes - so a caption can carry per-span styling, and above all a per-span "fontFamily": a caption that mixes scripts sets each span in the face that suits it instead of flattening to one. The flat text is always derived from the runs and stays the entry of record: it is what reaches the list of figures, the /Alt and the structure tree. A number ("Figure 3-2.") is prepended as the first run, unstyled, so it takes the block's own face.
changed	boolean		Revision / change bar: draw a vertical bar in the page margin beside this block's content to flag it as changed. A document-level changeBars object customizes the bar (width / color / side / distance); a condition tag declared with changeBar:true sets this automatically for tagged content.
clip	boolean		Clip this block's children (on a container) or its own ink to its box, with the style's corner "radius". Only bites where content is BIGGER than the box: a flow container grows to fit its children vertically, so give it "position" with a height, or a "minHeight", before expecting a vertical crop. Container-scoped, like "groupOpacity": on any other flow block it is refused with a warning. For a crop to a SHAPE rather than to the box, see "clipPath"; giving both INTERSECTS them.
clipPath	object		Crop this block to an arbitrary SHAPE stated inside its box -- a design tool's FRAME. Same vocabulary as the "shape" block and read by the same outline builder, so a shape and a clipPath given identical parameters describe the identical outline. Scoped exactly as "clip" is: a positioned block of any type (it needs a "position.height", because the shape is stated inside the box) or a container in the flow; anywhere else it is refused with a warning. Naming "clip" as well INTERSECTS the two. The shape turns, scales and skews with the block that carries it.
condition	string		Conditional text: render this block only when its condition-tag expression is shown under the document's conditions map. Whitespace/comma tag terms, each optionally '!'-negated, AND-combined; an unlisted tag defaults to shown. Empty means always shown.
cornerRadius	number		Image effect: clip the image to a rounded rectangle of this radius (points).
fit	string		How the image fills its box: "contain" fits it whole, "cover" crops to fill, "fill" stretches, "none" draws it at its natural size - one pixel per point. "cover" and "none" are cropped to the box; the picture is centred in it unless "focus" aims the crop (and "zoom" magnifies a cover). One of: contain, cover, fill, none.
flipH	boolean		Mirror the image left-to-right within its box.
flipV	boolean		Mirror the image top-to-bottom within its box.

Key	Type	Default	Description
`float`	`string`		Float this block out of the stacked flow to one side of its column; content that FOLLOWS it wraps into the space beside it. Works on ANY block type -- a table, a container, a chart, a code listing, a figure. "center" straddles the page content width, so text in every column wraps around it. The float box takes the block's own "width"; a block that declares none takes HALF the column, so a float always leaves room beside it. Ignored on a block that also sets "position" or "anchor" -- those have already chosen their placement. Not to be confused with "anchor": { "float": true }, a BOOLEAN about migrating an anchored frame to the next column that can hold it. One of: `left`, `right`, `center`.
`focus`	`string | number | array of number | object`		Where a cropping fit aims: [x, y] (or {"x":, "y":}, or one number for both axes) as fractions of the photo AS IT STANDS UPRIGHT -- after the file's own EXIF / TIFF / HEIF orientation (see autoOrient), before rotate, flipH and flipV, which carry the point along with the picture -- 0 = left / top, 1 = right / bottom. The point is brought to the centre of the box as far as the picture allows; the crop never pulls away from an edge, so [0, 0] and [1, 1] reach the corners and the default [0.5, 0.5] is the centred crop. Only "cover" and "none" crop; on any other fit it is ignored with a warning. Values outside 0..1 are clamped. A STRING is a data token, so each record of a template aims its own crop: "{{cards.0.frontFocus}}" resolves to the whole value at that path -- a number, [x, y], {x, y}, or text of numbers such as "0.3 0.4" (a CSV field) -- and a field the record lacks leaves the centred crop, silently. A bound value that is not numbers, or outside 0..1, is reported with the block's pointer (and the -merge record). A string with no token is read as it stands.
`groupBlend`	`string`		Composite this block AND its children once, then blend the RESULT with the page. NOT the same key as "blend", and the difference is larger than it is for opacity: two stacked children that each multiply the backdrop multiply it TWICE, which is darker than the group multiplying once, and no per-child setting reproduces it. That is what a design tool's blend-mode control on a group means. Container-scoped: on any other flow block it is refused with a warning. Also the way to blend an element together with its drop shadow, which is drawn as stacked alpha layers and so keeps its own normal compositing. See the blend-modes sample. One of: `normal`, `multiply`, `screen`, `overlay`, `darken`, `lighten`, `colorDodge`, `colorBurn`, `hardLight`, `softLight`, `difference`, `exclusion`, `hue`, `saturation`, ...
`groupOpacity`	`number`		Composite this block AND its children once, then fade the result -- true GROUP opacity. NOT the same key as "opacity", which fades each element as it paints: two overlapping opaque children at opacity 0.5 each show a seam where they cross, and under groupOpacity 0.5 they do not. The two compose. Container-scoped: on any other flow block it is refused with a warning. See the clipped-groups sample.
`height`	`dimension`		Drawn height - points, a percentage, or "auto" to keep the aspect ratio.
`link`	`string`		Clickable image: a web URI.
`linkFile`	`string`		Clickable image: open another PDF (GoToR; a relative path resolves next to the produced document).

Key	Type	Default	Description
`linkPage`	`integer`		1-based page in linkFile.
`maxHeight`	`number`		Cap on the drawn height in points, applied after scaling, so a tall image cannot overrun the page.
`mirrorH`	`boolean | string`		Mirror the whole block left-to-right about the centre of its "position" box, under the same restriction as "angle". Identical to scale x -1, and the two compose. NOT "flipH": that key belongs to the image block, where it mirrors the PICTURE INSIDE its frame. The two are separate words so an image can do both. A STRING is a data token, so each record of a template mirrors its own sticker: "{{s.flip}}" takes the record's boolean, or text "true" / "false" in any case, or 1 / 0 (a number or text); a record without it draws the block unmirrored, silently, and any other value ("yes" included) is reported with the block's pointer (and the -merge record) and drawn unmirrored. A string with no token is read as it stands.
`mirrorV`	`boolean | string`		Mirror the whole block top-to-bottom about the centre of its "position" box, under the same restriction as "angle". Identical to scale y -1. NOT "flipV" -- see "mirrorH". A STRING is a data token, so each record of a template mirrors its own sticker: "{{s.flip}}" takes the record's boolean, or text "true" / "false" in any case, or 1 / 0 (a number or text); a record without it draws the block unmirrored, silently, and any other value ("yes" included) is reported with the block's pointer (and the -merge record) and drawn unmirrored. A string with no token is read as it stands.
`number`	`string`		Pin a literal number/label for this block. Does not advance the counter.
`numbered`	`boolean`	true	Take part in automatic caption numbering when the document enables it. false leaves this one unnumbered and does not advance the counter.
`opacity`	`number`		Constant opacity for this block: 0 (invisible) to 1 (opaque, the default). REAL alpha -- a /CA /ca ExtGState in the PDF -- so a translucent panel over a photograph shows the photograph through it. Not to be confused with "style.background" {color, alpha}, which composites the tint toward WHITE at parse time and so only looks right over a white page. PER-ELEMENT, NOT GROUP: on a container it paints that container's own decoration -- background, border, accent bar, stripe -- and never its children, which draw at their own opacity (1 unless they set one). Two translucent shapes that overlap therefore composite twice, which is what per-element opacity means; group opacity, where a whole subtree is composited once and then faded, is a separate feature. Where a draw already carries its own alpha (an SVG fill-opacity, a gradient's alpha) the two MULTIPLY.
`radius`	`number`		Alias for cornerRadius.
`renderingIntent`	`renderingIntent`		This picture's own rendering intent, in place of the document's `renderingIntent` -- typically "perceptual" for a photograph on a page of relative-colorimetric brand color. In cmyk color mode its RGB is converted at this intent; and when it differs from the document's it is written as the image's /Intent, which a RIP or viewer converting the file uses for this image alone.

Key	Type	Default	Description
repeatAt	array of object		Imposition: draw this block ONCE and place the drawing again at each of these offsets, in points from the block's own position, y DOWNWARD as every position is. One entry = one extra copy, so a list of 7 puts 8 cards on a sheet. The drawing is captured into a Form XObject and each offset is a cm + Do of that one stream, so n copies cost one copy of the content. Translation only -- angle / scale / skew are not applied per placement. A REPEATED FORM MAY NOT CARRY TAGGED CONTENT: ISO 14289-1 7.20 allows a form's content ONE semantic parent, so repeating tagged text (or a shape with alt) fails PDF/UA -- veraPDF catches it, ap_addstruct's audit does not. The engine warns and draws it anyway. Repeat decoration: a shape or svg with no alt, or put the block in a master's frames, which is drawn as a page artifact. Only a block with position is drawn again: on a block without one (in the flow, or an anchored frame), or on a repeat / if / include block, the key is refused with a notice naming the block. Each copy takes the block's groupOpacity and groupBlend. The layout map (-map) reports the copies as repeats on the block's fragment, so a design tool can select and outline them.
rotate	integer		Rotate the image within its box: 0, 90, 180 or 270 degrees clockwise (other values snap to the nearest quarter turn). Applied after the file's own EXIF / TIFF / HEIF orientation, so it turns the picture as a viewer shows it upright. The box is sized to the displayed orientation, so a 90/270 rotation with a plain width auto-sizes to a portrait box and fills it undistorted.
scale	number | array of number | object | string		Scale the block about the same pivot as "angle" (the centre of its "position" box), and under the same restriction. A CONTAINER in the flow may also carry this: it becomes a GROUP that turns about its own measured box and reserves the bounding box of the turned result, and its children inherit the transform. Any other flow block ignores it with a warning -- wrap it in a container. A number scales both axes; [x, y] or {"x":, "y":} scales them separately. 1 = natural size. Applied AFTER layout, so a scaled block does NOT re-wrap its text at the scaled measure -- this is a design tool's scale handle, not a change of font size. A STRING is a data token, so each record of a template scales its own sticker: "{{s.scale}}" takes the record's number, [x, y] or {"x":, "y":} (members may be text of numbers), or text of one or two plain numbers such as a CSV field ("1.5", "1.5 0.8", "[1.5, 0.8]"); a record without it draws the block at its natural size, silently, and a value that is not a number or a pair is reported with the block's pointer (and the -merge record) and drawn at natural size. A string with no token is read as it stands.
shadow	shadow		-

Key	Type	Default	Description
skew	number | array of number | object | string		Shear the block, in DEGREES, about the same pivot as "angle", under the same restriction. [x, y] or {"x":, "y":}; a bare number skews x only. Positive x leans the top to the right and positive y lifts the right-hand side -- both counter-clockwise, matching "angle". Clamped to +/-89.9: a 90-degree skew is a singular matrix and draws nothing. A STRING is a data token, so each record of a template skews its own sticker: "{{s.skew}}" takes the record's number (x only), [x, y] or {"x":, "y":}, or text of one or two plain numbers; clamped as the written one is. A record without it draws the block unskewed, silently, and a value that is not a number or a pair is reported with the block's pointer (and the -merge record) and drawn unskewed. A string with no token is read as it stands.
src	string		PNG, JPEG (incl. CMYK JPEG, passed through), GIF, TIFF, WebP, or HEIC/HEIF (incl. AVIF; macOS builds only, since macOS decodes them -- other builds draw a placeholder box and say so). Relative paths resolve against the spec file. The file's own orientation (JPEG EXIF Orientation, TIFF tag 274, HEIF irot/imir) is honored: a phone photo stored sideways is drawn upright and sized as it stands, with no re-encode (see autoOrient). An ICC profile embedded in the file (PNG iCCP, JPEG APP2, TIFF, WebP ICCP; for HEIC/HEIF, the color space macOS decodes it in, e.g. Display P3) is kept: the image is written in that ICCBased color space, and in colorMode "cmyk" (or pdfx) the profile is instead the source of the image's conversion to CMYK. A profile that cannot be used (damaged, not version 2 to 4, not Gray/RGB/CMYK, or not matching the image) is reported and ignored. See images.colorProfiles.
straddle	boolean		Straddle across columns: when true and the body is multi-column, this block spans the FULL content width and text in every column wraps around it (above and below) - a spanning float generalized from center-floated images to any block. No effect in a single-column body, on a positioned / anchored block, or on an image/pdf already using float:"center" (the same mechanism). Does not balance the columns above it or act as a section divider.
verticalAlign	string		Where the content sits in its "position" box when it is shorter than the box's height: "top" (the default), "middle" ("center" is the same) or "bottom". A paragraph, heading or pre moves as a whole (its own box with it); a container keeps its box where it is and moves its children inside it (a child with its own "position" stays put). The offset is the box height less the content's height as it is actually placed -- the same lines, shrinkToFit size and padding that are then drawn -- halved for middle. Content as tall as the box or taller does not move: it is clipped and reported exactly as without the key. Needs a position with a height: on a block without one, on any other kind of block, on text set along a textPath "d" curve (stated in fractions of the whole box, so drawn across all of it; the "arc" form aligns), and on vertical text whose columns fill the box (columns run the full height unless a paragraph's or a heading's boxHeight is shorter), it is refused with a notice and ignored. The layout map keeps reporting the box. One of: top, middle, bottom, center.
width	dimension		Drawn width - points, a percentage of the column, or "auto".

Key	Type	Default	Description
z	number		CANVAS MODE ONLY (see the top-level "pages"): where this element sits in the PAINT order of its page. Lower draws first, further back; unset means "wherever I am in the list", and ties keep author order. It moves the PAINT and not the READING: the structure tree keeps the document order, so a caption written first because it is read first can still be painted last because it sits on top. That separation is what a flattened scene graph cannot express. Ignored outside canvas mode, where the flow already fixes both orders.
zoom	number | string	1	"cover" only: magnify the picture past the size that just covers the box, keeping "focus" in view (1 = plain cover, 2 = twice as close). The box does not change; only more of the picture is cropped away, so the pixels are spread over more of the page. On any other fit it is ignored with a warning. A STRING is a data token, as for focus: "{{cards.0.frontZoom}}" takes the record's number (or text of one); a field the record lacks leaves plain cover, and a value that is not a number, or below 1, is reported.

The shape element

Type name: `shape`.

Vector geometry drawn directly: a rectangle, line, ellipse, polygon or an arbitrary bezier `path`. Fills accept flat color or a gradient, and strokes accept width, color and dash. Use shapes for rules, callout pointers, diagram primitives and decorative panels.

Example

```
{
  "type": "shape",
  "shape": "rect",
  "width": 200,
  "height": 40,
  "fill": {
    "type": "linear",
    "angle": 0,
    "stops": [
      {
        "offset": 0,
        "color": "#1f4e79"
      },
      {
        "offset": 1,
        "color": "#7ba44f"
      }
    ]
  }
}
```

Keys

Key	Type	Default	Description
alt	string		Alternate text. A shape draws as a decorative artifact by default; giving it alt makes it a tagged <Figure> carrying this text, the same rule svg and math follow. Added 2026-08-30 -- before that a shape could not be described at all.
anchor	anchor		-
angle	number | string		Turn the block, in degrees counter-clockwise, about the CENTRE of its "position" box. A CONTAINER in the flow may also carry this: it becomes a GROUP that turns about its own measured box and reserves the bounding box of the turned result, and its children inherit the transform. Any other flow block ignores it with a warning -- wrap it in a container. The block lays out upright inside its box and the finished result is turned, so a "position.height" that truncates content truncates the same text it would unrotated, and the turned result is NOT clipped to its box: it may extend past it, exactly as an unrotated positioned block may. Not to be confused with the image block's "rotate", which is an integer QUARTER TURN CLOCKWISE applied to the picture INSIDE its box (and re-sizes that box to the turned orientation). The two compose -- an image can stand a sideways scan upright with "rotate" and tilt the whole frame with "angle". Same meaning as the watermark's "angle". A STRING is a data token, so each record of a template turns its own sticker: "{{s.angle}}" takes the record's number (or text of one, such as a CSV field); a record without it leaves the block upright, silently, and a value that is not a number is reported with the block's pointer (and the -merge record) and drawn upright. A string with no token is read as it stands.
arrow	string		Draw a filled arrowhead at one or both ends. A LINE or a PATH only -- a closed shape has no ends, and naming it there is refused with a warning rather than silently ignored. The head is turned to the path's own direction where it lands (for a curve, its tangent) and takes the stroke color. One of: none, start, end, both.
arrowSize	number		Arrowhead length in points. 0 (the default) derives it from "strokeWidth", which keeps the head in proportion when the line weight changes.
bleed	number		Step and Repeat: how far this placed block's box reaches past its TRIM on every side, in points -- the box IS the bleed box, and the trim is the box inset by this much. A design tool sizes a card by it (the trim plus twice the bleed) and draws the trim line inside each copy; the engine draws nothing from it, and keeps the bleed only through the block's own clip. Only a block with position has a trim: on any other block the key is refused with a notice naming the block.
blend	string		How this block's OWN ink combines with the page under it -- the sixteen PDF blend modes (ISO 32000-1 11.3.5), written as /BM. On a container that means its background, border and stripe, never its children; for the subtree as one object see "groupBlend". Case and separators are folded, so "colorDodge", "color-dodge" and "COLORDODGE" are one mode. An unrecognized name is refused with a warning and the block paints normally. See the blend-modes sample. One of: normal, multiply, screen, overlay, darken, lighten, colorDodge, colorBurn, hardLight, softLight, difference, exclusion, hue, saturation, ...

Key	Type	Default	Description
breakBefore	boolean		Pagination: open a new page before this block - a part divider, an appendix. Ignored when the page is already empty, so it never leaves a blank page behind.
changed	boolean		Revision / change bar: draw a vertical bar in the page margin beside this block's content to flag it as changed. A document-level changeBars object customizes the bar (width / color / side / distance); a condition tag declared with changeBar:true sets this automatically for tagged content.
clip	boolean		Clip this block's children (on a container) or its own ink to its box, with the style's corner "radius". Only bites where content is BIGGER than the box: a flow container grows to fit its children vertically, so give it "position" with a height, or a "minHeight", before expecting a vertical crop. Container-scoped, like "groupOpacity": on any other flow block it is refused with a warning. For a crop to a SHAPE rather than to the box, see "clipPath"; giving both INTERSECTS them.
clipPath	object		Crop this block to an arbitrary SHAPE stated inside its box -- a design tool's FRAME. Same vocabulary as the "shape" block and read by the same outline builder, so a shape and a clipPath given identical parameters describe the identical outline. Scoped exactly as "clip" is: a positioned block of any type (it needs a "position.height", because the shape is stated inside the box) or a container in the flow; anywhere else it is refused with a warning. Naming "clip" as well INTERSECTS the two. The shape turns, scales and skews with the block that carries it.
condition	string		Conditional text: render this block only when its condition-tag expression is shown under the document's conditions map. Whitespace/comma tag terms, each optionally '!'-negated, AND-combined; an unlisted tag defaults to shown. Empty means always shown.
d	string | array of object		Path data (shape: path): an SVG path string (M/L/H/V/C/Q/Z) or an op array (["M",x,y], ["C",x1,y1,x2,y2,x,y], ["Q",x1,y1,x,y], ["L",x,y], ["Z"]). Coordinates are in viewBox space.
dash	array of number | number		Dash pattern: alternating on/off run lengths in points, repeating. A single number means equal on and off. Omit or [] for a solid line. A pattern whose lengths are all zero is refused with a warning and the line draws solid -- an all-zero array is a division by zero in most renderers and an invisible line in the rest.
dashPhase	number		How far INTO the dash pattern the line starts, in points. Shifts the rhythm without changing it -- two parallel dashed rules with different phases read as a weave rather than as a grid.
fill	color | gradient		Solid color, or a gradient object. Ignored for a line.

Key	Type	Default	Description
`float`	`string`		Float this block out of the stacked flow to one side of its column; content that FOLLOWS it wraps into the space beside it. Works on ANY block type -- a table, a container, a chart, a code listing, a figure. "center" straddles the page content width, so text in every column wraps around it. The float box takes the block's own "width"; a block that declares none takes HALF the column, so a float always leaves room beside it. Ignored on a block that also sets "position" or "anchor" -- those have already chosen their placement. Not to be confused with "anchor": { "float": true }, a BOOLEAN about migrating an anchored frame to the next column that can hold it. One of: `left`, `right`, `center`.
`groupBlend`	`string`		Composite this block AND its children once, then blend the RESULT with the page. NOT the same key as "blend", and the difference is larger than it is for opacity: two stacked children that each multiply the backdrop multiply it TWICE, which is darker than the group multiplying once, and no per-child setting reproduces it. That is what a design tool's blend-mode control on a group means. Container-scoped: on any other flow block it is refused with a warning. Also the way to blend an element together with its drop shadow, which is drawn as stacked alpha layers and so keeps its own normal compositing. See the blend-modes sample. One of: `normal`, `multiply`, `screen`, `overlay`, `darken`, `lighten`, `colorDodge`, `colorBurn`, `hardLight`, `softLight`, `difference`, `exclusion`, `hue`, `saturation`, ...
`groupOpacity`	`number`		Composite this block AND its children once, then fade the result -- true GROUP opacity. NOT the same key as "opacity", which fades each element as it paints: two overlapping opaque children at opacity 0.5 each show a seam where they cross, and under groupOpacity 0.5 they do not. The two compose. Container-scoped: on any other flow block it is refused with a warning. See the clipped-groups sample.
`height`	`number`		Flow box height in points (an absolute shape uses position.h).
`lineCap`	`string`		The shape of a stroke's free ends, and of each dash. "butt" (the default) stops square at the endpoint; "round" and "square" both extend half the stroke width past it. A dotted line is a dash of [0, gap] with a ROUND cap -- with butt ends the same pattern draws nothing at all. One of: `butt`, `round`, `square`.
`lineJoin`	`string`		The shape of a stroke's corners. "miter" (the default) extends both edges to a point, "round" arcs them, "bevel" cuts the point off. Only meaningful where one PATH turns a corner: a polygon, a polyline, a path, or a rectangle's border. One of: `miter`, `round`, `bevel`.

Key	Type	Default	Description
mirrorH	boolean | string		Mirror the whole block left-to-right about the centre of its "position" box, under the same restriction as "angle". Identical to scale x -1, and the two compose. NOT "flipH": that key belongs to the image block, where it mirrors the PICTURE INSIDE its frame. The two are separate words so an image can do both. A STRING is a data token, so each record of a template mirrors its own sticker: "{{s.flip}}" takes the record's boolean, or text "true" / "false" in any case, or 1 / 0 (a number or text); a record without it draws the block unmirrored, silently, and any other value ("yes" included) is reported with the block's pointer (and the -merge record) and drawn unmirrored. A string with no token is read as it stands.
mirrorV	boolean | string		Mirror the whole block top-to-bottom about the centre of its "position" box, under the same restriction as "angle". Identical to scale y -1. NOT "flipV" -- see "mirrorH". A STRING is a data token, so each record of a template mirrors its own sticker: "{{s.flip}}" takes the record's boolean, or text "true" / "false" in any case, or 1 / 0 (a number or text); a record without it draws the block unmirrored, silently, and any other value ("yes" included) is reported with the block's pointer (and the -merge record) and drawn unmirrored. A string with no token is read as it stands.
miterLimit	number		How far a mitered corner may spike before it is cut to a bevel, as a multiple of the stroke width. PDF's default is 10 (SVG's too). A sharp angle at a heavy weight can throw a spike many times the line width, which is what this bounds.
opacity	number		Constant opacity for this block: 0 (invisible) to 1 (opaque, the default). REAL alpha -- a /CA /ca ExtGState in the PDF -- so a translucent panel over a photograph shows the photograph through it. Not to be confused with "style.background" {color, alpha}, which composites the tint toward WHITE at parse time and so only looks right over a white page. PER-ELEMENT, NOT GROUP: on a container it paints that container's own decoration -- background, border, accent bar, stripe -- and never its children, which draw at their own opacity (1 unless they set one). Two translucent shapes that overlap therefore composite twice, which is what per-element opacity means; group opacity, where a whole subtree is composited once and then faded, is a separate feature. Where a draw already carries its own alpha (an SVG fill-opacity, a gradient's alpha) the two MULTIPLY.
points	array of array of number		Line / polygon vertices as [x, y] fractions 0..1 of the box. OUT OF RANGE IS ALMOST ALWAYS viewBox UNITS OR POINTS BY MISTAKE, and it fails silently: the value is MULTIPLIED by the box, so [[0, 0], [100, 0]] in a 300pt-wide box draws a line 30000pt long -- about forty-nine pages off the right edge of a Letter sheet -- while reading as an ordinary line block in the source. The range is bounded here so the schema check names the block instead.

Key	Type	Default	Description
repeatAt	array of object		Imposition: draw this block ONCE and place the drawing again at each of these offsets, in points from the block's own position, y DOWNWARD as every position is. One entry = one extra copy, so a list of 7 puts 8 cards on a sheet. The drawing is captured into a Form XObject and each offset is a cm + Do of that one stream, so n copies cost one copy of the content. Translation only -- angle / scale / skew are not applied per placement. A REPEATED FORM MAY NOT CARRY TAGGED CONTENT: ISO 14289-1 7.20 allows a form's content ONE semantic parent, so repeating tagged text (or a shape with alt) fails PDF/UA -- veraPDF catches it, ap_addstruct's audit does not. The engine warns and draws it anyway. Repeat decoration: a shape or svg with no alt, or put the block in a master's frames, which is drawn as a page artifact. Only a block with position is drawn again: on a block without one (in the flow, or an anchored frame), or on a repeat / if / include block, the key is refused with a notice naming the block. Each copy takes the block's groupOpacity and groupBlend. The layout map (-map) reports the copies as repeats on the block's fragment, so a design tool can select and outline them.
scale	number | array of number | object | string		Scale the block about the same pivot as "angle" (the centre of its "position" box), and under the same restriction. A CONTAINER in the flow may also carry this: it becomes a GROUP that turns about its own measured box and reserves the bounding box of the turned result, and its children inherit the transform. Any other flow block ignores it with a warning -- wrap it in a container. A number scales both axes; [x, y] or {"x":, "y":} scales them separately. 1 = natural size. Applied AFTER layout, so a scaled block does NOT re-wrap its text at the scaled measure -- this is a design tool's scale handle, not a change of font size. A STRING is a data token, so each record of a template scales its own sticker: "{{s.scale}}" takes the record's number, [x, y] or {"x":, "y":} (members may be text of numbers), or text of one or two plain numbers such as a CSV field ("1.5", "1.5 0.8", "[1.5, 0.8]"); a record without it draws the block at its natural size, silently, and a value that is not a number or a pair is reported with the block's pointer (and the -merge record) and drawn at natural size. A string with no token is read as it stands.
shadow	shadow		-
shape	string		Shape kind (default rect). One of: rect, line, ellipse, circle, polygon, path.
skew	number | array of number | object | string		Shear the block, in DEGREES, about the same pivot as "angle", under the same restriction. [x, y] or {"x":, "y":}; a bare number skews x only. Positive x leans the top to the right and positive y lifts the right-hand side -- both counter-clockwise, matching "angle". Clamped to +/-89.9: a 90-degree skew is a singular matrix and draws nothing. A STRING is a data token, so each record of a template skews its own sticker: "{{s.skew}}" takes the record's number (x only), [x, y] or {"x":, "y":}, or text of one or two plain numbers; clamped as the written one is. A record without it draws the block unskewed, silently, and a value that is not a number or a pair is reported with the block's pointer (and the -merge record) and drawn unskewed. A string with no token is read as it stands.

Key	Type	Default	Description
straddle	boolean		Straddle across columns: when true and the body is multi-column, this block spans the FULL content width and text in every column wraps around it (above and below) - a spanning float generalized from center-floated images to any block. No effect in a single-column body, on a positioned / anchored block, or on an image/pdf already using float:"center" (the same mechanism). Does not balance the columns above it or act as a section divider.
stroke	color		Outline color. Omit for no outline.
strokeWidth	number		Outline thickness in points.
verticalAlign	string		Where the content sits in its "position" box when it is shorter than the box's height: "top" (the default), "middle" ("center" is the same) or "bottom". A paragraph, heading or pre moves as a whole (its own box with it); a container keeps its box where it is and moves its children inside it (a child with its own "position" stays put). The offset is the box height less the content's height as it is actually placed -- the same lines, shrinkToFit size and padding that are then drawn -- halved for middle. Content as tall as the box or taller does not move: it is clipped and reported exactly as without the key. Needs a position with a height: on a block without one, on any other kind of block, on text set along a textPath "d" curve (stated in fractions of the whole box, so drawn across all of it; the "arc" form aligns), and on vertical text whose columns fill the box (columns run the full height unless a paragraph's or a heading's boxHeight is shorter), it is refused with a notice and ignored. The layout map keeps reporting the box. One of: top, middle, bottom, center.
viewBox	array of number		Path coordinate space [x, y, w, h] (or [w, h]) mapped to the box; default 0 0 100 100.
width	dimension		Drawn width - points, a percentage of the column, or "auto".
z	number		CANVAS MODE ONLY (see the top-level "pages"): where this element sits in the PAINT order of its page. Lower draws first, further back; unset means "wherever I am in the list", and ties keep author order. It moves the PAINT and not the READING: the structure tree keeps the document order, so a caption written first because it is read first can still be painted last because it sits on top. That separation is what a flattened scene graph cannot express. Ignored outside canvas mode, where the flow already fixes both orders.

The svg element

Type name: svg.

An imported SVG drawing, rendered as real PDF vector art rather than a picture of one - so it stays sharp at any zoom and prints at full device resolution. A broad practical slice is supported: groups and nested transforms, every basic shape and the full path grammar, solid/gradient paint with per-stop opacity, a CSS <style> sheet, a <defs> library reused through <use>/<symbol>, multi-line <text>/<tspan>, and compositing - clip-path, <mask> (even-odd knockout), <pattern> tiling, a filter drop shadow, and the root viewport clip. Raster filters (a blur of content, color matrices) have no vector form and are skipped.

Example

```
{
  "type": "svg",
  "src": "logo.svg",
  "width": 120
}
```

Keys

Key	Type	Default	Description
`alt`	`string`		Alternate text; when set, the drawing is tagged as a Figure (else a decorative artifact).
`anchor`	`anchor`		-
`angle`	`number | string`		Turn the block, in degrees counter-clockwise, about the CENTRE of its "position" box. A CONTAINER in the flow may also carry this: it becomes a GROUP that turns about its own measured box and reserves the bounding box of the turned result, and its children inherit the transform. Any other flow block ignores it with a warning -- wrap it in a container. The block lays out upright inside its box and the finished result is turned, so a "position.height" that truncates content truncates the same text it would unrotated, and the turned result is NOT clipped to its box: it may extend past it, exactly as an unrotated positioned block may. Not to be confused with the image block's "rotate", which is an integer QUARTER TURN CLOCKWISE applied to the picture INSIDE its box (and re-sizes that box to the turned orientation). The two compose -- an image can stand a sideways scan upright with "rotate" and tilt the whole frame with "angle". Same meaning as the watermark's "angle". A STRING is a data token, so each record of a template turns its own sticker: "{{s.angle}}" takes the record's number (or text of one, such as a CSV field); a record without it leaves the block upright, silently, and a value that is not a number is reported with the block's pointer (and the -merge record) and drawn upright. A string with no token is read as it stands.
`bleed`	`number`		Step and Repeat: how far this placed block's box reaches past its TRIM on every side, in points -- the box IS the bleed box, and the trim is the box inset by this much. A design tool sizes a card by it (the trim plus twice the bleed) and draws the trim line inside each copy; the engine draws nothing from it, and keeps the bleed only through the block's own `clip`. Only a block with `position` has a trim: on any other block the key is refused with a notice naming the block.

Key	Type	Default	Description
`blend`	`string`		How this block's OWN ink combines with the page under it -- the sixteen PDF blend modes (ISO 32000-1 11.3.5), written as /BM. On a container that means its background, border and stripe, never its children; for the subtree as one object see "groupBlend". Case and separators are folded, so "colorDodge", "color-dodge" and "COLORDODGE" are one mode. An unrecognized name is refused with a warning and the block paints normally. See the blend-modes sample. One of: `normal`, `multiply`, `screen`, `overlay`, `darken`, `lighten`, `colorDodge`, `colorBurn`, `hardLight`, `softLight`, `difference`, `exclusion`, `hue`, `saturation`, ...
`breakBefore`	`boolean`		Pagination: open a new page before this block - a part divider, an appendix. Ignored when the page is already empty, so it never leaves a blank page behind.
`caption`	`string | runs`		A caption makes this a FIGURE: the text is drawn under the graphic, tagged <Caption>, and collected for the list of figures ("lof"). It also tags the drawing as a Figure even without "alt" (a captioned graphic is content, not decoration), and supplies the /Alt when "alt" is absent. Numbered when the document enables "numbering": { "figures": ... }. Accepts a plain string, an ARRAY OF RUNS, or a SINGLE RUN OBJECT - the same run forms a paragraph's "runs" takes - so a caption can carry per-span styling, and above all a per-span "fontFamily": a caption that mixes scripts sets each span in the face that suits it instead of flattening to one. The flat text is always derived from the runs and stays the entry of record: it is what reaches the list of figures, the /Alt and the structure tree. A number ("Figure 3-2.") is prepended as the first run, unstyled, so it takes the block's own face.
`changed`	`boolean`		Revision / change bar: draw a vertical bar in the page margin beside this block's content to flag it as changed. A document-level changeBars object customizes the bar (width / color / side / distance); a condition tag declared with changeBar:true sets this automatically for tagged content.
`clip`	`boolean`		Clip this block's children (on a container) or its own ink to its box, with the style's corner "radius". Only bites where content is BIGGER than the box: a flow container grows to fit its children vertically, so give it "position" with a height, or a "minHeight", before expecting a vertical crop. Container-scoped, like "groupOpacity": on any other flow block it is refused with a warning. For a crop to a SHAPE rather than to the box, see "clipPath"; giving both INTERSECTS them.
`clipPath`	`object`		Crop this block to an arbitrary SHAPE stated inside its box -- a design tool's FRAME. Same vocabulary as the "shape" block and read by the same outline builder, so a shape and a clipPath given identical parameters describe the identical outline. Scoped exactly as "clip" is: a positioned block of any type (it needs a "position.height", because the shape is stated inside the box) or a container in the flow; anywhere else it is refused with a warning. Naming "clip" as well INTERSECTS the two. The shape turns, scales and skews with the block that carries it.

Key	Type	Default	Description
`condition`	`string`		Conditional text: render this block only when its condition-tag expression is shown under the document's conditions map. Whitespace/comma tag terms, each optionally '!'-negated, AND-combined; an unlisted tag defaults to shown. Empty means always shown.
`content`	`string`		Inline SVG markup (wins over src).
`float`	`string`		Float this block out of the stacked flow to one side of its column; content that FOLLOWS it wraps into the space beside it. Works on ANY block type -- a table, a container, a chart, a code listing, a figure. "center" straddles the page content width, so text in every column wraps around it. The float box takes the block's own "width"; a block that declares none takes HALF the column, so a float always leaves room beside it. Ignored on a block that also sets "position" or "anchor" -- those have already chosen their placement. Not to be confused with "anchor": { "float": true }, a BOOLEAN about migrating an anchored frame to the next column that can hold it. One of: `left`, `right`, `center`.
`groupBlend`	`string`		Composite this block AND its children once, then blend the RESULT with the page. NOT the same key as "blend", and the difference is larger than it is for opacity: two stacked children that each multiply the backdrop multiply it TWICE, which is darker than the group multiplying once, and no per-child setting reproduces it. That is what a design tool's blend-mode control on a group means. Container-scoped: on any other flow block it is refused with a warning. Also the way to blend an element together with its drop shadow, which is drawn as stacked alpha layers and so keeps its own normal compositing. See the blend-modes sample. One of: `normal`, `multiply`, `screen`, `overlay`, `darken`, `lighten`, `colorDodge`, `colorBurn`, `hardLight`, `softLight`, `difference`, `exclusion`, `hue`, `saturation`, ...
`groupOpacity`	`number`		Composite this block AND its children once, then fade the result -- true GROUP opacity. NOT the same key as "opacity", which fades each element as it paints: two overlapping opaque children at opacity 0.5 each show a seam where they cross, and under groupOpacity 0.5 they do not. The two compose. Container-scoped: on any other flow block it is refused with a warning. See the clipped-groups sample.
`height`	`number`		Explicit box height in points (else derived from the SVG viewBox aspect). With a height that does not match that aspect, the drawing keeps its viewBox's shape inside the box, as its root preserveAspectRatio says: by default xMidYMid meet (fitted whole and centred); "none" stretches it to the box, and "slice" covers the box and is clipped to it.

Key	Type	Default	Description
mirrorH	boolean | string		Mirror the whole block left-to-right about the centre of its "position" box, under the same restriction as "angle". Identical to scale x -1, and the two compose. NOT "flipH": that key belongs to the image block, where it mirrors the PICTURE INSIDE its frame. The two are separate words so an image can do both. A STRING is a data token, so each record of a template mirrors its own sticker: "{{s.flip}}" takes the record's boolean, or text "true" / "false" in any case, or 1 / 0 (a number or text); a record without it draws the block unmirrored, silently, and any other value ("yes" included) is reported with the block's pointer (and the -merge record) and drawn unmirrored. A string with no token is read as it stands.
mirrorV	boolean | string		Mirror the whole block top-to-bottom about the centre of its "position" box, under the same restriction as "angle". Identical to scale y -1. NOT "flipV" -- see "mirrorH". A STRING is a data token, so each record of a template mirrors its own sticker: "{{s.flip}}" takes the record's boolean, or text "true" / "false" in any case, or 1 / 0 (a number or text); a record without it draws the block unmirrored, silently, and any other value ("yes" included) is reported with the block's pointer (and the -merge record) and drawn unmirrored. A string with no token is read as it stands.
number	string		Pin a literal number/label for this block. Does not advance the counter.
numbered	boolean	true	Take part in automatic caption numbering when the document enables it. false leaves this one unnumbered and does not advance the counter.
opacity	number		Constant opacity for this block: 0 (invisible) to 1 (opaque, the default). REAL alpha -- a /CA /ca ExtGState in the PDF -- so a translucent panel over a photograph shows the photograph through it. Not to be confused with "style.background" {color, alpha}, which composites the tint toward WHITE at parse time and so only looks right over a white page. PER-ELEMENT, NOT GROUP: on a container it paints that container's own decoration -- background, border, accent bar, stripe -- and never its children, which draw at their own opacity (1 unless they set one). Two translucent shapes that overlap therefore composite twice, which is what per-element opacity means; group opacity, where a whole subtree is composited once and then faded, is a separate feature. Where a draw already carries its own alpha (an SVG fill-opacity, a gradient's alpha) the two MULTIPLY.

Key	Type	Default	Description
repeatAt	array of object		Imposition: draw this block ONCE and place the drawing again at each of these offsets, in points from the block's own position, y DOWNWARD as every position is. One entry = one extra copy, so a list of 7 puts 8 cards on a sheet. The drawing is captured into a Form XObject and each offset is a cm + Do of that one stream, so n copies cost one copy of the content. Translation only -- angle / scale / skew are not applied per placement. A REPEATED FORM MAY NOT CARRY TAGGED CONTENT: ISO 14289-1 7.20 allows a form's content ONE semantic parent, so repeating tagged text (or a shape with alt) fails PDF/UA -- veraPDF catches it, ap_addstruct's audit does not. The engine warns and draws it anyway. Repeat decoration: a shape or svg with no alt, or put the block in a master's frames, which is drawn as a page artifact. Only a block with position is drawn again: on a block without one (in the flow, or an anchored frame), or on a repeat / if / include block, the key is refused with a notice naming the block. Each copy takes the block's groupOpacity and groupBlend. The layout map (-map) reports the copies as repeats on the block's fragment, so a design tool can select and outline them.
scale	number | array of number | object | string		Scale the block about the same pivot as "angle" (the centre of its "position" box), and under the same restriction. A CONTAINER in the flow may also carry this: it becomes a GROUP that turns about its own measured box and reserves the bounding box of the turned result, and its children inherit the transform. Any other flow block ignores it with a warning -- wrap it in a container. A number scales both axes; [x, y] or {"x":, "y":} scales them separately. 1 = natural size. Applied AFTER layout, so a scaled block does NOT re-wrap its text at the scaled measure -- this is a design tool's scale handle, not a change of font size. A STRING is a data token, so each record of a template scales its own sticker: "{{s.scale}}" takes the record's number, [x, y] or {"x":, "y":} (members may be text of numbers), or text of one or two plain numbers such as a CSV field ("1.5", "1.5 0.8", "[1.5, 0.8]"); a record without it draws the block at its natural size, silently, and a value that is not a number or a pair is reported with the block's pointer (and the -merge record) and drawn at natural size. A string with no token is read as it stands.
skew	number | array of number | object | string		Shear the block, in DEGREES, about the same pivot as "angle", under the same restriction. [x, y] or {"x":, "y":}; a bare number skews x only. Positive x leans the top to the right and positive y lifts the right-hand side -- both counter-clockwise, matching "angle". Clamped to +/-89.9: a 90-degree skew is a singular matrix and draws nothing. A STRING is a data token, so each record of a template skews its own sticker: "{{s.skew}}" takes the record's number (x only), [x, y] or {"x":, "y":}, or text of one or two plain numbers; clamped as the written one is. A record without it draws the block unskewed, silently, and a value that is not a number or a pair is reported with the block's pointer (and the -merge record) and drawn unskewed. A string with no token is read as it stands.
src	string		Path to an .svg file (relative to the spec, made absolute at load).

Key	Type	Default	Description
straddle	boolean		Straddle across columns: when true and the body is multi-column, this block spans the FULL content width and text in every column wraps around it (above and below) - a spanning float generalized from center-floated images to any block. No effect in a single-column body, on a positioned / anchored block, or on an image/pdf already using float:"center" (the same mechanism). Does not balance the columns above it or act as a section divider.
verticalAlign	string		Where the content sits in its "position" box when it is shorter than the box's height: "top" (the default), "middle" ("center" is the same) or "bottom". A paragraph, heading or pre moves as a whole (its own box with it); a container keeps its box where it is and moves its children inside it (a child with its own "position" stays put). The offset is the box height less the content's height as it is actually placed -- the same lines, shrinkToFit size and padding that are then drawn -- halved for middle. Content as tall as the box or taller does not move: it is clipped and reported exactly as without the key. Needs a position with a height: on a block without one, on any other kind of block, on text set along a textPath "d" curve (stated in fractions of the whole box, so drawn across all of it; the "arc" form aligns), and on vertical text whose columns fill the box (columns run the full height unless a paragraph's or a heading's boxHeight is shorter), it is refused with a notice and ignored. The layout map keeps reporting the box. One of: top, middle, bottom, center.
width	dimension		Box width (points or "NN%"); the height follows the SVG aspect unless given.
z	number		CANVAS MODE ONLY (see the top-level "pages"): where this element sits in the PAINT order of its page. Lower draws first, further back; unset means "wherever I am in the list", and ties keep author order. It moves the PAINT and not the READING: the structure tree keeps the document order, so a caption written first because it is read first can still be painted last because it sits on top. That separation is what a flattened scene graph cannot express. Ignored outside canvas mode, where the flow already fixes both orders.

The chart element

Type name: chart.

A data chart drawn by the engine: bar, line, area, pie, scatter, radar, heatmap, gauge, box plot, candlestick, treemap, sankey and more, including combination charts with a secondary axis. You supply the series; the engine chooses tick intervals, draws the axes, grid, legend and labels. Give it an alt so the picture has a spoken description.

Example

```
{
  "type": "chart",
  "chartType": "bar",
  "title": "Units by quarter",
  "size": {
    "width": "100%",
    "height": 180
  },
  "yAxis": {
    "label": "Units",
    "grid": true
  },
  "series": [
    {
      "name": "2026",
      "points": [
        [
          0,
          12
        ],
        [
          1,
          19
        ],
        [
          2,
          15
        ]
      ]
    }
  ],
  "alt": "A bar chart of units for three quarters: 12, 19 and 15."
}
```

Reference lines and a shaded value band

```
{
  "type": "chart",
  "chartType": "line",
  "title": "API latency (p95)",
  "size": {
    "height": 220
  },
  "xAxis": {
    "label": "Hour",
    "min": 0,
    "max": 24,
    "majorTick": 6
  },
  "yAxis": {
    "label": "ms",
    "min": 0,
    "max": 300,
    "majorTick": 50,
    "grid": true
  },
  "bands": [
    {
      "yFrom": 0,
      "yTo": 100,
      "color": "#2e8b57",
      "label": "Healthy"
    }
  ],
  "referenceLines": [
    {
      "y": 200,
      "label": "SLA",
      "color": "#c0202a"
    },
    {
      "y": 120,
      "label": "Target",
      "color": "#2e6f2e",
      "dashed": true
    }
  ],
  "series": [
    {
      "name": "p95",
      "points": [
        [
          0,
          60
        ],
        [
          6,
          88
        ],
        [
          12,
          175
        ],
```

```
          [
            18,
            205
          ],
          [
            24,
            95
          ]
        ]
      }
  ],
  "alt": "A latency line that crosses a 200 ms SLA reference line."
}
```

A horizontal bar with a value threshold on the X axis

```
{
  "type": "chart",
  "chartType": "bar",
  "horizontal": true,
  "title": "Revenue vs target",
  "size": {
    "height": 180
  },
  "categories": [
    "North",
    "South",
    "East",
    "West"
  ],
  "yAxis": {
    "unit": "K",
    "min": 0,
    "max": 120,
    "majorTick": 20
  },
  "referenceLines": [
    {
      "x": 80,
      "label": "Target",
      "color": "#c0202a"
    }
  ],
  "series": [
    {
      "name": "Revenue",
      "points": [
        [
          0,
          55
        ],
        [
          1,
          72
        ],
        [
          2,
          95
        ],
        [
          3,
          110
        ]
      ]
    }
  ],
  "alt": "Four regional revenue bars against an 80K target line."
}
```

Error bars — a symmetric mean +/- SD whisker on every point

```
{
  "type": "chart",
  "chartType": "line",
  "title": "Reaction time",
  "size": {
    "height": 200
  },
  "markers": true,
  "xAxis": {
    "label": "Session",
    "min": 1,
    "max": 4,
    "majorTick": 1
  },
  "yAxis": {
    "label": "ms",
    "min": 0,
    "max": 500,
    "majorTick": 100,
    "grid": true
  },
  "series": [
    {
      "name": "RT",
      "error": 35,
      "points": [
        [
          1,
          420
        ],
        [
          2,
          395
        ],
        [
          3,
          360
        ],
        [
          4,
          350
        ]
      ]
    }
  ],
  "alt": "A reaction-time line with a plus-or-minus 35 ms error bar on every point."
}
```

Keys

Key	Type	Default	Description
`alt`	`string`		Alternative text describing the chart, emitted for accessibility. Required for a tagged or PDF/UA document, since the chart itself is just artwork.
`anchor`	`anchor`		-
`angle`	`number | string`		Turn the block, in degrees counter-clockwise, about the CENTRE of its "position" box. A CONTAINER in the flow may also carry this: it becomes a GROUP that turns about its own measured box and reserves the bounding box of the turned result, and its children inherit the transform. Any other flow block ignores it with a warning -- wrap it in a container. The block lays out upright inside its box and the finished result is turned, so a "position.height" that truncates content truncates the same text it would unrotated, and the turned result is NOT clipped to its box: it may extend past it, exactly as an unrotated positioned block may. Not to be confused with the image block's "rotate", which is an integer QUARTER TURN CLOCKWISE applied to the picture INSIDE its box (and re-sizes that box to the turned orientation). The two compose -- an image can stand a sideways scan upright with "rotate" and tilt the whole frame with "angle". Same meaning as the watermark's "angle". A STRING is a data token, so each record of a template turns its own sticker: "{{s.angle}}" takes the record's number (or text of one, such as a CSV field); a record without it leaves the block upright, silently, and a value that is not a number is reported with the block's pointer (and the -merge record) and drawn upright. A string with no token is read as it stands.
`angleClockwise`	`boolean`		Polar chart: sweep angles clockwise instead of anticlockwise.
`angleStart`	`number`		Polar: degrees where angle 0 sits (0 = right / 3 o'clock).
`angleUnit`	`string`		Polar chart: whether angle values are degrees or radians. One of: `deg`, `rad`.
`bands`	`array of object`		Cartesian charts: shaded value-range bands painted behind the data (a control range, a target zone, an incident window). Rendered as an opaque tint so a tagged chart stays PDF/UA clean.
`bleed`	`number`		Step and Repeat: how far this placed block's box reaches past its TRIM on every side, in points -- the box IS the bleed box, and the trim is the box inset by this much. A design tool sizes a card by it (the trim plus twice the bleed) and draws the trim line inside each copy; the engine draws nothing from it, and keeps the bleed only through the block's own `clip`. Only a block with `position` has a trim: on any other block the key is refused with a notice naming the block.
`blend`	`string`		How this block's OWN ink combines with the page under it -- the sixteen PDF blend modes (ISO 32000-1 11.3.5), written as /BM. On a container that means its background, border and stripe, never its children; for the subtree as one object see "groupBlend". Case and separators are folded, so "colorDodge", "color-dodge" and "COLORDODGE" are one mode. An unrecognized name is refused with a warning and the block paints normally. See the blend-modes sample. One of: `normal`, `multiply`, `screen`, `overlay`, `darken`, `lighten`, `colorDodge`, `colorBurn`, `hardLight`, `softLight`, `difference`, `exclusion`, `hue`, `saturation`, ...

Key	Type	Default	Description
`breakBefore`	`boolean`		Pagination: open a new page before this block - a part divider, an appendix. Ignored when the page is already empty, so it never leaves a blank page behind.
`caption`	`string | runs`		A caption makes this a FIGURE: the text is drawn under the plot, tagged <Caption>, and collected for the list of figures ("lof"). Distinct from "title", which is drawn INSIDE the plot frame as part of the graphic. Supplies the /Alt when "alt" is absent. Numbered when the document enables "numbering": { "figures": ... }. Accepts a plain string, an ARRAY OF RUNS, or a SINGLE RUN OBJECT - the same run forms a paragraph's "runs" takes - so a caption can carry per-span styling, and above all a per-span "fontFamily": a caption that mixes scripts sets each span in the face that suits it instead of flattening to one. The flat text is always derived from the runs and stays the entry of record: it is what reaches the list of figures, the /Alt and the structure tree. A number ("Figure 3-2.") is prepended as the first run, unstyled, so it takes the block's own face.
`categories`	`array of string`		Labels for the slots along the shared category axis. A point is matched to its label by the RANK of its x among the distinct x values across every series - not by the x value itself - so points at x 1,2,3,4 with four categories still read as the first four labels, and a series that skips a category leaves that slot empty instead of shifting the labels after it. Honored on bar (either orientation), line, area and combo, where it turns the x axis into a category axis: one tick per category, no "nice" numeric rounding, and names in place of numbers. Also honored by radar (one spoke each), waterfall, funnel, treemap and sankey. Ignored on scatter and bubble, whose x is a real measurement. The array must DESCRIBE the whole axis for the category axis to engage: a name for every slot, and slots evenly spaced. If either fails - a short array, or x values that are not an arithmetic progression - the x axis stays numeric, which is what keeps a name off the wrong bar. A combo chart whose trend line sits off the bar grid has more slots than the bars do, so it keeps its numbers. Spare names past the end of the data are ignored.
`changed`	`boolean`		Revision / change bar: draw a vertical bar in the page margin beside this block's content to flag it as changed. A document-level changeBars object customizes the bar (width / color / side / distance); a condition tag declared with changeBar:true sets this automatically for tagged content.
`chartType`	`string`		Which chart to draw. The rest of the block's keys are read according to this - categories+series for bar/line/area/combo, slices for pie, x/y points for scatter. One of: `line`, `scatter`, `bar`, `polar`, `pie`, `doughnut`, `donut`, `radar`, `spider`, `area`, `bubble`, `heatmap`, `gauge`, `waterfall`, ...
`clip`	`boolean`		Clip this block's children (on a container) or its own ink to its box, with the style's corner "radius". Only bites where content is BIGGER than the box: a flow container grows to fit its children vertically, so give it "position" with a height, or a "minHeight", before expecting a vertical crop. Container-scoped, like "groupOpacity": on any other flow block it is refused with a warning. For a crop to a SHAPE rather than to the box, see "clipPath"; giving both INTERSECTS them.

Key	Type	Default	Description
clipPath	object		Crop this block to an arbitrary SHAPE stated inside its box -- a design tool's FRAME. Same vocabulary as the "shape" block and read by the same outline builder, so a shape and a clipPath given identical parameters describe the identical outline. Scoped exactly as "clip" is: a positioned block of any type (it needs a "position.height", because the shape is stated inside the box) or a container in the flow; anywhere else it is refused with a warning. Naming "clip" as well INTERSECTS the two. The shape turns, scales and skews with the block that carries it.
closed	boolean		Polar: connect the last point back to the first.
condition	string		Conditional text: render this block only when its condition-tag expression is shown under the document's conditions map. Whitespace/comma tag terms, each optionally '!'-negated, AND-combined; an unlisted tag defaults to shown. Empty means always shown.
fillOpacity	number		Radar: translucent fill strength.
float	string		Float this block out of the stacked flow to one side of its column; content that FOLLOWS it wraps into the space beside it. Works on ANY block type -- a table, a container, a chart, a code listing, a figure. "center" straddles the page content width, so text in every column wraps around it. The float box takes the block's own "width"; a block that declares none takes HALF the column, so a float always leaves room beside it. Ignored on a block that also sets "position" or "anchor" -- those have already chosen their placement. Not to be confused with "anchor": { "float": true }, a BOOLEAN about migrating an anchored frame to the next column that can hold it. One of: left, right, center.
groupBlend	string		Composite this block AND its children once, then blend the RESULT with the page. NOT the same key as "blend", and the difference is larger than it is for opacity: two stacked children that each multiply the backdrop multiply it TWICE, which is darker than the group multiplying once, and no per-child setting reproduces it. That is what a design tool's blend-mode control on a group means. Container-scoped: on any other flow block it is refused with a warning. Also the way to blend an element together with its drop shadow, which is drawn as stacked alpha layers and so keeps its own normal compositing. See the blend-modes sample. One of: normal, multiply, screen, overlay, darken, lighten, colorDodge, colorBurn, hardLight, softLight, difference, exclusion, hue, saturation, ...
groupOpacity	number		Composite this block AND its children once, then fade the result -- true GROUP opacity. NOT the same key as "opacity", which fades each element as it paints: two overlapping opaque children at opacity 0.5 each show a seam where they cross, and under groupOpacity 0.5 they do not. The two compose. Container-scoped: on any other flow block it is refused with a warning. See the clipped-groups sample.
height	dimension		Shorthand for size.height.
highColor	color		Heatmap ramp end (default the first palette color).

Key	Type	Default	Description
holeRatio	number		0 = pie; > 0 = doughnut hole.
horizontal	boolean		Draw the bars horizontally, so the category axis runs down the side and the value axis along the bottom. Worth it when the category names are long: the plot widens its left gutter to fit them, where a vertical bar has only its slot width and thins the labels to fit. Bar only.
legend	boolean		Draw the series legend.
legendAt	object		Put the legend at a point of your own: its box's top-left corner, measured from the chart's top-left -- x across and y down, each in points or as a percentage of the chart's width (x) or height (y). The legend is drawn over the chart and kept inside it. Wins over legendPos.
legendPos	string		Where the legend sits. Inside the plot, in a corner: tl, tr (the default), bl or br; it is drawn over the plot, so a tall bar can run under it. Outside the plot, on one side of the chart: top, bottom, left or right; the plot gives up the room the legend needs, and a top or bottom legend lists its entries side by side. legendAt, when set, wins over this. One of: tl, tr, bl, br, top, bottom, left, right.
lowColor	color		Heatmap ramp start (default white).
markers	boolean		Draw a point marker at each data value, not just the connecting line.
matrix	array of array of number		Heatmap: rows of values (row 0 at the top).
mirrorH	boolean | string		Mirror the whole block left-to-right about the centre of its "position" box, under the same restriction as "angle". Identical to scale x -1, and the two compose. NOT "flipH": that key belongs to the image block, where it mirrors the PICTURE INSIDE its frame. The two are separate words so an image can do both. A STRING is a data token, so each record of a template mirrors its own sticker: "{{s.flip}}" takes the record's boolean, or text "true" / "false" in any case, or 1 / 0 (a number or text); a record without it draws the block unmirrored, silently, and any other value ("yes" included) is reported with the block's pointer (and the -merge record) and drawn unmirrored. A string with no token is read as it stands.
mirrorV	boolean | string		Mirror the whole block top-to-bottom about the centre of its "position" box, under the same restriction as "angle". Identical to scale y -1. NOT "flipV" -- see "mirrorH". A STRING is a data token, so each record of a template mirrors its own sticker: "{{s.flip}}" takes the record's boolean, or text "true" / "false" in any case, or 1 / 0 (a number or text); a record without it draws the block unmirrored, silently, and any other value ("yes" included) is reported with the block's pointer (and the -merge record) and drawn unmirrored. A string with no token is read as it stands.
number	string		Pin a literal number/label for this block. Does not advance the counter.

Key	Type	Default	Description
numbered	boolean	true	Take part in automatic caption numbering when the document enables it. false leaves this one unnumbered and does not advance the counter.
opacity	number		Constant opacity for this block: 0 (invisible) to 1 (opaque, the default). REAL alpha -- a /CA /ca ExtGState in the PDF -- so a translucent panel over a photograph shows the photograph through it. Not to be confused with "style.background" {color, alpha}, which composites the tint toward WHITE at parse time and so only looks right over a white page. PER-ELEMENT, NOT GROUP: on a container it paints that container's own decoration -- background, border, accent bar, stripe -- and never its children, which draw at their own opacity (1 unless they set one). Two translucent shapes that overlap therefore composite twice, which is what per-element opacity means; group opacity, where a whole subtree is composited once and then faded, is a separate feature. Where a draw already carries its own alpha (an SVG fill-opacity, a gradient's alpha) the two MULTIPLY.
referenceLines	array of object		Cartesian charts: reference / threshold rules drawn across the plot at a fixed data value (a target, a budget, a spec limit). Drawn over the data.
repeatAt	array of object		Imposition: draw this block ONCE and place the drawing again at each of these offsets, in points from the block's own position, y DOWNWARD as every position is. One entry = one extra copy, so a list of 7 puts 8 cards on a sheet. The drawing is captured into a Form XObject and each offset is a cm + Do of that one stream, so n copies cost one copy of the content. Translation only -- angle / scale / skew are not applied per placement. A REPEATED FORM MAY NOT CARRY TAGGED CONTENT: ISO 14289-1 7.20 allows a form's content ONE semantic parent, so repeating tagged text (or a shape with alt) fails PDF/UA -- veraPDF catches it, ap_addstruct's audit does not. The engine warns and draws it anyway. Repeat decoration: a shape or svg with no alt, or put the block in a master's frames, which is drawn as a page artifact. Only a block with position is drawn again: on a block without one (in the flow, or an anchored frame), or on a repeat / if / include block, the key is refused with a notice naming the block. Each copy takes the block's groupOpacity and groupBlend. The layout map (-map) reports the copies as repeats on the block's fragment, so a design tool can select and outline them.

Key	Type	Default	Description
scale	number | array of number | object | string		Scale the block about the same pivot as "angle" (the centre of its "position" box), and under the same restriction. A CONTAINER in the flow may also carry this: it becomes a GROUP that turns about its own measured box and reserves the bounding box of the turned result, and its children inherit the transform. Any other flow block ignores it with a warning -- wrap it in a container. A number scales both axes; [x, y] or {"x":, "y":} scales them separately. 1 = natural size. Applied AFTER layout, so a scaled block does NOT re-wrap its text at the scaled measure -- this is a design tool's scale handle, not a change of font size. A STRING is a data token, so each record of a template scales its own sticker: "{{s.scale}}" takes the record's number, [x, y] or {"x":, "y":} (members may be text of numbers), or text of one or two plain numbers such as a CSV field ("1.5", "1.5 0.8", "[1.5, 0.8]"); a record without it draws the block at its natural size, silently, and a value that is not a number or a pair is reported with the block's pointer (and the -merge record) and drawn at natural size. A string with no token is read as it stands.
series	array of object		The data series. Each carries a name, a color and its points.
size	object		Explicit plot size, when the chart should not take the full column width.
skew	number | array of number | object | string		Shear the block, in DEGREES, about the same pivot as "angle", under the same restriction. [x, y] or {"x":, "y":}; a bare number skews x only. Positive x leans the top to the right and positive y lifts the right-hand side -- both counter-clockwise, matching "angle". Clamped to +/-89.9: a 90-degree skew is a singular matrix and draws nothing. A STRING is a data token, so each record of a template skews its own sticker: "{{s.skew}}" takes the record's number (x only), [x, y] or {"x":, "y":}, or text of one or two plain numbers; clamped as the written one is. A record without it draws the block unskewed, silently, and a value that is not a number or a pair is reported with the block's pointer (and the -merge record) and drawn unskewed. A string with no token is read as it stands.
smooth	boolean		Centripetal Catmull-Rom spline (line + polar).
stack100	boolean		Stack the series AND scale each category to fill 100%, showing proportion rather than magnitude.
stacked	boolean		Stack the series on top of each other instead of drawing them side by side.
straddle	boolean		Straddle across columns: when true and the body is multi-column, this block spans the FULL content width and text in every column wraps around it (above and below) - a spanning float generalized from center-floated images to any block. No effect in a single-column body, on a positioned / anchored block, or on an image/pdf already using float:"center" (the same mechanism). Does not balance the columns above it or act as a section divider.
subtitle	string		Secondary line drawn under the title.
title	string		Heading drawn above the plot area.
value	number		Gauge: the value the needle points at.
valueLabels	boolean		Print each value as text on the chart.

Key	Type	Default	Description
verticalAlign	string		Where the content sits in its "position" box when it is shorter than the box's height: "top" (the default), "middle" ("center" is the same) or "bottom". A paragraph, heading or pre moves as a whole (its own box with it); a container keeps its box where it is and moves its children inside it (a child with its own "position" stays put). The offset is the box height less the content's height as it is actually placed -- the same lines, shrinkToFit size and padding that are then drawn -- halved for middle. Content as tall as the box or taller does not move: it is clipped and reported exactly as without the key. Needs a position with a height: on a block without one, on any other kind of block, on text set along a textPath "d" curve (stated in fractions of the whole box, so drawn across all of it; the "arc" form aligns), and on vertical text whose columns fill the box (columns run the full height unless a paragraph's or a heading's boxHeight is shorter), it is refused with a notice and ignored. The layout map keeps reporting the box. One of: top, middle, bottom, center.
width	dimension		Shorthand for size.width.
xAxis	chartAxis		X-axis configuration - title, range, tick spacing, gridlines.
xLabels	array of string		Column labels along the bottom of a matrix chart: heatmap, candlestick and box-plot. The cartesian types use categories instead.
yAxis	chartAxis		Y-axis configuration - title, range, tick spacing, gridlines.
yAxis2	chartAxis		Combo: the secondary (right) value axis for series with axis/secondaryAxis.
yLabels	array of string		Heatmap row labels.
z	number		CANVAS MODE ONLY (see the top-level "pages"): where this element sits in the PAINT order of its page. Lower draws first, further back; unset means "wherever I am in the list", and ties keep author order. It moves the PAINT and not the READING: the structure tree keeps the document order, so a caption written first because it is read first can still be painted last because it sits on top. That separation is what a flattened scene graph cannot express. Ignored outside canvas mode, where the flow already fixes both orders.
zones	array of object		Gauge: colored bands up to each threshold, in range order.

The barcode element

Type name: `barcode`.
A barcode or 2-D symbol - QR, Data Matrix, PDF417, Code 128, EAN, UPC and the rest of the zint symbology set. Supply the payload as `data`; the engine encodes and draws it at a size that scans.

Example

```
{
  "type": "barcode",
  "code": "qrcode",
  "data": "https://appligent.com",
  "width": 90,
  "alt": "QR code linking to appligent.com"
}
```

Keys

Key	Type	Default	Description
`alt`	`string`		Tagged-PDF /Alt for the Figure (default: the encoded data).
`anchor`	`anchor`		-
`angle`	`number` | `string`		Turn the block, in degrees counter-clockwise, about the CENTRE of its "position" box. A CONTAINER in the flow may also carry this: it becomes a GROUP that turns about its own measured box and reserves the bounding box of the turned result, and its children inherit the transform. Any other flow block ignores it with a warning -- wrap it in a container. The block lays out upright inside its box and the finished result is turned, so a "position.height" that truncates content truncates the same text it would unrotated, and the turned result is NOT clipped to its box: it may extend past it, exactly as an unrotated positioned block may. Not to be confused with the image block's "rotate", which is an integer QUARTER TURN CLOCKWISE applied to the picture INSIDE its box (and re-sizes that box to the turned orientation). The two compose -- an image can stand a sideways scan upright with "rotate" and tilt the whole frame with "angle". Same meaning as the watermark's "angle". A STRING is a data token, so each record of a template turns its own sticker: "{{s.angle}}" takes the record's number (or text of one, such as a CSV field); a record without it leaves the block upright, silently, and a value that is not a number is reported with the block's pointer (and the -merge record) and drawn upright. A string with no token is read as it stands.
`bleed`	`number`		Step and Repeat: how far this placed block's box reaches past its TRIM on every side, in points -- the box IS the bleed box, and the trim is the box inset by this much. A design tool sizes a card by it (the trim plus twice the bleed) and draws the trim line inside each copy; the engine draws nothing from it, and keeps the bleed only through the block's own `clip`. Only a block with `position` has a trim: on any other block the key is refused with a notice naming the block.

Key	Type	Default	Description
blend	string		How this block's OWN ink combines with the page under it -- the sixteen PDF blend modes (ISO 32000-1 11.3.5), written as /BM. On a container that means its background, border and stripe, never its children; for the subtree as one object see "groupBlend". Case and separators are folded, so "colorDodge", "color-dodge" and "COLORDODGE" are one mode. An unrecognized name is refused with a warning and the block paints normally. See the blend-modes sample. One of: normal, multiply, screen, overlay, darken, lighten, colorDodge, colorBurn, hardLight, softLight, difference, exclusion, hue, saturation, ...
breakBefore	boolean		Pagination: open a new page before this block - a part divider, an appendix. Ignored when the page is already empty, so it never leaves a blank page behind.
changed	boolean		Revision / change bar: draw a vertical bar in the page margin beside this block's content to flag it as changed. A document-level changeBars object customizes the bar (width / color / side / distance); a condition tag declared with changeBar:true sets this automatically for tagged content.
clip	boolean		Clip this block's children (on a container) or its own ink to its box, with the style's corner "radius". Only bites where content is BIGGER than the box: a flow container grows to fit its children vertically, so give it "position" with a height, or a "minHeight", before expecting a vertical crop. Container-scoped, like "groupOpacity": on any other flow block it is refused with a warning. For a crop to a SHAPE rather than to the box, see "clipPath"; giving both INTERSECTS them.
clipPath	object		Crop this block to an arbitrary SHAPE stated inside its box -- a design tool's FRAME. Same vocabulary as the "shape" block and read by the same outline builder, so a shape and a clipPath given identical parameters describe the identical outline. Scoped exactly as "clip" is: a positioned block of any type (it needs a "position.height", because the shape is stated inside the box) or a container in the flow; anywhere else it is refused with a warning. Naming "clip" as well INTERSECTS the two. The shape turns, scales and skews with the block that carries it.
code	string		Symbology name - every symbology the engine draws, plus convenient aliases. A name the engine does not know draws a gray placeholder box and leaves the figure untagged (the engine says so), which is why a misspelling is refused here. The names below are the engine's own table. Like anchor.to, the ENGINE normalizes by lowercasing and dropping punctuation, so "GS1-128" renders and does NOT validate; the schema is the strict companion, and lists the normalized spelling. Write the normalized form and both agree. One of: 128, 39, 93, auspost, ausredirect, ausreply, ausroute, azrune, aztec, bc412, c25iata, c25ind, c25inter, c25logic, ...
condition	string		Conditional text: render this block only when its condition-tag expression is shown under the document's conditions map. Whitespace/comma tag terms, each optionally '!'-negated, AND-combined; an unlisted tag defaults to shown. Empty means always shown.

Key	Type	Default	Description
`data`	`string`		The content to encode (alias: "text").
`ecc`	`integer`		Alias for "security".
`float`	`string`		Float this block out of the stacked flow to one side of its column; content that FOLLOWS it wraps into the space beside it. Works on ANY block type -- a table, a container, a chart, a code listing, a figure. "center" straddles the page content width, so text in every column wraps around it. The float box takes the block's own "width"; a block that declares none takes HALF the column, so a float always leaves room beside it. Ignored on a block that also sets "position" or "anchor" -- those have already chosen their placement. Not to be confused with "anchor": { "float": true }, a BOOLEAN about migrating an anchored frame to the next column that can hold it. One of: `left`, `right`, `center`.
`groupBlend`	`string`		Composite this block AND its children once, then blend the RESULT with the page. NOT the same key as "blend", and the difference is larger than it is for opacity: two stacked children that each multiply the backdrop multiply it TWICE, which is darker than the group multiplying once, and no per-child setting reproduces it. That is what a design tool's blend-mode control on a group means. Container-scoped: on any other flow block it is refused with a warning. Also the way to blend an element together with its drop shadow, which is drawn as stacked alpha layers and so keeps its own normal compositing. See the blend-modes sample. One of: `normal`, `multiply`, `screen`, `overlay`, `darken`, `lighten`, `colorDodge`, `colorBurn`, `hardLight`, `softLight`, `difference`, `exclusion`, `hue`, `saturation`, ...
`groupOpacity`	`number`		Composite this block AND its children once, then fade the result -- true GROUP opacity. NOT the same key as "opacity", which fades each element as it paints: two overlapping opaque children at opacity 0.5 each show a seam where they cross, and under groupOpacity 0.5 they do not. The two compose. Container-scoped: on any other flow block it is refused with a warning. See the clipped-groups sample.
`mirrorH`	`boolean | string`		Mirror the whole block left-to-right about the centre of its "position" box, under the same restriction as "angle". Identical to scale x -1, and the two compose. NOT "flipH": that key belongs to the image block, where it mirrors the PICTURE INSIDE its frame. The two are separate words so an image can do both. A STRING is a data token, so each record of a template mirrors its own sticker: "{{s.flip}}" takes the record's boolean, or text "true" / "false" in any case, or 1 / 0 (a number or text); a record without it draws the block unmirrored, silently, and any other value ("yes" included) is reported with the block's pointer (and the -merge record) and drawn unmirrored. A string with no token is read as it stands.

Key	Type	Default	Description
`mirrorV`	`boolean | string`		Mirror the whole block top-to-bottom about the centre of its "position" box, under the same restriction as "angle". Identical to scale y -1. NOT "flipV" -- see "mirrorH". A STRING is a data token, so each record of a template mirrors its own sticker: "{{s.flip}}" takes the record's boolean, or text "true" / "false" in any case, or 1 / 0 (a number or text); a record without it draws the block unmirrored, silently, and any other value ("yes" included) is reported with the block's pointer (and the -merge record) and drawn unmirrored. A string with no token is read as it stands.
`opacity`	`number`		Constant opacity for this block: 0 (invisible) to 1 (opaque, the default). REAL alpha -- a /CA /ca ExtGState in the PDF -- so a translucent panel over a photograph shows the photograph through it. Not to be confused with "style.background" {color, alpha}, which composites the tint toward WHITE at parse time and so only looks right over a white page. PER-ELEMENT, NOT GROUP: on a container it paints that container's own decoration -- background, border, accent bar, stripe -- and never its children, which draw at their own opacity (1 unless they set one). Two translucent shapes that overlap therefore composite twice, which is what per-element opacity means; group opacity, where a whole subtree is composited once and then faded, is a separate feature. Where a draw already carries its own alpha (an SVG fill-opacity, a gradient's alpha) the two MULTIPLY.
`repeatAt`	`array of object`		Imposition: draw this block ONCE and place the drawing again at each of these offsets, in points from the block's own position, y DOWNWARD as every `position` is. One entry = one extra copy, so a list of 7 puts 8 cards on a sheet. The drawing is captured into a Form XObject and each offset is a `cm` + `Do` of that one stream, so n copies cost one copy of the content. Translation only -- `angle` / `scale` / `skew` are not applied per placement. A REPEATED FORM MAY NOT CARRY TAGGED CONTENT: ISO 14289-1 7.20 allows a form's content ONE semantic parent, so repeating tagged text (or a shape with `alt`) fails PDF/UA -- veraPDF catches it, ap_addstruct's audit does not. The engine warns and draws it anyway. Repeat decoration: a shape or svg with no `alt`, or put the block in a master's `frames`, which is drawn as a page artifact. Only a block with `position` is drawn again: on a block without one (in the flow, or an anchored frame), or on a repeat / if / include block, the key is refused with a notice naming the block. Each copy takes the block's `groupOpacity` and `groupBlend`. The layout map (-map) reports the copies as `repeats` on the block's fragment, so a design tool can select and outline them.

Key	Type	Default	Description
scale	number | array of number | object | string		Scale the block about the same pivot as "angle" (the centre of its "position" box), and under the same restriction. A CONTAINER in the flow may also carry this: it becomes a GROUP that turns about its own measured box and reserves the bounding box of the turned result, and its children inherit the transform. Any other flow block ignores it with a warning -- wrap it in a container. A number scales both axes; [x, y] or {"x":, "y":} scales them separately. 1 = natural size. Applied AFTER layout, so a scaled block does NOT re-wrap its text at the scaled measure -- this is a design tool's scale handle, not a change of font size. A STRING is a data token, so each record of a template scales its own sticker: "{{s.scale}}" takes the record's number, [x, y] or {"x":, "y":} (members may be text of numbers), or text of one or two plain numbers such as a CSV field ("1.5", "1.5 0.8", "[1.5, 0.8]"); a record without it draws the block at its natural size, silently, and a value that is not a number or a pair is reported with the block's pointer (and the -merge record) and drawn at natural size. A string with no token is read as it stands.
security	integer		Error correction: QR's ECC level 1-4, PDF417's security level 0-8 (alias: "ecc").
showText	boolean		Human-readable text line (drawn with the document's fonts).
skew	number | array of number | object | string		Shear the block, in DEGREES, about the same pivot as "angle", under the same restriction. [x, y] or {"x":, "y":}; a bare number skews x only. Positive x leans the top to the right and positive y lifts the right-hand side -- both counter-clockwise, matching "angle". Clamped to +/-89.9: a 90-degree skew is a singular matrix and draws nothing. A STRING is a data token, so each record of a template skews its own sticker: "{{s.skew}}" takes the record's number (x only), [x, y] or {"x":, "y":}, or text of one or two plain numbers; clamped as the written one is. A record without it draws the block unskewed, silently, and a value that is not a number or a pair is reported with the block's pointer (and the -merge record) and drawn unskewed. A string with no token is read as it stands.
straddle	boolean		Straddle across columns: when true and the body is multi-column, this block spans the FULL content width and text in every column wraps around it (above and below) - a spanning float generalized from center-floated images to any block. No effect in a single-column body, on a positioned / anchored block, or on an image/pdf already using float:"center" (the same mechanism). Does not balance the columns above it or act as a section divider.
symbology	string		Alias for "code".
text	string		Alias for "data".

Key	Type	Default	Description
verticalAlign	string		Where the content sits in its "position" box when it is shorter than the box's height: "top" (the default), "middle" ("center" is the same) or "bottom". A paragraph, heading or pre moves as a whole (its own box with it); a container keeps its box where it is and moves its children inside it (a child with its own "position" stays put). The offset is the box height less the content's height as it is actually placed -- the same lines, shrinkToFit size and padding that are then drawn -- halved for middle. Content as tall as the box or taller does not move: it is clipped and reported exactly as without the key. Needs a position with a height: on a block without one, on any other kind of block, on text set along a textPath "d" curve (stated in fractions of the whole box, so drawn across all of it; the "arc" form aligns), and on vertical text whose columns fill the box (columns run the full height unless a paragraph's or a heading's boxHeight is shorter), it is refused with a notice and ignored. The layout map keeps reporting the box. One of: top, middle, bottom, center.
width	number		Display width in points (default: 120 square symbols, 216 wide).
z	number		CANVAS MODE ONLY (see the top-level "pages"): where this element sits in the PAINT order of its page. Lower draws first, further back; unset means "wherever I am in the list", and ties keep author order. It moves the PAINT and not the READING: the structure tree keeps the document order, so a caption written first because it is read first can still be painted last because it sits on top. That separation is what a flattened scene graph cannot express. Ignored outside canvas mode, where the flow already fixes both orders.

The math element

Type name: math. Accepted aliases: equation.

A display equation. Supply the visual as pre-rendered SVG and the meaning as Presentation MathML: the engine draws the SVG and embeds the MathML as an associated file on the Formula structure element, so assistive technology reads the real mathematics rather than a picture. With equation numbering enabled the engine adds the number in the right gutter.

Example

```
{
  "type": "math",
  "align": "center",
  "mathml":
    "<math><mi>E</mi><mo>=</mo><mi>m</mi><msup><mi>c</mi><mn>2</mn></msup></math>",
  "src": "energy.svg",
  "alt": "E equals m c squared"
}
```

Keys

Key	Type	Default	Description
alt	string		A spoken reading of the equation (the <Formula> /Alt). Also what an "as": "title" cross-reference to this equation reads back.

Key	Type	Default	Description
anchor	anchor		-
angle	number | string		Turn the block, in degrees counter-clockwise, about the CENTRE of its "position" box. A CONTAINER in the flow may also carry this: it becomes a GROUP that turns about its own measured box and reserves the bounding box of the turned result, and its children inherit the transform. Any other flow block ignores it with a warning -- wrap it in a container. The block lays out upright inside its box and the finished result is turned, so a "position.height" that truncates content truncates the same text it would unrotated, and the turned result is NOT clipped to its box: it may extend past it, exactly as an unrotated positioned block may. Not to be confused with the image block's "rotate", which is an integer QUARTER TURN CLOCKWISE applied to the picture INSIDE its box (and re-sizes that box to the turned orientation). The two compose -- an image can stand a sideways scan upright with "rotate" and tilt the whole frame with "angle". Same meaning as the watermark's "angle". A STRING is a data token, so each record of a template turns its own sticker: "{{s.angle}}" takes the record's number (or text of one, such as a CSV field); a record without it leaves the block upright, silently, and a value that is not a number is reported with the block's pointer (and the -merge record) and drawn upright. A string with no token is read as it stands.
bleed	number		Step and Repeat: how far this placed block's box reaches past its TRIM on every side, in points -- the box IS the bleed box, and the trim is the box inset by this much. A design tool sizes a card by it (the trim plus twice the bleed) and draws the trim line inside each copy; the engine draws nothing from it, and keeps the bleed only through the block's own clip. Only a block with position has a trim: on any other block the key is refused with a notice naming the block.
blend	string		How this block's OWN ink combines with the page under it -- the sixteen PDF blend modes (ISO 32000-1 11.3.5), written as /BM. On a container that means its background, border and stripe, never its children; for the subtree as one object see "groupBlend". Case and separators are folded, so "colorDodge", "color-dodge" and "COLORDODGE" are one mode. An unrecognized name is refused with a warning and the block paints normally. See the blend-modes sample. One of: normal, multiply, screen, overlay, darken, lighten, colorDodge, colorBurn, hardLight, softLight, difference, exclusion, hue, saturation, ...
breakBefore	boolean		Pagination: open a new page before this block - a part divider, an appendix. Ignored when the page is already empty, so it never leaves a blank page behind.
changed	boolean		Revision / change bar: draw a vertical bar in the page margin beside this block's content to flag it as changed. A document-level changeBars object customizes the bar (width / color / side / distance); a condition tag declared with changeBar:true sets this automatically for tagged content.

Key	Type	Default	Description
`clip`	`boolean`		Clip this block's children (on a container) or its own ink to its box, with the style's corner "radius". Only bites where content is BIGGER than the box: a flow container grows to fit its children vertically, so give it "position" with a height, or a "minHeight", before expecting a vertical crop. Container-scoped, like "groupOpacity": on any other flow block it is refused with a warning. For a crop to a SHAPE rather than to the box, see "clipPath"; giving both INTERSECTS them.
`clipPath`	`object`		Crop this block to an arbitrary SHAPE stated inside its box -- a design tool's FRAME. Same vocabulary as the "shape" block and read by the same outline builder, so a shape and a clipPath given identical parameters describe the identical outline. Scoped exactly as "clip" is: a positioned block of any type (it needs a "position.height", because the shape is stated inside the box) or a container in the flow; anywhere else it is refused with a warning. Naming "clip" as well INTERSECTS the two. The shape turns, scales and skews with the block that carries it.
`condition`	`string`		Conditional text: render this block only when its condition-tag expression is shown under the document's conditions map. Whitespace/comma tag terms, each optionally '!'-negated, AND-combined; an unlisted tag defaults to shown. Empty means always shown.
`content`	`string`		Pre-rendered inline SVG markup of the equation.
`float`	`string`		Float this block out of the stacked flow to one side of its column; content that FOLLOWS it wraps into the space beside it. Works on ANY block type -- a table, a container, a chart, a code listing, a figure. "center" straddles the page content width, so text in every column wraps around it. The float box takes the block's own "width"; a block that declares none takes HALF the column, so a float always leaves room beside it. Ignored on a block that also sets "position" or "anchor" -- those have already chosen their placement. Not to be confused with "anchor": { "float": true }, a BOOLEAN about migrating an anchored frame to the next column that can hold it. One of: `left`, `right`, `center`.
`groupBlend`	`string`		Composite this block AND its children once, then blend the RESULT with the page. NOT the same key as "blend", and the difference is larger than it is for opacity: two stacked children that each multiply the backdrop multiply it TWICE, which is darker than the group multiplying once, and no per-child setting reproduces it. That is what a design tool's blend-mode control on a group means. Container-scoped: on any other flow block it is refused with a warning. Also the way to blend an element together with its drop shadow, which is drawn as stacked alpha layers and so keeps its own normal compositing. See the blend-modes sample. One of: `normal`, `multiply`, `screen`, `overlay`, `darken`, `lighten`, `colorDodge`, `colorBurn`, `hardLight`, `softLight`, `difference`, `exclusion`, `hue`, `saturation`, ...

Key	Type	Default	Description
groupOpacity	number		Composite this block AND its children once, then fade the result -- true GROUP opacity. NOT the same key as "opacity", which fades each element as it paints: two overlapping opaque children at opacity 0.5 each show a seam where they cross, and under groupOpacity 0.5 they do not. The two compose. Container-scoped: on any other flow block it is refused with a warning. See the clipped-groups sample.
height	number		Drawn height in points.
mathML	string		Alias for "mathml".
mathml	string		Presentation MathML source. Embedded on the <Formula> element as a /Supplement associated file.
mirrorH	boolean | string		Mirror the whole block left-to-right about the centre of its "position" box, under the same restriction as "angle". Identical to scale x -1, and the two compose. NOT "flipH": that key belongs to the image block, where it mirrors the PICTURE INSIDE its frame. The two are separate words so an image can do both. A STRING is a data token, so each record of a template mirrors its own sticker: "{{s.flip}}" takes the record's boolean, or text "true" / "false" in any case, or 1 / 0 (a number or text); a record without it draws the block unmirrored, silently, and any other value ("yes" included) is reported with the block's pointer (and the -merge record) and drawn unmirrored. A string with no token is read as it stands.
mirrorV	boolean | string		Mirror the whole block top-to-bottom about the centre of its "position" box, under the same restriction as "angle". Identical to scale y -1. NOT "flipV" -- see "mirrorH". A STRING is a data token, so each record of a template mirrors its own sticker: "{{s.flip}}" takes the record's boolean, or text "true" / "false" in any case, or 1 / 0 (a number or text); a record without it draws the block unmirrored, silently, and any other value ("yes" included) is reported with the block's pointer (and the -merge record) and drawn unmirrored. A string with no token is read as it stands.
number	string		Pin a literal number/label for this block. Does not advance the counter.
numbered	boolean	true	Take part in automatic equation numbering when the document enables it. false leaves this one unnumbered and does not advance the counter.
opacity	number		Constant opacity for this block: 0 (invisible) to 1 (opaque, the default). REAL alpha -- a /CA /ca ExtGState in the PDF -- so a translucent panel over a photograph shows the photograph through it. Not to be confused with "style.background" {color, alpha}, which composites the tint toward WHITE at parse time and so only looks right over a white page. PER-ELEMENT, NOT GROUP: on a container it paints that container's own decoration -- background, border, accent bar, stripe -- and never its children, which draw at their own opacity (1 unless they set one). Two translucent shapes that overlap therefore composite twice, which is what per-element opacity means; group opacity, where a whole subtree is composited once and then faded, is a separate feature. Where a draw already carries its own alpha (an SVG fill-opacity, a gradient's alpha) the two MULTIPLY.

Key	Type	Default	Description
repeatAt	array of object		Imposition: draw this block ONCE and place the drawing again at each of these offsets, in points from the block's own position, y DOWNWARD as every position is. One entry = one extra copy, so a list of 7 puts 8 cards on a sheet. The drawing is captured into a Form XObject and each offset is a cm + Do of that one stream, so n copies cost one copy of the content. Translation only -- angle / scale / skew are not applied per placement. A REPEATED FORM MAY NOT CARRY TAGGED CONTENT: ISO 14289-1 7.20 allows a form's content ONE semantic parent, so repeating tagged text (or a shape with alt) fails PDF/UA -- veraPDF catches it, ap_addstruct's audit does not. The engine warns and draws it anyway. Repeat decoration: a shape or svg with no alt, or put the block in a master's frames, which is drawn as a page artifact. Only a block with position is drawn again: on a block without one (in the flow, or an anchored frame), or on a repeat / if / include block, the key is refused with a notice naming the block. Each copy takes the block's groupOpacity and groupBlend. The layout map (-map) reports the copies as repeats on the block's fragment, so a design tool can select and outline them.
scale	number | array of number | object | string		Scale the block about the same pivot as "angle" (the centre of its "position" box), and under the same restriction. A CONTAINER in the flow may also carry this: it becomes a GROUP that turns about its own measured box and reserves the bounding box of the turned result, and its children inherit the transform. Any other flow block ignores it with a warning -- wrap it in a container. A number scales both axes; [x, y] or {"x":, "y":} scales them separately. 1 = natural size. Applied AFTER layout, so a scaled block does NOT re-wrap its text at the scaled measure -- this is a design tool's scale handle, not a change of font size. A STRING is a data token, so each record of a template scales its own sticker: "{{s.scale}}" takes the record's number, [x, y] or {"x":, "y":} (members may be text of numbers), or text of one or two plain numbers such as a CSV field ("1.5", "1.5 0.8", "[1.5, 0.8]"); a record without it draws the block at its natural size, silently, and a value that is not a number or a pair is reported with the block's pointer (and the -merge record) and drawn at natural size. A string with no token is read as it stands.
skew	number | array of number | object | string		Shear the block, in DEGREES, about the same pivot as "angle", under the same restriction. [x, y] or {"x":, "y":}; a bare number skews x only. Positive x leans the top to the right and positive y lifts the right-hand side -- both counter-clockwise, matching "angle". Clamped to +/-89.9: a 90-degree skew is a singular matrix and draws nothing. A STRING is a data token, so each record of a template skews its own sticker: "{{s.skew}}" takes the record's number (x only), [x, y] or {"x":, "y":}, or text of one or two plain numbers; clamped as the written one is. A record without it draws the block unskewed, silently, and a value that is not a number or a pair is reported with the block's pointer (and the -merge record) and drawn unskewed. A string with no token is read as it stands.
src	string		Path to a pre-rendered SVG file.

Key	Type	Default	Description
straddle	boolean		Straddle across columns: when true and the body is multi-column, this block spans the FULL content width and text in every column wraps around it (above and below) - a spanning float generalized from center-floated images to any block. No effect in a single-column body, on a positioned / anchored block, or on an image/pdf already using float:"center" (the same mechanism). Does not balance the columns above it or act as a section divider.
svg	string		Alias for "content".
verticalAlign	string		Where the content sits in its "position" box when it is shorter than the box's height: "top" (the default), "middle" ("center" is the same) or "bottom". A paragraph, heading or pre moves as a whole (its own box with it); a container keeps its box where it is and moves its children inside it (a child with its own "position" stays put). The offset is the box height less the content's height as it is actually placed -- the same lines, shrinkToFit size and padding that are then drawn -- halved for middle. Content as tall as the box or taller does not move: it is clipped and reported exactly as without the key. Needs a position with a height: on a block without one, on any other kind of block, on text set along a textPath "d" curve (stated in fractions of the whole box, so drawn across all of it; the "arc" form aligns), and on vertical text whose columns fill the box (columns run the full height unless a paragraph's or a heading's boxHeight is shorter), it is refused with a notice and ignored. The layout map keeps reporting the box. One of: top, middle, bottom, center.
width	dimension		Drawn width - points, a percentage of the column, or "auto".
z	number		CANVAS MODE ONLY (see the top-level "pages"): where this element sits in the PAINT order of its page. Lower draws first, further back; unset means "wherever I am in the list", and ties keep author order. It moves the PAINT and not the READING: the structure tree keeps the document order, so a caption written first because it is read first can still be painted last because it sits on top. That separation is what a flattened scene graph cannot express. Ignored outside canvas mode, where the flow already fixes both orders.

The mermaid element

Type name: mermaid. Accepted aliases: diagram.

A diagram written as code. Supply the rendered SVG and the engine places it as vector art; without one it degrades to a captioned panel showing the diagram source, so the information is never lost.

Example

```
{
  "type": "mermaid",
  "src": "flow.svg",
  "source": "graph TD; A-->B;",
  "alt": "A two-node build pipeline."
}
```

Keys

Key	Type	Default	Description
alt	string		Alternate text (/Alt on the tagged Figure).
anchor	anchor		-
angle	number | string		Turn the block, in degrees counter-clockwise, about the CENTRE of its "position" box. A CONTAINER in the flow may also carry this: it becomes a GROUP that turns about its own measured box and reserves the bounding box of the turned result, and its children inherit the transform. Any other flow block ignores it with a warning -- wrap it in a container. The block lays out upright inside its box and the finished result is turned, so a "position.height" that truncates content truncates the same text it would unrotated, and the turned result is NOT clipped to its box: it may extend past it, exactly as an unrotated positioned block may. Not to be confused with the image block's "rotate", which is an integer QUARTER TURN CLOCKWISE applied to the picture INSIDE its box (and re-sizes that box to the turned orientation). The two compose -- an image can stand a sideways scan upright with "rotate" and tilt the whole frame with "angle". Same meaning as the watermark's "angle". A STRING is a data token, so each record of a template turns its own sticker: "{{s.angle}}" takes the record's number (or text of one, such as a CSV field); a record without it leaves the block upright, silently, and a value that is not a number is reported with the block's pointer (and the -merge record) and drawn upright. A string with no token is read as it stands.
bleed	number		Step and Repeat: how far this placed block's box reaches past its TRIM on every side, in points -- the box IS the bleed box, and the trim is the box inset by this much. A design tool sizes a card by it (the trim plus twice the bleed) and draws the trim line inside each copy; the engine draws nothing from it, and keeps the bleed only through the block's own clip. Only a block with position has a trim: on any other block the key is refused with a notice naming the block.
blend	string		How this block's OWN ink combines with the page under it -- the sixteen PDF blend modes (ISO 32000-1 11.3.5), written as /BM. On a container that means its background, border and stripe, never its children; for the subtree as one object see "groupBlend". Case and separators are folded, so "colorDodge", "color-dodge" and "COLORDODGE" are one mode. An unrecognized name is refused with a warning and the block paints normally. See the blend-modes sample. One of: normal, multiply, screen, overlay, darken, lighten, colorDodge, colorBurn, hardLight, softLight, difference, exclusion, hue, saturation, ...
breakBefore	boolean		Pagination: open a new page before this block - a part divider, an appendix. Ignored when the page is already empty, so it never leaves a blank page behind.
changed	boolean		Revision / change bar: draw a vertical bar in the page margin beside this block's content to flag it as changed. A document-level changeBars object customizes the bar (width / color / side / distance); a condition tag declared with changeBar:true sets this automatically for tagged content.

Key	Type	Default	Description
`clip`	`boolean`		Clip this block's children (on a container) or its own ink to its box, with the style's corner "radius". Only bites where content is BIGGER than the box: a flow container grows to fit its children vertically, so give it "position" with a height, or a "minHeight", before expecting a vertical crop. Container-scoped, like "groupOpacity": on any other flow block it is refused with a warning. For a crop to a SHAPE rather than to the box, see "clipPath"; giving both INTERSECTS them.
`clipPath`	`object`		Crop this block to an arbitrary SHAPE stated inside its box -- a design tool's FRAME. Same vocabulary as the "shape" block and read by the same outline builder, so a shape and a clipPath given identical parameters describe the identical outline. Scoped exactly as "clip" is: a positioned block of any type (it needs a "position.height", because the shape is stated inside the box) or a container in the flow; anywhere else it is refused with a warning. Naming "clip" as well INTERSECTS the two. The shape turns, scales and skews with the block that carries it.
`condition`	`string`		Conditional text: render this block only when its condition-tag expression is shown under the document's conditions map. Whitespace/comma tag terms, each optionally '!'-negated, AND-combined; an unlisted tag defaults to shown. Empty means always shown.
`content`	`string`		Pre-rendered inline SVG markup of the diagram.
`float`	`string`		Float this block out of the stacked flow to one side of its column; content that FOLLOWS it wraps into the space beside it. Works on ANY block type -- a table, a container, a chart, a code listing, a figure. "center" straddles the page content width, so text in every column wraps around it. The float box takes the block's own "width"; a block that declares none takes HALF the column, so a float always leaves room beside it. Ignored on a block that also sets "position" or "anchor" -- those have already chosen their placement. Not to be confused with "anchor": { "float": true }, a BOOLEAN about migrating an anchored frame to the next column that can hold it. One of: `left`, `right`, `center`.
`groupBlend`	`string`		Composite this block AND its children once, then blend the RESULT with the page. NOT the same key as "blend", and the difference is larger than it is for opacity: two stacked children that each multiply the backdrop multiply it TWICE, which is darker than the group multiplying once, and no per-child setting reproduces it. That is what a design tool's blend-mode control on a group means. Container-scoped: on any other flow block it is refused with a warning. Also the way to blend an element together with its drop shadow, which is drawn as stacked alpha layers and so keeps its own normal compositing. See the blend-modes sample. One of: `normal`, `multiply`, `screen`, `overlay`, `darken`, `lighten`, `colorDodge`, `colorBurn`, `hardLight`, `softLight`, `difference`, `exclusion`, `hue`, `saturation`, ...

Key	Type	Default	Description
groupOpacity	number		Composite this block AND its children once, then fade the result -- true GROUP opacity. NOT the same key as "opacity", which fades each element as it paints: two overlapping opaque children at opacity 0.5 each show a seam where they cross, and under groupOpacity 0.5 they do not. The two compose. Container-scoped: on any other flow block it is refused with a warning. See the clipped-groups sample.
height	number		Drawn height in points.
mirrorH	boolean | string		Mirror the whole block left-to-right about the centre of its "position" box, under the same restriction as "angle". Identical to scale x -1, and the two compose. NOT "flipH": that key belongs to the image block, where it mirrors the PICTURE INSIDE its frame. The two are separate words so an image can do both. A STRING is a data token, so each record of a template mirrors its own sticker: "{{s.flip}}" takes the record's boolean, or text "true" / "false" in any case, or 1 / 0 (a number or text); a record without it draws the block unmirrored, silently, and any other value ("yes" included) is reported with the block's pointer (and the -merge record) and drawn unmirrored. A string with no token is read as it stands.
mirrorV	boolean | string		Mirror the whole block top-to-bottom about the centre of its "position" box, under the same restriction as "angle". Identical to scale y -1. NOT "flipV" -- see "mirrorH". A STRING is a data token, so each record of a template mirrors its own sticker: "{{s.flip}}" takes the record's boolean, or text "true" / "false" in any case, or 1 / 0 (a number or text); a record without it draws the block unmirrored, silently, and any other value ("yes" included) is reported with the block's pointer (and the -merge record) and drawn unmirrored. A string with no token is read as it stands.
opacity	number		Constant opacity for this block: 0 (invisible) to 1 (opaque, the default). REAL alpha -- a /CA /ca ExtGState in the PDF -- so a translucent panel over a photograph shows the photograph through it. Not to be confused with "style.background" {color, alpha}, which composites the tint toward WHITE at parse time and so only looks right over a white page. PER-ELEMENT, NOT GROUP: on a container it paints that container's own decoration -- background, border, accent bar, stripe -- and never its children, which draw at their own opacity (1 unless they set one). Two translucent shapes that overlap therefore composite twice, which is what per-element opacity means; group opacity, where a whole subtree is composited once and then faded, is a separate feature. Where a draw already carries its own alpha (an SVG fill-opacity, a gradient's alpha) the two MULTIPLY.

Key	Type	Default	Description
repeatAt	array of object		Imposition: draw this block ONCE and place the drawing again at each of these offsets, in points from the block's own position, y DOWNWARD as every position is. One entry = one extra copy, so a list of 7 puts 8 cards on a sheet. The drawing is captured into a Form XObject and each offset is a cm + Do of that one stream, so n copies cost one copy of the content. Translation only -- angle / scale / skew are not applied per placement. A REPEATED FORM MAY NOT CARRY TAGGED CONTENT: ISO 14289-1 7.20 allows a form's content ONE semantic parent, so repeating tagged text (or a shape with alt) fails PDF/UA -- veraPDF catches it, ap_addstruct's audit does not. The engine warns and draws it anyway. Repeat decoration: a shape or svg with no alt, or put the block in a master's frames, which is drawn as a page artifact. Only a block with position is drawn again: on a block without one (in the flow, or an anchored frame), or on a repeat / if / include block, the key is refused with a notice naming the block. Each copy takes the block's groupOpacity and groupBlend. The layout map (-map) reports the copies as repeats on the block's fragment, so a design tool can select and outline them.
scale	number | array of number | object | string		Scale the block about the same pivot as "angle" (the centre of its "position" box), and under the same restriction. A CONTAINER in the flow may also carry this: it becomes a GROUP that turns about its own measured box and reserves the bounding box of the turned result, and its children inherit the transform. Any other flow block ignores it with a warning -- wrap it in a container. A number scales both axes; [x, y] or {"x":, "y":} scales them separately. 1 = natural size. Applied AFTER layout, so a scaled block does NOT re-wrap its text at the scaled measure -- this is a design tool's scale handle, not a change of font size. A STRING is a data token, so each record of a template scales its own sticker: "{{s.scale}}" takes the record's number, [x, y] or {"x":, "y":} (members may be text of numbers), or text of one or two plain numbers such as a CSV field ("1.5", "1.5 0.8", "[1.5, 0.8]"); a record without it draws the block at its natural size, silently, and a value that is not a number or a pair is reported with the block's pointer (and the -merge record) and drawn at natural size. A string with no token is read as it stands.
skew	number | array of number | object | string		Shear the block, in DEGREES, about the same pivot as "angle", under the same restriction. [x, y] or {"x":, "y":}; a bare number skews x only. Positive x leans the top to the right and positive y lifts the right-hand side -- both counter-clockwise, matching "angle". Clamped to +/-89.9: a 90-degree skew is a singular matrix and draws nothing. A STRING is a data token, so each record of a template skews its own sticker: "{{s.skew}}" takes the record's number (x only), [x, y] or {"x":, "y":}, or text of one or two plain numbers; clamped as the written one is. A record without it draws the block unskewed, silently, and a value that is not a number or a pair is reported with the block's pointer (and the -merge record) and drawn unskewed. A string with no token is read as it stands.
source	string		The diagram source (e.g. mermaid text). Shown as a code panel when no rendered SVG is supplied.
src	string		Path to a pre-rendered SVG file.

Key	Type	Default	Description
straddle	boolean		Straddle across columns: when true and the body is multi-column, this block spans the FULL content width and text in every column wraps around it (above and below) - a spanning float generalized from center-floated images to any block. No effect in a single-column body, on a positioned / anchored block, or on an image/pdf already using float:"center" (the same mechanism). Does not balance the columns above it or act as a section divider.
svg	string		Alias for "content".
verticalAlign	string		Where the content sits in its "position" box when it is shorter than the box's height: "top" (the default), "middle" ("center" is the same) or "bottom". A paragraph, heading or pre moves as a whole (its own box with it); a container keeps its box where it is and moves its children inside it (a child with its own "position" stays put). The offset is the box height less the content's height as it is actually placed -- the same lines, shrinkToFit size and padding that are then drawn -- halved for middle. Content as tall as the box or taller does not move: it is clipped and reported exactly as without the key. Needs a position with a height: on a block without one, on any other kind of block, on text set along a textPath "d" curve (stated in fractions of the whole box, so drawn across all of it; the "arc" form aligns), and on vertical text whose columns fill the box (columns run the full height unless a paragraph's or a heading's boxHeight is shorter), it is refused with a notice and ignored. The layout map keeps reporting the box. One of: top, middle, bottom, center.
width	dimension		Drawn width - points, a percentage of the column, or "auto".
z	number		CANVAS MODE ONLY (see the top-level "pages"): where this element sits in the PAINT order of its page. Lower draws first, further back; unset means "wherever I am in the list", and ties keep author order. It moves the PAINT and not the READING: the structure tree keeps the document order, so a caption written first because it is read first can still be painted last because it sits on top. That separation is what a flattened scene graph cannot express. Ignored outside canvas mode, where the flow already fixes both orders.

9. Placing existing PDFs

Importing pages from PDF files you already have.

The pdf element

Type name: pdf.

One or more pages lifted from an existing PDF and placed into the flow as real vector content - not a screenshot. Use it to fold a supplied datasheet, certificate or signed page into a generated document.

Example

```
{
  "type": "pdf",
  "src": "datasheet.pdf",
  "page": 1,
  "width": "100%",
  "caption": "Supplied datasheet, page 1."
}
```

Keys

Key	Type	Default	Description
`alt`	`string`		Alternate text (/Alt on the tagged Figure).
`anchor`	`anchor`		-
`angle`	`number | string`		Turn the block, in degrees counter-clockwise, about the CENTRE of its "position" box. A CONTAINER in the flow may also carry this: it becomes a GROUP that turns about its own measured box and reserves the bounding box of the turned result, and its children inherit the transform. Any other flow block ignores it with a warning -- wrap it in a container. The block lays out upright inside its box and the finished result is turned, so a "position.height" that truncates content truncates the same text it would unrotated, and the turned result is NOT clipped to its box: it may extend past it, exactly as an unrotated positioned block may. Not to be confused with the image block's "rotate", which is an integer QUARTER TURN CLOCKWISE applied to the picture INSIDE its box (and re-sizes that box to the turned orientation). The two compose -- an image can stand a sideways scan upright with "rotate" and tilt the whole frame with "angle". Same meaning as the watermark's "angle". A STRING is a data token, so each record of a template turns its own sticker: "{{s.angle}}" takes the record's number (or text of one, such as a CSV field); a record without it leaves the block upright, silently, and a value that is not a number is reported with the block's pointer (and the -merge record) and drawn upright. A string with no token is read as it stands.
`bleed`	`number`		Step and Repeat: how far this placed block's box reaches past its TRIM on every side, in points -- the box IS the bleed box, and the trim is the box inset by this much. A design tool sizes a card by it (the trim plus twice the bleed) and draws the trim line inside each copy; the engine draws nothing from it, and keeps the bleed only through the block's own `clip`. Only a block with `position` has a trim: on any other block the key is refused with a notice naming the block.

Key	Type	Default	Description
`blend`	`string`		How this block's OWN ink combines with the page under it -- the sixteen PDF blend modes (ISO 32000-1 11.3.5), written as /BM. On a container that means its background, border and stripe, never its children; for the subtree as one object see "groupBlend". Case and separators are folded, so "colorDodge", "color-dodge" and "COLORDODGE" are one mode. An unrecognized name is refused with a warning and the block paints normally. See the blend-modes sample. One of: `normal`, `multiply`, `screen`, `overlay`, `darken`, `lighten`, `colorDodge`, `colorBurn`, `hardLight`, `softLight`, `difference`, `exclusion`, `hue`, `saturation`, ...
`breakBefore`	`boolean`		Pagination: open a new page before this block - a part divider, an appendix. Ignored when the page is already empty, so it never leaves a blank page behind.
`caption`	`string | runs`		Caption printed under the placed page. A caption also enrolls it in the Figure counter and gives a cross-reference something to resolve against. Accepts a plain string, an ARRAY OF RUNS, or a SINGLE RUN OBJECT - the same run forms a paragraph's "runs" takes - so a caption can carry per-span styling, and above all a per-span "fontFamily": a caption that mixes scripts sets each span in the face that suits it instead of flattening to one. The flat text is always derived from the runs and stays the entry of record: it is what reaches the list of figures, the /Alt and the structure tree. A number ("Figure 3-2.") is prepended as the first run, unstyled, so it takes the block's own face.
`changed`	`boolean`		Revision / change bar: draw a vertical bar in the page margin beside this block's content to flag it as changed. A document-level changeBars object customizes the bar (width / color / side / distance); a condition tag declared with changeBar:true sets this automatically for tagged content.
`clip`	`boolean`		Clip this block's children (on a container) or its own ink to its box, with the style's corner "radius". Only bites where content is BIGGER than the box: a flow container grows to fit its children vertically, so give it "position" with a height, or a "minHeight", before expecting a vertical crop. Container-scoped, like "groupOpacity": on any other flow block it is refused with a warning. For a crop to a SHAPE rather than to the box, see "clipPath"; giving both INTERSECTS them.
`clipPath`	`object`		Crop this block to an arbitrary SHAPE stated inside its box -- a design tool's FRAME. Same vocabulary as the "shape" block and read by the same outline builder, so a shape and a clipPath given identical parameters describe the identical outline. Scoped exactly as "clip" is: a positioned block of any type (it needs a "position.height", because the shape is stated inside the box) or a container in the flow; anywhere else it is refused with a warning. Naming "clip" as well INTERSECTS the two. The shape turns, scales and skews with the block that carries it.
`condition`	`string`		Conditional text: render this block only when its condition-tag expression is shown under the document's conditions map. Whitespace/comma tag terms, each optionally '!'-negated, AND-combined; an unlisted tag defaults to shown. Empty means always shown.

Key	Type	Default	Description
`fit`	`string`		How the placed page fills its box: "contain" fits it whole, "cover" crops to fill, "fill" stretches, "none" draws it at its natural size - the source page's own CropBox. "cover" and "none" are cropped to the box; the page is centred in it unless "focus" aims the crop (and "zoom" magnifies a cover). One of: `contain`, `cover`, `fill`, `none`.
`float`	`string`		Float this block out of the stacked flow to one side of its column; content that FOLLOWS it wraps into the space beside it. Works on ANY block type -- a table, a container, a chart, a code listing, a figure. "center" straddles the page content width, so text in every column wraps around it. The float box takes the block's own "width"; a block that declares none takes HALF the column, so a float always leaves room beside it. Ignored on a block that also sets "position" or "anchor" -- those have already chosen their placement. Not to be confused with "anchor": { "float": true }, a BOOLEAN about migrating an anchored frame to the next column that can hold it. One of: `left`, `right`, `center`.
`focus`	`string | number | array of number | object`		Where a cropping fit aims, as an image's "focus": [x, y] (or {"x":, "y":}, or one number for both axes) as fractions of the placed page, 0 = left / top, 1 = right / bottom. The point is brought to the centre of the box as far as the page allows; the crop never pulls away from an edge, so [0, 0] and [1, 1] reach the corners -- [1, 0] keeps the top-right corner, which is how a whole-envelope postage page is placed at 100% ("fit": "none") with only its corner on a card -- and the default [0.5, 0.5] is the centred crop. Only "cover" and "none" crop; on any other fit it is ignored with a warning. Values outside 0..1 are clamped. A STRING is a data token, as an image's: "{{cards.0.postageFocus}}" resolves to the whole value at that path -- a number, [x, y], {x, y}, or text of numbers -- and a field the record lacks leaves the centred crop, silently. A bound value that is not numbers, or outside 0..1, is reported with the block's pointer (and the -merge record). A string with no token is read as it stands.
`groupBlend`	`string`		Composite this block AND its children once, then blend the RESULT with the page. NOT the same key as "blend", and the difference is larger than it is for opacity: two stacked children that each multiply the backdrop multiply it TWICE, which is darker than the group multiplying once, and no per-child setting reproduces it. That is what a design tool's blend-mode control on a group means. Container-scoped: on any other flow block it is refused with a warning. Also the way to blend an element together with its drop shadow, which is drawn as stacked alpha layers and so keeps its own normal compositing. See the blend-modes sample. One of: `normal`, `multiply`, `screen`, `overlay`, `darken`, `lighten`, `colorDodge`, `colorBurn`, `hardLight`, `softLight`, `difference`, `exclusion`, `hue`, `saturation`, ...
`groupOpacity`	`number`		Composite this block AND its children once, then fade the result -- true GROUP opacity. NOT the same key as "opacity", which fades each element as it paints: two overlapping opaque children at opacity 0.5 each show a seam where they cross, and under groupOpacity 0.5 they do not. The two compose. Container-scoped: on any other flow block it is refused with a warning. See the clipped-groups sample.

Key	Type	Default	Description
height	dimension		Drawn height - points, a percentage, or "auto" to keep the aspect ratio.
maxHeight	number		Cap on the drawn height in points, applied after scaling.
mirrorH	boolean | string		Mirror the whole block left-to-right about the centre of its "position" box, under the same restriction as "angle". Identical to scale x -1, and the two compose. NOT "flipH": that key belongs to the image block, where it mirrors the PICTURE INSIDE its frame. The two are separate words so an image can do both. A STRING is a data token, so each record of a template mirrors its own sticker: "{{s.flip}}" takes the record's boolean, or text "true" / "false" in any case, or 1 / 0 (a number or text); a record without it draws the block unmirrored, silently, and any other value ("yes" included) is reported with the block's pointer (and the -merge record) and drawn unmirrored. A string with no token is read as it stands.
mirrorV	boolean | string		Mirror the whole block top-to-bottom about the centre of its "position" box, under the same restriction as "angle". Identical to scale y -1. NOT "flipV" -- see "mirrorH". A STRING is a data token, so each record of a template mirrors its own sticker: "{{s.flip}}" takes the record's boolean, or text "true" / "false" in any case, or 1 / 0 (a number or text); a record without it draws the block unmirrored, silently, and any other value ("yes" included) is reported with the block's pointer (and the -merge record) and drawn unmirrored. A string with no token is read as it stands.
number	string		Pin a literal number/label for this block. Does not advance the counter.
numbered	boolean	true	Take part in automatic caption numbering when the document enables it. false leaves this one unnumbered and does not advance the counter.
opacity	number		Constant opacity for this block: 0 (invisible) to 1 (opaque, the default). REAL alpha -- a /CA /ca ExtGState in the PDF -- so a translucent panel over a photograph shows the photograph through it. Not to be confused with "style.background" {color, alpha}, which composites the tint toward WHITE at parse time and so only looks right over a white page. PER-ELEMENT, NOT GROUP: on a container it paints that container's own decoration -- background, border, accent bar, stripe -- and never its children, which draw at their own opacity (1 unless they set one). Two translucent shapes that overlap therefore composite twice, which is what per-element opacity means; group opacity, where a whole subtree is composited once and then faded, is a separate feature. Where a draw already carries its own alpha (an SVG fill-opacity, a gradient's alpha) the two MULTIPLY.
page	integer		Which page of the source PDF to place, 1-based. Default 1.

Key	Type	Default	Description
repeatAt	array of object		Imposition: draw this block ONCE and place the drawing again at each of these offsets, in points from the block's own position, y DOWNWARD as every position is. One entry = one extra copy, so a list of 7 puts 8 cards on a sheet. The drawing is captured into a Form XObject and each offset is a cm + Do of that one stream, so n copies cost one copy of the content. Translation only -- angle / scale / skew are not applied per placement. A REPEATED FORM MAY NOT CARRY TAGGED CONTENT: ISO 14289-1 7.20 allows a form's content ONE semantic parent, so repeating tagged text (or a shape with alt) fails PDF/UA -- veraPDF catches it, ap_addstruct's audit does not. The engine warns and draws it anyway. Repeat decoration: a shape or svg with no alt, or put the block in a master's frames, which is drawn as a page artifact. Only a block with position is drawn again: on a block without one (in the flow, or an anchored frame), or on a repeat / if / include block, the key is refused with a notice naming the block. Each copy takes the block's groupOpacity and groupBlend. The layout map (-map) reports the copies as repeats on the block's fragment, so a design tool can select and outline them.
scale	number | array of number | object | string		Scale the block about the same pivot as "angle" (the centre of its "position" box), and under the same restriction. A CONTAINER in the flow may also carry this: it becomes a GROUP that turns about its own measured box and reserves the bounding box of the turned result, and its children inherit the transform. Any other flow block ignores it with a warning -- wrap it in a container. A number scales both axes; [x, y] or {"x":, "y":} scales them separately. 1 = natural size. Applied AFTER layout, so a scaled block does NOT re-wrap its text at the scaled measure -- this is a design tool's scale handle, not a change of font size. A STRING is a data token, so each record of a template scales its own sticker: "{{s.scale}}" takes the record's number, [x, y] or {"x":, "y":} (members may be text of numbers), or text of one or two plain numbers such as a CSV field ("1.5", "1.5 0.8", "[1.5, 0.8]"); a record without it draws the block at its natural size, silently, and a value that is not a number or a pair is reported with the block's pointer (and the -merge record) and drawn at natural size. A string with no token is read as it stands.
skew	number | array of number | object | string		Shear the block, in DEGREES, about the same pivot as "angle", under the same restriction. [x, y] or {"x":, "y":}; a bare number skews x only. Positive x leans the top to the right and positive y lifts the right-hand side -- both counter-clockwise, matching "angle". Clamped to +/-89.9: a 90-degree skew is a singular matrix and draws nothing. A STRING is a data token, so each record of a template skews its own sticker: "{{s.skew}}" takes the record's number (x only), [x, y] or {"x":, "y":}, or text of one or two plain numbers; clamped as the written one is. A record without it draws the block unskewed, silently, and a value that is not a number or a pair is reported with the block's pointer (and the -merge record) and drawn unskewed. A string with no token is read as it stands.
src	string		Source PDF; the page is imported as a vector Form XObject. For PDF/UA output its fonts must be embedded.

Key	Type	Default	Description
straddle	boolean		Straddle across columns: when true and the body is multi-column, this block spans the FULL content width and text in every column wraps around it (above and below) - a spanning float generalized from center-floated images to any block. No effect in a single-column body, on a positioned / anchored block, or on an image/pdf already using float:"center" (the same mechanism). Does not balance the columns above it or act as a section divider.
verticalAlign	string		Where the content sits in its "position" box when it is shorter than the box's height: "top" (the default), "middle" ("center" is the same) or "bottom". A paragraph, heading or pre moves as a whole (its own box with it); a container keeps its box where it is and moves its children inside it (a child with its own "position" stays put). The offset is the box height less the content's height as it is actually placed -- the same lines, shrinkToFit size and padding that are then drawn -- halved for middle. Content as tall as the box or taller does not move: it is clipped and reported exactly as without the key. Needs a position with a height: on a block without one, on any other kind of block, on text set along a textPath "d" curve (stated in fractions of the whole box, so drawn across all of it; the "arc" form aligns), and on vertical text whose columns fill the box (columns run the full height unless a paragraph's or a heading's boxHeight is shorter), it is refused with a notice and ignored. The layout map keeps reporting the box. One of: top, middle, bottom, center.
width	dimension		Drawn width - points, a percentage of the column, or "auto".
z	number		CANVAS MODE ONLY (see the top-level "pages"): where this element sits in the PAINT order of its page. Lower draws first, further back; unset means "wherever I am in the list", and ties keep author order. It moves the PAINT and not the READING: the structure tree keeps the document order, so a caption written first because it is read first can still be painted last because it sits on top. That separation is what a flattened scene graph cannot express. Ignored outside canvas mode, where the flow already fixes both orders.
zoom	number | string	1	"cover" only: magnify the placed page past the size that just covers the box, keeping "focus" in view (1 = plain cover, 2 = twice as close). The box does not change; only more of the page is cropped away. On any other fit it is ignored with a warning. A STRING is a data token, as for focus; a field the record lacks leaves plain cover, and a value that is not a number, or below 1, is reported.

The pdf-contact-sheet element

Type name: pdf-contact-sheet. Accepted aliases: contact-sheet.

A grid of thumbnails of every page in a PDF, with optional page numbers. A quick visual index of a large document.

Example

```
{
  "type": "pdf-contact-sheet",
  "src": "report.pdf",
  "columns": 4,
  "labels": true
}
```

Keys

Key	Type	Default	Description
`alt`	`string`		Alternate text (/Alt on the tagged Figure).
`anchor`	`anchor`		-
`angle`	`number | string`		Turn the block, in degrees counter-clockwise, about the CENTRE of its "position" box. A CONTAINER in the flow may also carry this: it becomes a GROUP that turns about its own measured box and reserves the bounding box of the turned result, and its children inherit the transform. Any other flow block ignores it with a warning -- wrap it in a container. The block lays out upright inside its box and the finished result is turned, so a "position.height" that truncates content truncates the same text it would unrotated, and the turned result is NOT clipped to its box: it may extend past it, exactly as an unrotated positioned block may. Not to be confused with the image block's "rotate", which is an integer QUARTER TURN CLOCKWISE applied to the picture INSIDE its box (and re-sizes that box to the turned orientation). The two compose -- an image can stand a sideways scan upright with "rotate" and tilt the whole frame with "angle". Same meaning as the watermark's "angle". A STRING is a data token, so each record of a template turns its own sticker: "{{s.angle}}" takes the record's number (or text of one, such as a CSV field); a record without it leaves the block upright, silently, and a value that is not a number is reported with the block's pointer (and the -merge record) and drawn upright. A string with no token is read as it stands.
`bleed`	`number`		Step and Repeat: how far this placed block's box reaches past its TRIM on every side, in points -- the box IS the bleed box, and the trim is the box inset by this much. A design tool sizes a card by it (the trim plus twice the bleed) and draws the trim line inside each copy; the engine draws nothing from it, and keeps the bleed only through the block's own `clip`. Only a block with `position` has a trim: on any other block the key is refused with a notice naming the block.

Key	Type	Default	Description
`blend`	`string`		How this block's OWN ink combines with the page under it -- the sixteen PDF blend modes (ISO 32000-1 11.3.5), written as /BM. On a container that means its background, border and stripe, never its children; for the subtree as one object see "groupBlend". Case and separators are folded, so "colorDodge", "color-dodge" and "COLORDODGE" are one mode. An unrecognized name is refused with a warning and the block paints normally. See the blend-modes sample. One of: `normal`, `multiply`, `screen`, `overlay`, `darken`, `lighten`, `colorDodge`, `colorBurn`, `hardLight`, `softLight`, `difference`, `exclusion`, `hue`, `saturation`, ...
`breakBefore`	`boolean`		Pagination: open a new page before this block - a part divider, an appendix. Ignored when the page is already empty, so it never leaves a blank page behind.
`clip`	`boolean`		Clip this block's children (on a container) or its own ink to its box, with the style's corner "radius". Only bites where content is BIGGER than the box: a flow container grows to fit its children vertically, so give it "position" with a height, or a "minHeight", before expecting a vertical crop. Container-scoped, like "groupOpacity": on any other flow block it is refused with a warning. For a crop to a SHAPE rather than to the box, see "clipPath"; giving both INTERSECTS them.
`clipPath`	`object`		Crop this block to an arbitrary SHAPE stated inside its box -- a design tool's FRAME. Same vocabulary as the "shape" block and read by the same outline builder, so a shape and a clipPath given identical parameters describe the identical outline. Scoped exactly as "clip" is: a positioned block of any type (it needs a "position.height", because the shape is stated inside the box) or a container in the flow; anywhere else it is refused with a warning. Naming "clip" as well INTERSECTS the two. The shape turns, scales and skews with the block that carries it.
`columns`	`integer`		How many thumbnails per row.
`condition`	`string`		Conditional text: render this block only when its condition-tag expression is shown under the document's conditions map. Whitespace/comma tag terms, each optionally '!'-negated, AND-combined; an unlisted tag defaults to shown. Empty means always shown.
`float`	`string`		Float this block out of the stacked flow to one side of its column; content that FOLLOWS it wraps into the space beside it. Works on ANY block type -- a table, a container, a chart, a code listing, a figure. "center" straddles the page content width, so text in every column wraps around it. The float box takes the block's own "width"; a block that declares none takes HALF the column, so a float always leaves room beside it. Ignored on a block that also sets "position" or "anchor" -- those have already chosen their placement. Not to be confused with "anchor": { "float": true }, a BOOLEAN about migrating an anchored frame to the next column that can hold it. One of: `left`, `right`, `center`.
`frame`	`boolean`		Draw a thin border around each thumbnail cell. (On a table "frame" is a border object instead - a different key with the same name.)
`gap`	`number`		Gap between thumbnails, in points.

Key	Type	Default	Description
groupBlend	string		Composite this block AND its children once, then blend the RESULT with the page. NOT the same key as "blend", and the difference is larger than it is for opacity: two stacked children that each multiply the backdrop multiply it TWICE, which is darker than the group multiplying once, and no per-child setting reproduces it. That is what a design tool's blend-mode control on a group means. Container-scoped: on any other flow block it is refused with a warning. Also the way to blend an element together with its drop shadow, which is drawn as stacked alpha layers and so keeps its own normal compositing. See the blend-modes sample. One of: normal, multiply, screen, overlay, darken, lighten, colorDodge, colorBurn, hardLight, softLight, difference, exclusion, hue, saturation, ...
groupOpacity	number		Composite this block AND its children once, then fade the result -- true GROUP opacity. NOT the same key as "opacity", which fades each element as it paints: two overlapping opaque children at opacity 0.5 each show a seam where they cross, and under groupOpacity 0.5 they do not. The two compose. Container-scoped: on any other flow block it is refused with a warning. See the clipped-groups sample.
items	array of string | object		Explicit cells; may mix source files.
labels	boolean		"p.N" label under each thumbnail.
mirrorH	boolean | string		Mirror the whole block left-to-right about the centre of its "position" box, under the same restriction as "angle". Identical to scale x -1, and the two compose. NOT "flipH": that key belongs to the image block, where it mirrors the PICTURE INSIDE its frame. The two are separate words so an image can do both. A STRING is a data token, so each record of a template mirrors its own sticker: "{{s.flip}}" takes the record's boolean, or text "true" / "false" in any case, or 1 / 0 (a number or text); a record without it draws the block unmirrored, silently, and any other value ("yes" included) is reported with the block's pointer (and the -merge record) and drawn unmirrored. A string with no token is read as it stands.
mirrorV	boolean | string		Mirror the whole block top-to-bottom about the centre of its "position" box, under the same restriction as "angle". Identical to scale y -1. NOT "flipV" -- see "mirrorH". A STRING is a data token, so each record of a template mirrors its own sticker: "{{s.flip}}" takes the record's boolean, or text "true" / "false" in any case, or 1 / 0 (a number or text); a record without it draws the block unmirrored, silently, and any other value ("yes" included) is reported with the block's pointer (and the -merge record) and drawn unmirrored. A string with no token is read as it stands.

Key	Type	Default	Description
`opacity`	`number`		Constant opacity for this block: 0 (invisible) to 1 (opaque, the default). REAL alpha -- a /CA /ca ExtGState in the PDF -- so a translucent panel over a photograph shows the photograph through it. Not to be confused with "style.background" {color, alpha}, which composites the tint toward WHITE at parse time and so only looks right over a white page. PER-ELEMENT, NOT GROUP: on a container it paints that container's own decoration -- background, border, accent bar, stripe -- and never its children, which draw at their own opacity (1 unless they set one). Two translucent shapes that overlap therefore composite twice, which is what per-element opacity means; group opacity, where a whole subtree is composited once and then faded, is a separate feature. Where a draw already carries its own alpha (an SVG fill-opacity, a gradient's alpha) the two MULTIPLY.
`pages`	`string | integer | array of integer`		"all", "1-5", "1,3,5", a single number, or an array of page numbers.
`repeatAt`	`array of object`		Imposition: draw this block ONCE and place the drawing again at each of these offsets, in points from the block's own position, y DOWNWARD as every `position` is. One entry = one extra copy, so a list of 7 puts 8 cards on a sheet. The drawing is captured into a Form XObject and each offset is a `cm` + `Do` of that one stream, so n copies cost one copy of the content. Translation only -- `angle` / `scale` / `skew` are not applied per placement. A REPEATED FORM MAY NOT CARRY TAGGED CONTENT: ISO 14289-1 7.20 allows a form's content ONE semantic parent, so repeating tagged text (or a shape with `alt`) fails PDF/UA -- veraPDF catches it, ap_addstruct's audit does not. The engine warns and draws it anyway. Repeat decoration: a shape or svg with no `alt`, or put the block in a master's `frames`, which is drawn as a page artifact. Only a block with `position` is drawn again: on a block without one (in the flow, or an anchored frame), or on a repeat / if / include block, the key is refused with a notice naming the block. Each copy takes the block's `groupOpacity` and `groupBlend`. The layout map (-map) reports the copies as `repeats` on the block's fragment, so a design tool can select and outline them.

Key	Type	Default	Description
`scale`	`number | array of number | object | string`		Scale the block about the same pivot as "angle" (the centre of its "position" box), and under the same restriction. A CONTAINER in the flow may also carry this: it becomes a GROUP that turns about its own measured box and reserves the bounding box of the turned result, and its children inherit the transform. Any other flow block ignores it with a warning -- wrap it in a container. A number scales both axes; [x, y] or {"x":, "y":} scales them separately. 1 = natural size. Applied AFTER layout, so a scaled block does NOT re-wrap its text at the scaled measure -- this is a design tool's scale handle, not a change of font size. A STRING is a data token, so each record of a template scales its own sticker: "{{s.scale}}" takes the record's number, [x, y] or {"x":, "y":} (members may be text of numbers), or text of one or two plain numbers such as a CSV field ("1.5", "1.5 0.8", "[1.5, 0.8]"); a record without it draws the block at its natural size, silently, and a value that is not a number or a pair is reported with the block's pointer (and the -merge record) and drawn at natural size. A string with no token is read as it stands.
`skew`	`number | array of number | object | string`		Shear the block, in DEGREES, about the same pivot as "angle", under the same restriction. [x, y] or {"x":, "y":}; a bare number skews x only. Positive x leans the top to the right and positive y lifts the right-hand side -- both counter-clockwise, matching "angle". Clamped to +/-89.9: a 90-degree skew is a singular matrix and draws nothing. A STRING is a data token, so each record of a template skews its own sticker: "{{s.skew}}" takes the record's number (x only), [x, y] or {"x":, "y":}, or text of one or two plain numbers; clamped as the written one is. A record without it draws the block unskewed, silently, and a value that is not a number or a pair is reported with the block's pointer (and the -merge record) and drawn unskewed. A string with no token is read as it stands.
`src`	`string`		Source PDF whose pages are laid out as thumbnails.
`title`	`string`		Heading printed above the sheet.
`uniformCells`	`boolean`		Give every cell the same height, sized to the tallest page, instead of letting rows vary.
`verticalAlign`	`string`		Where the content sits in its "position" box when it is shorter than the box's height: "top" (the default), "middle" ("center" is the same) or "bottom". A paragraph, heading or pre moves as a whole (its own box with it); a container keeps its box where it is and moves its children inside it (a child with its own "position" stays put). The offset is the box height less the content's height as it is actually placed -- the same lines, shrinkToFit size and padding that are then drawn -- halved for middle. Content as tall as the box or taller does not move: it is clipped and reported exactly as without the key. Needs a position with a height: on a block without one, on any other kind of block, on text set along a textPath "d" curve (stated in fractions of the whole box, so drawn across all of it; the "arc" form aligns), and on vertical text whose columns fill the box (columns run the full height unless a paragraph's or a heading's boxHeight is shorter), it is refused with a notice and ignored. The layout map keeps reporting the box. One of: `top`, `middle`, `bottom`, `center`.

Key	Type	Default	Description
z	number		CANVAS MODE ONLY (see the top-level "pages"): where this element sits in the PAINT order of its page. Lower draws first, further back; unset means "wherever I am in the list", and ties keep author order. It moves the PAINT and not the READING: the structure tree keeps the document order, so a caption written first because it is read first can still be painted last because it sits on top. That separation is what a flattened scene graph cannot express. Ignored outside canvas mode, where the flow already fixes both orders.

10. Navigation and apparatus

Generated front and back matter.

The toc element

Type name: `toc`. Accepted aliases: `lof`, `lot`, `lol`, `loc`.

An automatic table of contents built from the document's headings, with dot leaders and exact page numbers, every row an internal link. The page numbers are exact because the engine iterates: the contents list changes the pagination, so it re-counts until the numbers stop moving.

Example

```
{
  "type": "toc",
  "title": "Contents",
  "maxLevel": 2
}
```

Keys

Key	Type	Default	Description
anchor	anchor		-

Key	Type	Default	Description
angle	number | string		Turn the block, in degrees counter-clockwise, about the CENTRE of its "position" box. A CONTAINER in the flow may also carry this: it becomes a GROUP that turns about its own measured box and reserves the bounding box of the turned result, and its children inherit the transform. Any other flow block ignores it with a warning -- wrap it in a container. The block lays out upright inside its box and the finished result is turned, so a "position.height" that truncates content truncates the same text it would unrotated, and the turned result is NOT clipped to its box: it may extend past it, exactly as an unrotated positioned block may. Not to be confused with the image block's "rotate", which is an integer QUARTER TURN CLOCKWISE applied to the picture INSIDE its box (and re-sizes that box to the turned orientation). The two compose -- an image can stand a sideways scan upright with "rotate" and tilt the whole frame with "angle". Same meaning as the watermark's "angle". A STRING is a data token, so each record of a template turns its own sticker: "{{s.angle}}" takes the record's number (or text of one, such as a CSV field); a record without it leaves the block upright, silently, and a value that is not a number is reported with the block's pointer (and the -merge record) and drawn upright. A string with no token is read as it stands.
bleed	number		Step and Repeat: how far this placed block's box reaches past its TRIM on every side, in points -- the box IS the bleed box, and the trim is the box inset by this much. A design tool sizes a card by it (the trim plus twice the bleed) and draws the trim line inside each copy; the engine draws nothing from it, and keeps the bleed only through the block's own clip. Only a block with position has a trim: on any other block the key is refused with a notice naming the block.
blend	string		How this block's OWN ink combines with the page under it -- the sixteen PDF blend modes (ISO 32000-1 11.3.5), written as /BM. On a container that means its background, border and stripe, never its children; for the subtree as one object see "groupBlend". Case and separators are folded, so "colorDodge", "color-dodge" and "COLORDODGE" are one mode. An unrecognized name is refused with a warning and the block paints normally. See the blend-modes sample. One of: normal, multiply, screen, overlay, darken, lighten, colorDodge, colorBurn, hardLight, softLight, difference, exclusion, hue, saturation, ...
breakBefore	boolean		Pagination: open a new page before this block - a part divider, an appendix. Ignored when the page is already empty, so it never leaves a blank page behind.
changed	boolean		Revision / change bar: draw a vertical bar in the page margin beside this block's content to flag it as changed. A document-level changeBars object customizes the bar (width / color / side / distance); a condition tag declared with changeBar:true sets this automatically for tagged content.

Key	Type	Default	Description
`clip`	`boolean`		Clip this block's children (on a container) or its own ink to its box, with the style's corner "radius". Only bites where content is BIGGER than the box: a flow container grows to fit its children vertically, so give it "position" with a height, or a "minHeight", before expecting a vertical crop. Container-scoped, like "groupOpacity": on any other flow block it is refused with a warning. For a crop to a SHAPE rather than to the box, see "clipPath"; giving both INTERSECTS them.
`clipPath`	`object`		Crop this block to an arbitrary SHAPE stated inside its box -- a design tool's FRAME. Same vocabulary as the "shape" block and read by the same outline builder, so a shape and a clipPath given identical parameters describe the identical outline. Scoped exactly as "clip" is: a positioned block of any type (it needs a "position.height", because the shape is stated inside the box) or a container in the flow; anywhere else it is refused with a warning. Naming "clip" as well INTERSECTS the two. The shape turns, scales and skews with the block that carries it.
`condition`	`string`		Conditional text: render this block only when its condition-tag expression is shown under the document's conditions map. Whitespace/comma tag terms, each optionally '!'-negated, AND-combined; an unlisted tag defaults to shown. Empty means always shown.
`continuationIndent`	`number`	0	Points added to the level indent for lines 2..N of a wrapped row, so a continuation reads as part of its entry rather than as a new one. Only meaningful with maxLines. Note that every line of the row is laid out to the narrowed width, the first included, so the wrap point is the same on all of them.
`count`	`integer`		List of Figures (type "lof"): number of figure entries to reserve before the auto-filled captions converge.
`float`	`string`		Float this block out of the stacked flow to one side of its column; content that FOLLOWS it wraps into the space beside it. Works on ANY block type -- a table, a container, a chart, a code listing, a figure. "center" straddles the page content width, so text in every column wraps around it. The float box takes the block's own "width"; a block that declares none takes HALF the column, so a float always leaves room beside it. Ignored on a block that also sets "position" or "anchor" -- those have already chosen their placement. Not to be confused with "anchor": { "float": true }, a BOOLEAN about migrating an anchored frame to the next column that can hold it. One of: `left`, `right`, `center`.
`groupBlend`	`string`		Composite this block AND its children once, then blend the RESULT with the page. NOT the same key as "blend", and the difference is larger than it is for opacity: two stacked children that each multiply the backdrop multiply it TWICE, which is darker than the group multiplying once, and no per-child setting reproduces it. That is what a design tool's blend-mode control on a group means. Container-scoped: on any other flow block it is refused with a warning. Also the way to blend an element together with its drop shadow, which is drawn as stacked alpha layers and so keeps its own normal compositing. See the blend-modes sample. One of: `normal`, `multiply`, `screen`, `overlay`, `darken`, `lighten`, `colorDodge`, `colorBurn`, `hardLight`, `softLight`, `difference`, `exclusion`, `hue`, `saturation`, ...

Key	Type	Default	Description
groupOpacity	number		Composite this block AND its children once, then fade the result -- true GROUP opacity. NOT the same key as "opacity", which fades each element as it paints: two overlapping opaque children at opacity 0.5 each show a seam where they cross, and under groupOpacity 0.5 they do not. The two compose. Container-scoped: on any other flow block it is refused with a warning. See the clipped-groups sample.
leader	string	.	Glyph repeated between an entry and its page number. Empty turns the leader off; a Chinese book may prefer a middle dot · or an em dash.
levelStyles	object		Per heading level, a style delta applied to that level's rows and resolved over the block's own style -- so a contents list can set its top level larger, bolder or coloured, and give it space above, without touching the rest. Keys are heading levels as strings ("1", "2", ...). A level with no entry renders exactly as it does today. This does NOT switch on inheritance from the heading itself: a bold heading still yields a non-bold row unless a level says otherwise. On a flat list (lof / lot / lol / loc) every row is level 1.
maxLevel	integer		Deepest heading level to list. Level 3 lists H1-H3 and ignores anything deeper.
maxLines	integer	1	Lines one generated row may occupy. The default of 1 is the historical behaviour exactly: an over-long entry is truncated with an ellipsis so it cannot run into the page number. Any other value WRAPS the entry instead, on word boundaries, through the same line breaker body text uses (so hyphenation and CJK rules apply); continuation lines carry no leader and no page number, which stay on the last line, and the whole row remains ONE entry -- one TOCI, one Link, one clickable target over every line. A row that still does not fit within maxLines has its final permitted line clamped with the ellipsis as before. 0 means unbounded. Wrapping is opt-in because turning it on changes where existing documents break their pages.
mirrorH	boolean | string		Mirror the whole block left-to-right about the centre of its "position" box, under the same restriction as "angle". Identical to scale x -1, and the two compose. NOT "flipH": that key belongs to the image block, where it mirrors the PICTURE INSIDE its frame. The two are separate words so an image can do both. A STRING is a data token, so each record of a template mirrors its own sticker: "{{s.flip}}" takes the record's boolean, or text "true" / "false" in any case, or 1 / 0 (a number or text); a record without it draws the block unmirrored, silently, and any other value ("yes" included) is reported with the block's pointer (and the -merge record) and drawn unmirrored. A string with no token is read as it stands.
mirrorV	boolean | string		Mirror the whole block top-to-bottom about the centre of its "position" box, under the same restriction as "angle". Identical to scale y -1. NOT "flipV" -- see "mirrorH". A STRING is a data token, so each record of a template mirrors its own sticker: "{{s.flip}}" takes the record's boolean, or text "true" / "false" in any case, or 1 / 0 (a number or text); a record without it draws the block unmirrored, silently, and any other value ("yes" included) is reported with the block's pointer (and the -merge record) and drawn unmirrored. A string with no token is read as it stands.
numbers	boolean		Prefix each entry with its outline number (1, 1.1, 1.1.3) derived from the heading hierarchy.

Key	Type	Default	Description
`opacity`	`number`		Constant opacity for this block: 0 (invisible) to 1 (opaque, the default). REAL alpha -- a /CA /ca ExtGState in the PDF -- so a translucent panel over a photograph shows the photograph through it. Not to be confused with "style.background" {color, alpha}, which composites the tint toward WHITE at parse time and so only looks right over a white page. PER-ELEMENT, NOT GROUP: on a container it paints that container's own decoration -- background, border, accent bar, stripe -- and never its children, which draw at their own opacity (1 unless they set one). Two translucent shapes that overlap therefore composite twice, which is what per-element opacity means; group opacity, where a whole subtree is composited once and then faded, is a separate feature. Where a draw already carries its own alpha (an SVG fill-opacity, a gradient's alpha) the two MULTIPLY.
`repeatAt`	`array of object`		Imposition: draw this block ONCE and place the drawing again at each of these offsets, in points from the block's own position, y DOWNWARD as every `position` is. One entry = one extra copy, so a list of 7 puts 8 cards on a sheet. The drawing is captured into a Form XObject and each offset is a `cm` + `Do` of that one stream, so n copies cost one copy of the content. Translation only -- `angle` / `scale` / `skew` are not applied per placement. A REPEATED FORM MAY NOT CARRY TAGGED CONTENT: ISO 14289-1 7.20 allows a form's content ONE semantic parent, so repeating tagged text (or a shape with `alt`) fails PDF/UA -- veraPDF catches it, ap_addstruct's audit does not. The engine warns and draws it anyway. Repeat decoration: a shape or svg with no `alt`, or put the block in a master's `frames`, which is drawn as a page artifact. Only a block with `position` is drawn again: on a block without one (in the flow, or an anchored frame), or on a repeat / if / include block, the key is refused with a notice naming the block. Each copy takes the block's `groupOpacity` and `groupBlend`. The layout map (-map) reports the copies as `repeats` on the block's fragment, so a design tool can select and outline them.
`scale`	`number | array of number | object | string`		Scale the block about the same pivot as "angle" (the centre of its "position" box), and under the same restriction. A CONTAINER in the flow may also carry this: it becomes a GROUP that turns about its own measured box and reserves the bounding box of the turned result, and its children inherit the transform. Any other flow block ignores it with a warning -- wrap it in a container. A number scales both axes; [x, y] or {"x":, "y":} scales them separately. 1 = natural size. Applied AFTER layout, so a scaled block does NOT re-wrap its text at the scaled measure -- this is a design tool's scale handle, not a change of font size. A STRING is a data token, so each record of a template scales its own sticker: "{{s.scale}}" takes the record's number, [x, y] or {"x":, "y":} (members may be text of numbers), or text of one or two plain numbers such as a CSV field ("1.5", "1.5 0.8", "[1.5, 0.8]"); a record without it draws the block at its natural size, silently, and a value that is not a number or a pair is reported with the block's pointer (and the -merge record) and drawn at natural size. A string with no token is read as it stands.

Key	Type	Default	Description
scope	string	document	How much of the document this block draws on. "document" (the default) lists every entry in the document. "group" lists only entries whose page falls in the same page-numbering group as the block itself - the run of sections since the last restartNumbering - so one volume can hold several complete sub-documents, each with its own contents, figures and index restarting alongside its own page numbers. One of: document, group, section.
skew	number | array of number | object | string		Shear the block, in DEGREES, about the same pivot as "angle", under the same restriction. [x, y] or {"x":, "y":}; a bare number skews x only. Positive x leans the top to the right and positive y lifts the right-hand side -- both counter-clockwise, matching "angle". Clamped to +/-89.9: a 90-degree skew is a singular matrix and draws nothing. A STRING is a data token, so each record of a template skews its own sticker: "{{s.skew}}" takes the record's number (x only), [x, y] or {"x":, "y":}, or text of one or two plain numbers; clamped as the written one is. A record without it draws the block unskewed, silently, and a value that is not a number or a pair is reported with the block's pointer (and the -merge record) and drawn unskewed. A string with no token is read as it stands.
straddle	boolean		Straddle across columns: when true and the body is multi-column, this block spans the FULL content width and text in every column wraps around it (above and below) - a spanning float generalized from center-floated images to any block. No effect in a single-column body, on a positioned / anchored block, or on an image/pdf already using float:"center" (the same mechanism). Does not balance the columns above it or act as a section divider.
title	string		Heading printed above the list. Omit for no heading.
verticalAlign	string		Where the content sits in its "position" box when it is shorter than the box's height: "top" (the default), "middle" ("center" is the same) or "bottom". A paragraph, heading or pre moves as a whole (its own box with it); a container keeps its box where it is and moves its children inside it (a child with its own "position" stays put). The offset is the box height less the content's height as it is actually placed -- the same lines, shrinkToFit size and padding that are then drawn -- halved for middle. Content as tall as the box or taller does not move: it is clipped and reported exactly as without the key. Needs a position with a height: on a block without one, on any other kind of block, on text set along a textPath "d" curve (stated in fractions of the whole box, so drawn across all of it; the "arc" form aligns), and on vertical text whose columns fill the box (columns run the full height unless a paragraph's or a heading's boxHeight is shorter), it is refused with a notice and ignored. The layout map keeps reporting the box. One of: top, middle, bottom, center.
z	number		CANVAS MODE ONLY (see the top-level "pages"): where this element sits in the PAINT order of its page. Lower draws first, further back; unset means "wherever I am in the list", and ties keep author order. It moves the PAINT and not the READING: the structure tree keeps the document order, so a caption written first because it is read first can still be painted last because it sits on top. That separation is what a flattened scene graph cannot express. Ignored outside canvas mode, where the flow already fixes both orders.

The toc element

Type name: `toc`. Accepted aliases: `lof`, `lot`, `lol`, `loc`.
An automatic list of figures, tables or listings - `lof`, `lot` and `lol` are the same element filtered to a different kind of caption. Each row is a linked "caption ..... page" line, built the same converging way as the table of contents.

Example

```
{
  "type": "lof",
  "title": "List of Figures"
}
```

A list of tables (the lot alias)

```
{
  "type": "lot",
  "title": "List of Tables"
}
```

Keys

Key	Type	Default	Description
`anchor`	`anchor`		-
`angle`	`number | string`		Turn the block, in degrees counter-clockwise, about the CENTRE of its "position" box. A CONTAINER in the flow may also carry this: it becomes a GROUP that turns about its own measured box and reserves the bounding box of the turned result, and its children inherit the transform. Any other flow block ignores it with a warning -- wrap it in a container. The block lays out upright inside its box and the finished result is turned, so a "position.height" that truncates content truncates the same text it would unrotated, and the turned result is NOT clipped to its box: it may extend past it, exactly as an unrotated positioned block may. Not to be confused with the image block's "rotate", which is an integer QUARTER TURN CLOCKWISE applied to the picture INSIDE its box (and re-sizes that box to the turned orientation). The two compose -- an image can stand a sideways scan upright with "rotate" and tilt the whole frame with "angle". Same meaning as the watermark's "angle". A STRING is a data token, so each record of a template turns its own sticker: "{{s.angle}}" takes the record's number (or text of one, such as a CSV field); a record without it leaves the block upright, silently, and a value that is not a number is reported with the block's pointer (and the -merge record) and drawn upright. A string with no token is read as it stands.

Key	Type	Default	Description
bleed	number		Step and Repeat: how far this placed block's box reaches past its TRIM on every side, in points -- the box IS the bleed box, and the trim is the box inset by this much. A design tool sizes a card by it (the trim plus twice the bleed) and draws the trim line inside each copy; the engine draws nothing from it, and keeps the bleed only through the block's own clip. Only a block with position has a trim: on any other block the key is refused with a notice naming the block.
blend	string		How this block's OWN ink combines with the page under it -- the sixteen PDF blend modes (ISO 32000-1 11.3.5), written as /BM. On a container that means its background, border and stripe, never its children; for the subtree as one object see "groupBlend". Case and separators are folded, so "colorDodge", "color-dodge" and "COLORDODGE" are one mode. An unrecognized name is refused with a warning and the block paints normally. See the blend-modes sample. One of: normal, multiply, screen, overlay, darken, lighten, colorDodge, colorBurn, hardLight, softLight, difference, exclusion, hue, saturation, ...
breakBefore	boolean		Pagination: open a new page before this block - a part divider, an appendix. Ignored when the page is already empty, so it never leaves a blank page behind.
changed	boolean		Revision / change bar: draw a vertical bar in the page margin beside this block's content to flag it as changed. A document-level changeBars object customizes the bar (width / color / side / distance); a condition tag declared with changeBar:true sets this automatically for tagged content.
clip	boolean		Clip this block's children (on a container) or its own ink to its box, with the style's corner "radius". Only bites where content is BIGGER than the box: a flow container grows to fit its children vertically, so give it "position" with a height, or a "minHeight", before expecting a vertical crop. Container-scoped, like "groupOpacity": on any other flow block it is refused with a warning. For a crop to a SHAPE rather than to the box, see "clipPath"; giving both INTERSECTS them.
clipPath	object		Crop this block to an arbitrary SHAPE stated inside its box -- a design tool's FRAME. Same vocabulary as the "shape" block and read by the same outline builder, so a shape and a clipPath given identical parameters describe the identical outline. Scoped exactly as "clip" is: a positioned block of any type (it needs a "position.height", because the shape is stated inside the box) or a container in the flow; anywhere else it is refused with a warning. Naming "clip" as well INTERSECTS the two. The shape turns, scales and skews with the block that carries it.
condition	string		Conditional text: render this block only when its condition-tag expression is shown under the document's conditions map. Whitespace/comma tag terms, each optionally '!'-negated, AND-combined; an unlisted tag defaults to shown. Empty means always shown.

Key	Type	Default	Description
`continuationIndent`	`number`	0	Points added to the level indent for lines 2..N of a wrapped row, so a continuation reads as part of its entry rather than as a new one. Only meaningful with maxLines. Note that every line of the row is laid out to the narrowed width, the first included, so the wrap point is the same on all of them.
`count`	`integer`		List of Figures (type "lof"): number of figure entries to reserve before the auto-filled captions converge.
`float`	`string`		Float this block out of the stacked flow to one side of its column; content that FOLLOWS it wraps into the space beside it. Works on ANY block type -- a table, a container, a chart, a code listing, a figure. "center" straddles the page content width, so text in every column wraps around it. The float box takes the block's own "width"; a block that declares none takes HALF the column, so a float always leaves room beside it. Ignored on a block that also sets "position" or "anchor" -- those have already chosen their placement. Not to be confused with "anchor": { "float": true }, a BOOLEAN about migrating an anchored frame to the next column that can hold it. One of: `left`, `right`, `center`.
`groupBlend`	`string`		Composite this block AND its children once, then blend the RESULT with the page. NOT the same key as "blend", and the difference is larger than it is for opacity: two stacked children that each multiply the backdrop multiply it TWICE, which is darker than the group multiplying once, and no per-child setting reproduces it. That is what a design tool's blend-mode control on a group means. Container-scoped: on any other flow block it is refused with a warning. Also the way to blend an element together with its drop shadow, which is drawn as stacked alpha layers and so keeps its own normal compositing. See the blend-modes sample. One of: `normal`, `multiply`, `screen`, `overlay`, `darken`, `lighten`, `colorDodge`, `colorBurn`, `hardLight`, `softLight`, `difference`, `exclusion`, `hue`, `saturation`, ...
`groupOpacity`	`number`		Composite this block AND its children once, then fade the result -- true GROUP opacity. NOT the same key as "opacity", which fades each element as it paints: two overlapping opaque children at opacity 0.5 each show a seam where they cross, and under groupOpacity 0.5 they do not. The two compose. Container-scoped: on any other flow block it is refused with a warning. See the clipped-groups sample.
`leader`	`string`	.	Glyph repeated between an entry and its page number. Empty turns the leader off; a Chinese book may prefer a middle dot · or an em dash.
`levelStyles`	`object`		Per heading level, a style delta applied to that level's rows and resolved over the block's own style -- so a contents list can set its top level larger, bolder or coloured, and give it space above, without touching the rest. Keys are heading levels as strings ("1", "2", ...). A level with no entry renders exactly as it does today. This does NOT switch on inheritance from the heading itself: a bold heading still yields a non-bold row unless a level says otherwise. On a flat list (lof / lot / lol / loc) every row is level 1.
`maxLevel`	`integer`		Deepest heading level to list. Level 3 lists H1-H3 and ignores anything deeper.

Key	Type	Default	Description
maxLines	integer	1	Lines one generated row may occupy. The default of 1 is the historical behaviour exactly: an over-long entry is truncated with an ellipsis so it cannot run into the page number. Any other value WRAPS the entry instead, on word boundaries, through the same line breaker body text uses (so hyphenation and CJK rules apply); continuation lines carry no leader and no page number, which stay on the last line, and the whole row remains ONE entry -- one TOCI, one Link, one clickable target over every line. A row that still does not fit within maxLines has its final permitted line clamped with the ellipsis as before. 0 means unbounded. Wrapping is opt-in because turning it on changes where existing documents break their pages.
mirrorH	boolean | string		Mirror the whole block left-to-right about the centre of its "position" box, under the same restriction as "angle". Identical to scale x -1, and the two compose. NOT "flipH": that key belongs to the image block, where it mirrors the PICTURE INSIDE its frame. The two are separate words so an image can do both. A STRING is a data token, so each record of a template mirrors its own sticker: "{{s.flip}}" takes the record's boolean, or text "true" / "false" in any case, or 1 / 0 (a number or text); a record without it draws the block unmirrored, silently, and any other value ("yes" included) is reported with the block's pointer (and the -merge record) and drawn unmirrored. A string with no token is read as it stands.
mirrorV	boolean | string		Mirror the whole block top-to-bottom about the centre of its "position" box, under the same restriction as "angle". Identical to scale y -1. NOT "flipV" -- see "mirrorH". A STRING is a data token, so each record of a template mirrors its own sticker: "{{s.flip}}" takes the record's boolean, or text "true" / "false" in any case, or 1 / 0 (a number or text); a record without it draws the block unmirrored, silently, and any other value ("yes" included) is reported with the block's pointer (and the -merge record) and drawn unmirrored. A string with no token is read as it stands.
numbers	boolean		Prefix each entry with its outline number (1, 1.1, 1.1.3) derived from the heading hierarchy.
opacity	number		Constant opacity for this block: 0 (invisible) to 1 (opaque, the default). REAL alpha -- a /CA /ca ExtGState in the PDF -- so a translucent panel over a photograph shows the photograph through it. Not to be confused with "style.background" {color, alpha}, which composites the tint toward WHITE at parse time and so only looks right over a white page. PER-ELEMENT, NOT GROUP: on a container it paints that container's own decoration -- background, border, accent bar, stripe -- and never its children, which draw at their own opacity (1 unless they set one). Two translucent shapes that overlap therefore composite twice, which is what per-element opacity means; group opacity, where a whole subtree is composited once and then faded, is a separate feature. Where a draw already carries its own alpha (an SVG fill-opacity, a gradient's alpha) the two MULTIPLY.

Key	Type	Default	Description
repeatAt	array of object		Imposition: draw this block ONCE and place the drawing again at each of these offsets, in points from the block's own position, y DOWNWARD as every position is. One entry = one extra copy, so a list of 7 puts 8 cards on a sheet. The drawing is captured into a Form XObject and each offset is a cm + Do of that one stream, so n copies cost one copy of the content. Translation only -- angle / scale / skew are not applied per placement. A REPEATED FORM MAY NOT CARRY TAGGED CONTENT: ISO 14289-1 7.20 allows a form's content ONE semantic parent, so repeating tagged text (or a shape with alt) fails PDF/UA -- veraPDF catches it, ap_addstruct's audit does not. The engine warns and draws it anyway. Repeat decoration: a shape or svg with no alt, or put the block in a master's frames, which is drawn as a page artifact. Only a block with position is drawn again: on a block without one (in the flow, or an anchored frame), or on a repeat / if / include block, the key is refused with a notice naming the block. Each copy takes the block's groupOpacity and groupBlend. The layout map (-map) reports the copies as repeats on the block's fragment, so a design tool can select and outline them.
scale	number | array of number | object | string		Scale the block about the same pivot as "angle" (the centre of its "position" box), and under the same restriction. A CONTAINER in the flow may also carry this: it becomes a GROUP that turns about its own measured box and reserves the bounding box of the turned result, and its children inherit the transform. Any other flow block ignores it with a warning -- wrap it in a container. A number scales both axes; [x, y] or {"x":, "y":} scales them separately. 1 = natural size. Applied AFTER layout, so a scaled block does NOT re-wrap its text at the scaled measure -- this is a design tool's scale handle, not a change of font size. A STRING is a data token, so each record of a template scales its own sticker: "{{s.scale}}" takes the record's number, [x, y] or {"x":, "y":} (members may be text of numbers), or text of one or two plain numbers such as a CSV field ("1.5", "1.5 0.8", "[1.5, 0.8]"); a record without it draws the block at its natural size, silently, and a value that is not a number or a pair is reported with the block's pointer (and the -merge record) and drawn at natural size. A string with no token is read as it stands.
scope	string	document	How much of the document this block draws on. "document" (the default) lists every entry in the document. "group" lists only entries whose page falls in the same page-numbering group as the block itself - the run of sections since the last restartNumbering - so one volume can hold several complete sub-documents, each with its own contents, figures and index restarting alongside its own page numbers. One of: document, group, section.

Key	Type	Default	Description
skew	number | array of number | object | string		Shear the block, in DEGREES, about the same pivot as "angle", under the same restriction. [x, y] or {"x":, "y":}; a bare number skews x only. Positive x leans the top to the right and positive y lifts the right-hand side -- both counter-clockwise, matching "angle". Clamped to +/-89.9: a 90-degree skew is a singular matrix and draws nothing. A STRING is a data token, so each record of a template skews its own sticker: "{{s.skew}}" takes the record's number (x only), [x, y] or {"x":, "y":}, or text of one or two plain numbers; clamped as the written one is. A record without it draws the block unskewed, silently, and a value that is not a number or a pair is reported with the block's pointer (and the -merge record) and drawn unskewed. A string with no token is read as it stands.
straddle	boolean		Straddle across columns: when true and the body is multi-column, this block spans the FULL content width and text in every column wraps around it (above and below) - a spanning float generalized from center-floated images to any block. No effect in a single-column body, on a positioned / anchored block, or on an image/pdf already using float:"center" (the same mechanism). Does not balance the columns above it or act as a section divider.
title	string		Heading printed above the list. Omit for no heading.
verticalAlign	string		Where the content sits in its "position" box when it is shorter than the box's height: "top" (the default), "middle" ("center" is the same) or "bottom". A paragraph, heading or pre moves as a whole (its own box with it); a container keeps its box where it is and moves its children inside it (a child with its own "position" stays put). The offset is the box height less the content's height as it is actually placed -- the same lines, shrinkToFit size and padding that are then drawn -- halved for middle. Content as tall as the box or taller does not move: it is clipped and reported exactly as without the key. Needs a position with a height: on a block without one, on any other kind of block, on text set along a textPath "d" curve (stated in fractions of the whole box, so drawn across all of it; the "arc" form aligns), and on vertical text whose columns fill the box (columns run the full height unless a paragraph's or a heading's boxHeight is shorter), it is refused with a notice and ignored. The layout map keeps reporting the box. One of: top, middle, bottom, center.
z	number		CANVAS MODE ONLY (see the top-level "pages"): where this element sits in the PAINT order of its page. Lower draws first, further back; unset means "wherever I am in the list", and ties keep author order. It moves the PAINT and not the READING: the structure tree keeps the document order, so a caption written first because it is read first can still be painted last because it sits on top. That separation is what a flattened scene graph cannot express. Ignored outside canvas mode, where the flow already fixes both orders.

The index element

Type name: index.

A back-of-book index. You supply the terms; the engine scans the drawn text as it lays the document out, records every page a term actually appears on, collapses runs of consecutive pages into ranges, and renders the result in columns with letter headings - every page number a working link. Chinese indexes sort the way their readers expect:

`collation` chooses pinyin (Simplified) or stroke-count-then-radical (Traditional) order from an embedded Unihan-derived table, and a per-entry `sortAs` overrides the reading where dictionaries disagree.

Example

```
{
  "type": "index",
  "columns": 2,
  "letterHeadings": true,
  "collapseRanges": true,
  "terms": [
    "stability",
    "recovery",
    "motor"
  ]
}
```

Keys

Key	Type	Default	Description
`anchor`	`anchor`		-
`angle`	`number | string`		Turn the block, in degrees counter-clockwise, about the CENTRE of its "position" box. A CONTAINER in the flow may also carry this: it becomes a GROUP that turns about its own measured box and reserves the bounding box of the turned result, and its children inherit the transform. Any other flow block ignores it with a warning -- wrap it in a container. The block lays out upright inside its box and the finished result is turned, so a "position.height" that truncates content truncates the same text it would unrotated, and the turned result is NOT clipped to its box: it may extend past it, exactly as an unrotated positioned block may. Not to be confused with the image block's "rotate", which is an integer QUARTER TURN CLOCKWISE applied to the picture INSIDE its box (and re-sizes that box to the turned orientation). The two compose -- an image can stand a sideways scan upright with "rotate" and tilt the whole frame with "angle". Same meaning as the watermark's "angle". A STRING is a data token, so each record of a template turns its own sticker: "{{s.angle}}" takes the record's number (or text of one, such as a CSV field); a record without it leaves the block upright, silently, and a value that is not a number is reported with the block's pointer (and the -merge record) and drawn upright. A string with no token is read as it stands.
`bleed`	`number`		Step and Repeat: how far this placed block's box reaches past its TRIM on every side, in points -- the box IS the bleed box, and the trim is the box inset by this much. A design tool sizes a card by it (the trim plus twice the bleed) and draws the trim line inside each copy; the engine draws nothing from it, and keeps the bleed only through the block's own `clip`. Only a block with `position` has a trim: on any other block the key is refused with a notice naming the block.

Key	Type	Default	Description
blend	string		How this block's OWN ink combines with the page under it -- the sixteen PDF blend modes (ISO 32000-1 11.3.5), written as /BM. On a container that means its background, border and stripe, never its children; for the subtree as one object see "groupBlend". Case and separators are folded, so "colorDodge", "color-dodge" and "COLORDODGE" are one mode. An unrecognized name is refused with a warning and the block paints normally. See the blend-modes sample. One of: normal, multiply, screen, overlay, darken, lighten, colorDodge, colorBurn, hardLight, softLight, difference, exclusion, hue, saturation, ...
breakBefore	boolean		Pagination: open a new page before this block - a part divider, an appendix. Ignored when the page is already empty, so it never leaves a blank page behind.
changed	boolean		Revision / change bar: draw a vertical bar in the page margin beside this block's content to flag it as changed. A document-level changeBars object customizes the bar (width / color / side / distance); a condition tag declared with changeBar:true sets this automatically for tagged content.
clip	boolean		Clip this block's children (on a container) or its own ink to its box, with the style's corner "radius". Only bites where content is BIGGER than the box: a flow container grows to fit its children vertically, so give it "position" with a height, or a "minHeight", before expecting a vertical crop. Container-scoped, like "groupOpacity": on any other flow block it is refused with a warning. For a crop to a SHAPE rather than to the box, see "clipPath"; giving both INTERSECTS them.
clipPath	object		Crop this block to an arbitrary SHAPE stated inside its box -- a design tool's FRAME. Same vocabulary as the "shape" block and read by the same outline builder, so a shape and a clipPath given identical parameters describe the identical outline. Scoped exactly as "clip" is: a positioned block of any type (it needs a "position.height", because the shape is stated inside the box) or a container in the flow; anywhere else it is refused with a warning. Naming "clip" as well INTERSECTS the two. The shape turns, scales and skews with the block that carries it.
collapseRanges	boolean		Collapse runs of 3+ consecutive pages to a range (5-7).
collation	string	unicode	How the entries are ordered. "unicode" (the default) is code-point order with ASCII case folded - fine for Latin, meaningless for Chinese, where it is Unicode block order. "pinyin" orders by the Hanyu Pinyin reading then the tone, so 北京 files under B; it is the dominant order for Simplified Chinese, and Latin entries interleave alphabetically. "strokes" orders by total stroke count, then Kangxi radical, then code point - the Traditional and reference-work order - and gathers Latin entries into their own run first. Readings and stroke counts come from the Unicode Unihan database, embedded for CJK Unified Ideographs and Extension A; anything outside that (Extension B and beyond) falls back to code-point order, as does any character Unihan gives no reading. One collation per document: like collapseRanges and letterHeadings, the last index block in the document sets it. One of: unicode, pinyin, strokes.

Key	Type	Default	Description
`columns`	`integer`		How many columns the index entries are set in.
`condition`	`string`		Conditional text: render this block only when its condition-tag expression is shown under the document's conditions map. Whitespace/comma tag terms, each optionally '!'-negated, AND-combined; an unlisted tag defaults to shown. Empty means always shown.
`float`	`string`		Float this block out of the stacked flow to one side of its column; content that FOLLOWS it wraps into the space beside it. Works on ANY block type -- a table, a container, a chart, a code listing, a figure. "center" straddles the page content width, so text in every column wraps around it. The float box takes the block's own "width"; a block that declares none takes HALF the column, so a float always leaves room beside it. Ignored on a block that also sets "position" or "anchor" -- those have already chosen their placement. Not to be confused with "anchor": { "float": true }, a BOOLEAN about migrating an anchored frame to the next column that can hold it. One of: `left`, `right`, `center`.
`groupBlend`	`string`		Composite this block AND its children once, then blend the RESULT with the page. NOT the same key as "blend", and the difference is larger than it is for opacity: two stacked children that each multiply the backdrop multiply it TWICE, which is darker than the group multiplying once, and no per-child setting reproduces it. That is what a design tool's blend-mode control on a group means. Container-scoped: on any other flow block it is refused with a warning. Also the way to blend an element together with its drop shadow, which is drawn as stacked alpha layers and so keeps its own normal compositing. See the blend-modes sample. One of: `normal`, `multiply`, `screen`, `overlay`, `darken`, `lighten`, `colorDodge`, `colorBurn`, `hardLight`, `softLight`, `difference`, `exclusion`, `hue`, `saturation`, ...
`groupOpacity`	`number`		Composite this block AND its children once, then fade the result -- true GROUP opacity. NOT the same key as "opacity", which fades each element as it paints: two overlapping opaque children at opacity 0.5 each show a seam where they cross, and under groupOpacity 0.5 they do not. The two compose. Container-scoped: on any other flow block it is refused with a warning. See the clipped-groups sample.
`letterHeadings`	`boolean`		Print an A / B / C letter heading before each alphabetical group.
`mirrorH`	`boolean | string`		Mirror the whole block left-to-right about the centre of its "position" box, under the same restriction as "angle". Identical to scale x -1, and the two compose. NOT "flipH": that key belongs to the image block, where it mirrors the PICTURE INSIDE its frame. The two are separate words so an image can do both. A STRING is a data token, so each record of a template mirrors its own sticker: "{{s.flip}}" takes the record's boolean, or text "true" / "false" in any case, or 1 / 0 (a number or text); a record without it draws the block unmirrored, silently, and any other value ("yes" included) is reported with the block's pointer (and the -merge record) and drawn unmirrored. A string with no token is read as it stands.

Key	Type	Default	Description
`mirrorV`	`boolean | string`		Mirror the whole block top-to-bottom about the centre of its "position" box, under the same restriction as "angle". Identical to scale y -1. NOT "flipV" -- see "mirrorH". A STRING is a data token, so each record of a template mirrors its own sticker: "{{s.flip}}" takes the record's boolean, or text "true" / "false" in any case, or 1 / 0 (a number or text); a record without it draws the block unmirrored, silently, and any other value ("yes" included) is reported with the block's pointer (and the -merge record) and drawn unmirrored. A string with no token is read as it stands.
`opacity`	`number`		Constant opacity for this block: 0 (invisible) to 1 (opaque, the default). REAL alpha -- a /CA /ca ExtGState in the PDF -- so a translucent panel over a photograph shows the photograph through it. Not to be confused with "style.background" {color, alpha}, which composites the tint toward WHITE at parse time and so only looks right over a white page. PER-ELEMENT, NOT GROUP: on a container it paints that container's own decoration -- background, border, accent bar, stripe -- and never its children, which draw at their own opacity (1 unless they set one). Two translucent shapes that overlap therefore composite twice, which is what per-element opacity means; group opacity, where a whole subtree is composited once and then faded, is a separate feature. Where a draw already carries its own alpha (an SVG fill-opacity, a gradient's alpha) the two MULTIPLY.
`repeatAt`	`array of object`		Imposition: draw this block ONCE and place the drawing again at each of these offsets, in points from the block's own position, y DOWNWARD as every `position` is. One entry = one extra copy, so a list of 7 puts 8 cards on a sheet. The drawing is captured into a Form XObject and each offset is a `cm` + `Do` of that one stream, so n copies cost one copy of the content. Translation only -- `angle` / `scale` / `skew` are not applied per placement. A REPEATED FORM MAY NOT CARRY TAGGED CONTENT: ISO 14289-1 7.20 allows a form's content ONE semantic parent, so repeating tagged text (or a shape with `alt`) fails PDF/UA -- veraPDF catches it, ap_addstruct's audit does not. The engine warns and draws it anyway. Repeat decoration: a shape or svg with no `alt`, or put the block in a master's `frames`, which is drawn as a page artifact. Only a block with `position` is drawn again: on a block without one (in the flow, or an anchored frame), or on a repeat / if / include block, the key is refused with a notice naming the block. Each copy takes the block's `groupOpacity` and `groupBlend`. The layout map (-map) reports the copies as `repeats` on the block's fragment, so a design tool can select and outline them.

Key	Type	Default	Description
scale	number | array of number | object | string		Scale the block about the same pivot as "angle" (the centre of its "position" box), and under the same restriction. A CONTAINER in the flow may also carry this: it becomes a GROUP that turns about its own measured box and reserves the bounding box of the turned result, and its children inherit the transform. Any other flow block ignores it with a warning -- wrap it in a container. A number scales both axes; [x, y] or {"x":, "y":} scales them separately. 1 = natural size. Applied AFTER layout, so a scaled block does NOT re-wrap its text at the scaled measure -- this is a design tool's scale handle, not a change of font size. A STRING is a data token, so each record of a template scales its own sticker: "{{s.scale}}" takes the record's number, [x, y] or {"x":, "y":} (members may be text of numbers), or text of one or two plain numbers such as a CSV field ("1.5", "1.5 0.8", "[1.5, 0.8]"); a record without it draws the block at its natural size, silently, and a value that is not a number or a pair is reported with the block's pointer (and the -merge record) and drawn at natural size. A string with no token is read as it stands.
scope	string	document	How much of the document this block draws on. "document" (the default) lists every entry in the document. "group" lists only entries whose page falls in the same page-numbering group as the block itself - the run of sections since the last restartNumbering - so one volume can hold several complete sub-documents, each with its own contents, figures and index restarting alongside its own page numbers. One of: document, group, section.
skew	number | array of number | object | string		Shear the block, in DEGREES, about the same pivot as "angle", under the same restriction. [x, y] or {"x":, "y":}; a bare number skews x only. Positive x leans the top to the right and positive y lifts the right-hand side -- both counter-clockwise, matching "angle". Clamped to +/-89.9: a 90-degree skew is a singular matrix and draws nothing. A STRING is a data token, so each record of a template skews its own sticker: "{{s.skew}}" takes the record's number (x only), [x, y] or {"x":, "y":}, or text of one or two plain numbers; clamped as the written one is. A record without it draws the block unskewed, silently, and a value that is not a number or a pair is reported with the block's pointer (and the -merge record) and drawn unskewed. A string with no token is read as it stands.
straddle	boolean		Straddle across columns: when true and the body is multi-column, this block spans the FULL content width and text in every column wraps around it (above and below) - a spanning float generalized from center-floated images to any block. No effect in a single-column body, on a positioned / anchored block, or on an image/pdf already using float:"center" (the same mechanism). Does not balance the columns above it or act as a section divider.
strokeLabel	string		Drawn after a stroke-count group divider: "" (the default) gives a bare "12", "画" gives 12画 and "畫" the Traditional form. No effect unless collation is "strokes".
terms	array of indexTerm		Terms to index. Each collects the pages where it was found in the text.
title	string		Heading printed above the index.

Key	Type	Default	Description
`verticalAlign`	`string`		Where the content sits in its "position" box when it is shorter than the box's height: "top" (the default), "middle" ("center" is the same) or "bottom". A paragraph, heading or pre moves as a whole (its own box with it); a container keeps its box where it is and moves its children inside it (a child with its own "position" stays put). The offset is the box height less the content's height as it is actually placed -- the same lines, shrinkToFit size and padding that are then drawn -- halved for middle. Content as tall as the box or taller does not move: it is clipped and reported exactly as without the key. Needs a position with a height: on a block without one, on any other kind of block, on text set along a textPath "d" curve (stated in fractions of the whole box, so drawn across all of it; the "arc" form aligns), and on vertical text whose columns fill the box (columns run the full height unless a paragraph's or a heading's boxHeight is shorter), it is refused with a notice and ignored. The layout map keeps reporting the box. One of: `top`, `middle`, `bottom`, `center`.
`words`	`array of indexTerm`		Alias for "terms".
`z`	`number`		CANVAS MODE ONLY (see the top-level "pages"): where this element sits in the PAINT order of its page. Lower draws first, further back; unset means "wherever I am in the list", and ties keep author order. It moves the PAINT and not the READING: the structure tree keeps the document order, so a caption written first because it is read first can still be painted last because it sits on top. That separation is what a flattened scene graph cannot express. Ignored outside canvas mode, where the flow already fixes both orders.

The glossary element

Type name: `glossary`.

A list of defined terms. With `autoLink` on, the first appearance of each term in the body becomes a link to its definition.

Example

```
{
  "type": "glossary",
  "autoLink": true,
  "entries": [
    {
      "term": "apogee",
      "definition": "The highest point of the flight."
    }
  ]
}
```

Keys

Key	Type	Default	Description
anchor	anchor		-
angle	number | string		Turn the block, in degrees counter-clockwise, about the CENTRE of its "position" box. A CONTAINER in the flow may also carry this: it becomes a GROUP that turns about its own measured box and reserves the bounding box of the turned result, and its children inherit the transform. Any other flow block ignores it with a warning -- wrap it in a container. The block lays out upright inside its box and the finished result is turned, so a "position.height" that truncates content truncates the same text it would unrotated, and the turned result is NOT clipped to its box: it may extend past it, exactly as an unrotated positioned block may. Not to be confused with the image block's "rotate", which is an integer QUARTER TURN CLOCKWISE applied to the picture INSIDE its box (and re-sizes that box to the turned orientation). The two compose -- an image can stand a sideways scan upright with "rotate" and tilt the whole frame with "angle". Same meaning as the watermark's "angle". A STRING is a data token, so each record of a template turns its own sticker: "{{s.angle}}" takes the record's number (or text of one, such as a CSV field); a record without it leaves the block upright, silently, and a value that is not a number is reported with the block's pointer (and the -merge record) and drawn upright. A string with no token is read as it stands.
autoLink	boolean		Link the first body use of each term to its definition.
bleed	number		Step and Repeat: how far this placed block's box reaches past its TRIM on every side, in points -- the box IS the bleed box, and the trim is the box inset by this much. A design tool sizes a card by it (the trim plus twice the bleed) and draws the trim line inside each copy; the engine draws nothing from it, and keeps the bleed only through the block's own clip. Only a block with position has a trim: on any other block the key is refused with a notice naming the block.
blend	string		How this block's OWN ink combines with the page under it -- the sixteen PDF blend modes (ISO 32000-1 11.3.5), written as /BM. On a container that means its background, border and stripe, never its children; for the subtree as one object see "groupBlend". Case and separators are folded, so "colorDodge", "color-dodge" and "COLORDODGE" are one mode. An unrecognized name is refused with a warning and the block paints normally. See the blend-modes sample. One of: normal, multiply, screen, overlay, darken, lighten, colorDodge, colorBurn, hardLight, softLight, difference, exclusion, hue, saturation, ...
breakBefore	boolean		Pagination: open a new page before this block - a part divider, an appendix. Ignored when the page is already empty, so it never leaves a blank page behind.
changed	boolean		Revision / change bar: draw a vertical bar in the page margin beside this block's content to flag it as changed. A document-level changeBars object customizes the bar (width / color / side / distance); a condition tag declared with changeBar:true sets this automatically for tagged content.

Key	Type	Default	Description
`clip`	`boolean`		Clip this block's children (on a container) or its own ink to its box, with the style's corner "radius". Only bites where content is BIGGER than the box: a flow container grows to fit its children vertically, so give it "position" with a height, or a "minHeight", before expecting a vertical crop. Container-scoped, like "groupOpacity": on any other flow block it is refused with a warning. For a crop to a SHAPE rather than to the box, see "clipPath"; giving both INTERSECTS them.
`clipPath`	`object`		Crop this block to an arbitrary SHAPE stated inside its box -- a design tool's FRAME. Same vocabulary as the "shape" block and read by the same outline builder, so a shape and a clipPath given identical parameters describe the identical outline. Scoped exactly as "clip" is: a positioned block of any type (it needs a "position.height", because the shape is stated inside the box) or a container in the flow; anywhere else it is refused with a warning. Naming "clip" as well INTERSECTS the two. The shape turns, scales and skews with the block that carries it.
`condition`	`string`		Conditional text: render this block only when its condition-tag expression is shown under the document's conditions map. Whitespace/comma tag terms, each optionally '!'-negated, AND-combined; an unlisted tag defaults to shown. Empty means always shown.
`entries`	`array of object`		The glossary entries - each a term and its definition.
`float`	`string`		Float this block out of the stacked flow to one side of its column; content that FOLLOWS it wraps into the space beside it. Works on ANY block type -- a table, a container, a chart, a code listing, a figure. "center" straddles the page content width, so text in every column wraps around it. The float box takes the block's own "width"; a block that declares none takes HALF the column, so a float always leaves room beside it. Ignored on a block that also sets "position" or "anchor" -- those have already chosen their placement. Not to be confused with "anchor": { "float": true }, a BOOLEAN about migrating an anchored frame to the next column that can hold it. One of: `left`, `right`, `center`.
`groupBlend`	`string`		Composite this block AND its children once, then blend the RESULT with the page. NOT the same key as "blend", and the difference is larger than it is for opacity: two stacked children that each multiply the backdrop multiply it TWICE, which is darker than the group multiplying once, and no per-child setting reproduces it. That is what a design tool's blend-mode control on a group means. Container-scoped: on any other flow block it is refused with a warning. Also the way to blend an element together with its drop shadow, which is drawn as stacked alpha layers and so keeps its own normal compositing. See the blend-modes sample. One of: `normal`, `multiply`, `screen`, `overlay`, `darken`, `lighten`, `colorDodge`, `colorBurn`, `hardLight`, `softLight`, `difference`, `exclusion`, `hue`, `saturation`, ...

Key	Type	Default	Description
groupOpacity	number		Composite this block AND its children once, then fade the result -- true GROUP opacity. NOT the same key as "opacity", which fades each element as it paints: two overlapping opaque children at opacity 0.5 each show a seam where they cross, and under groupOpacity 0.5 they do not. The two compose. Container-scoped: on any other flow block it is refused with a warning. See the clipped-groups sample.
mirrorH	boolean | string		Mirror the whole block left-to-right about the centre of its "position" box, under the same restriction as "angle". Identical to scale x -1, and the two compose. NOT "flipH": that key belongs to the image block, where it mirrors the PICTURE INSIDE its frame. The two are separate words so an image can do both. A STRING is a data token, so each record of a template mirrors its own sticker: "{{s.flip}}" takes the record's boolean, or text "true" / "false" in any case, or 1 / 0 (a number or text); a record without it draws the block unmirrored, silently, and any other value ("yes" included) is reported with the block's pointer (and the -merge record) and drawn unmirrored. A string with no token is read as it stands.
mirrorV	boolean | string		Mirror the whole block top-to-bottom about the centre of its "position" box, under the same restriction as "angle". Identical to scale y -1. NOT "flipV" -- see "mirrorH". A STRING is a data token, so each record of a template mirrors its own sticker: "{{s.flip}}" takes the record's boolean, or text "true" / "false" in any case, or 1 / 0 (a number or text); a record without it draws the block unmirrored, silently, and any other value ("yes" included) is reported with the block's pointer (and the -merge record) and drawn unmirrored. A string with no token is read as it stands.
opacity	number		Constant opacity for this block: 0 (invisible) to 1 (opaque, the default). REAL alpha -- a /CA /ca ExtGState in the PDF -- so a translucent panel over a photograph shows the photograph through it. Not to be confused with "style.background" {color, alpha}, which composites the tint toward WHITE at parse time and so only looks right over a white page. PER-ELEMENT, NOT GROUP: on a container it paints that container's own decoration -- background, border, accent bar, stripe -- and never its children, which draw at their own opacity (1 unless they set one). Two translucent shapes that overlap therefore composite twice, which is what per-element opacity means; group opacity, where a whole subtree is composited once and then faded, is a separate feature. Where a draw already carries its own alpha (an SVG fill-opacity, a gradient's alpha) the two MULTIPLY.

Key	Type	Default	Description
`repeatAt`	`array of object`		Imposition: draw this block ONCE and place the drawing again at each of these offsets, in points from the block's own position, y DOWNWARD as every `position` is. One entry = one extra copy, so a list of 7 puts 8 cards on a sheet. The drawing is captured into a Form XObject and each offset is a `cm` + `Do` of that one stream, so n copies cost one copy of the content. Translation only -- `angle` / `scale` / `skew` are not applied per placement. A REPEATED FORM MAY NOT CARRY TAGGED CONTENT: ISO 14289-1 7.20 allows a form's content ONE semantic parent, so repeating tagged text (or a shape with `alt`) fails PDF/UA -- veraPDF catches it, ap_addstruct's audit does not. The engine warns and draws it anyway. Repeat decoration: a shape or svg with no `alt`, or put the block in a master's `frames`, which is drawn as a page artifact. Only a block with `position` is drawn again: on a block without one (in the flow, or an anchored frame), or on a repeat / if / include block, the key is refused with a notice naming the block. Each copy takes the block's `groupOpacity` and `groupBlend`. The layout map (-map) reports the copies as `repeats` on the block's fragment, so a design tool can select and outline them.
`scale`	`number | array of number | object | string`		Scale the block about the same pivot as "angle" (the centre of its "position" box), and under the same restriction. A CONTAINER in the flow may also carry this: it becomes a GROUP that turns about its own measured box and reserves the bounding box of the turned result, and its children inherit the transform. Any other flow block ignores it with a warning -- wrap it in a container. A number scales both axes; [x, y] or {"x":, "y":} scales them separately. 1 = natural size. Applied AFTER layout, so a scaled block does NOT re-wrap its text at the scaled measure -- this is a design tool's scale handle, not a change of font size. A STRING is a data token, so each record of a template scales its own sticker: "{{s.scale}}" takes the record's number, [x, y] or {"x":, "y":} (members may be text of numbers), or text of one or two plain numbers such as a CSV field ("1.5", "1.5 0.8", "[1.5, 0.8]"); a record without it draws the block at its natural size, silently, and a value that is not a number or a pair is reported with the block's pointer (and the -merge record) and drawn at natural size. A string with no token is read as it stands.
`skew`	`number | array of number | object | string`		Shear the block, in DEGREES, about the same pivot as "angle", under the same restriction. [x, y] or {"x":, "y":}; a bare number skews x only. Positive x leans the top to the right and positive y lifts the right-hand side -- both counter-clockwise, matching "angle". Clamped to +/-89.9: a 90-degree skew is a singular matrix and draws nothing. A STRING is a data token, so each record of a template skews its own sticker: "{{s.skew}}" takes the record's number (x only), [x, y] or {"x":, "y":}, or text of one or two plain numbers; clamped as the written one is. A record without it draws the block unskewed, silently, and a value that is not a number or a pair is reported with the block's pointer (and the -merge record) and drawn unskewed. A string with no token is read as it stands.

Key	Type	Default	Description
straddle	boolean		Straddle across columns: when true and the body is multi-column, this block spans the FULL content width and text in every column wraps around it (above and below) - a spanning float generalized from center-floated images to any block. No effect in a single-column body, on a positioned / anchored block, or on an image/pdf already using float:"center" (the same mechanism). Does not balance the columns above it or act as a section divider.
title	string		Heading printed above the glossary.
verticalAlign	string		Where the content sits in its "position" box when it is shorter than the box's height: "top" (the default), "middle" ("center" is the same) or "bottom". A paragraph, heading or pre moves as a whole (its own box with it); a container keeps its box where it is and moves its children inside it (a child with its own "position" stays put). The offset is the box height less the content's height as it is actually placed -- the same lines, shrinkToFit size and padding that are then drawn -- halved for middle. Content as tall as the box or taller does not move: it is clipped and reported exactly as without the key. Needs a position with a height: on a block without one, on any other kind of block, on text set along a textPath "d" curve (stated in fractions of the whole box, so drawn across all of it; the "arc" form aligns), and on vertical text whose columns fill the box (columns run the full height unless a paragraph's or a heading's boxHeight is shorter), it is refused with a notice and ignored. The layout map keeps reporting the box. One of: top, middle, bottom, center.
z	number		CANVAS MODE ONLY (see the top-level "pages"): where this element sits in the PAINT order of its page. Lower draws first, further back; unset means "wherever I am in the list", and ties keep author order. It moves the PAINT and not the READING: the structure tree keeps the document order, so a caption written first because it is read first can still be painted last because it sits on top. That separation is what a flattened scene graph cannot express. Ignored outside canvas mode, where the flow already fixes both orders.

The bibliography element

Type name: bibliography. Accepted aliases: references.

A reference list built from structured entries (author, title, container, publisher, year, DOI...) or a pre-formatted text, cited from body runs with cite. The numeric style numbers entries in order of first citation (Vancouver) and renders markers as [1] or [1, 3]; the author-year style renders (Knuth 1984) and sorts the list alphabetically. Every citation links to its entry, and with backlink on each entry links back to its first citation. references is an accepted alias.

Keys

Key	Type	Default	Description
`backlink`	`boolean`	true	Make each entry's [n] label a link back to the first citation of it. Default true; numeric style only, since author-year draws no label.
`breakBefore`	`boolean`		-
`citedOnly`	`boolean`		Drop entries no `cite` run named. Defaults to TRUE under "numeric" (the list IS the citation order, so an uncited work has no number and no place in it) and FALSE under "author-year" (the marker carries the name, so uncited background reading belongs).
`entries`	`array of object`		The works. Every field is optional except `key` (which is what a `cite` run names); an entry with no key still prints but can never be cited.
`opacity`	`number`		Constant opacity for this block: 0 (invisible) to 1 (opaque, the default). REAL alpha -- a /CA /ca ExtGState in the PDF -- so a translucent panel over a photograph shows the photograph through it. Not to be confused with "style.background" {color, alpha}, which composites the tint toward WHITE at parse time and so only looks right over a white page. PER-ELEMENT, NOT GROUP: on a container it paints that container's own decoration -- background, border, accent bar, stripe -- and never its children, which draw at their own opacity (1 unless they set one). Two translucent shapes that overlap therefore composite twice, which is what per-element opacity means; group opacity, where a whole subtree is composited once and then faded, is a separate feature. Where a draw already carries its own alpha (an SVG fill-opacity, a gradient's alpha) the two MULTIPLY.
`title`	`string`		Heading printed above the reference list ("References", "Works Cited").

The endnotes element

Type name: `endnotes`.

Collects every `note` run in the document and renders them as a numbered end-matter list, each marker linking to its note and each note linking back - the alternative to page-bottom footnotes for journals and books that gather notes at the end.

Keys

Key	Type	Default	Description
`backlink`	`boolean`		Turn each note's number into a link back to its body marker (default true) -- the first place it is drawn, never a contents row's or a list-of row's copy of it, on every form of heading (run-in, curved and vertical draw their own marker). A contents row's copy stands in only for a marker a fixed-height box clips away while its heading is still listed. The forward link (marker to note) is always made.
`breakBefore`	`boolean`		Pagination: open a new page before this block - a part divider, an appendix. Ignored when the page is already empty, so it never leaves a blank page behind.
`condition`	`string`		Conditional text: render this block only when its condition-tag expression is shown under the document's conditions map.
`opacity`	`number`		Constant opacity for this block: 0 (invisible) to 1 (opaque, the default). REAL alpha -- a /CA /ca ExtGState in the PDF -- so a translucent panel over a photograph shows the photograph through it. Not to be confused with "style.background" {color, alpha}, which composites the tint toward WHITE at parse time and so only looks right over a white page. PER-ELEMENT, NOT GROUP: on a container it paints that container's own decoration -- background, border, accent bar, stripe -- and never its children, which draw at their own opacity (1 unless they set one). Two translucent shapes that overlap therefore composite twice, which is what per-element opacity means; group opacity, where a whole subtree is composited once and then faded, is a separate feature. Where a draw already carries its own alpha (an SVG fill-opacity, a gradient's alpha) the two MULTIPLY.
`title`	`string`		Heading printed above the gathered notes (e.g. "Notes").

11. Notes and asides

Set-off editorial material.

The callout element

Type name: `callout`. Accepted aliases: `admonition`, `note`.

A set-off note, tip or warning in a tinted, bordered box. Tagged as an `Aside`, so assistive technology announces it as an aside rather than as body text.

Example

```
{
  "type": "callout",
  "variant": "warning",
  "title": "Caution",
  "blocks": [
    {
      "type": "paragraph",
      "text": "Never approach a rocket that failed to launch."
    }
  ]
}
```

A tip

```
{
  "type": "callout",
  "variant": "tip",
  "blocks": [
    {
      "type": "paragraph",
      "text": "Render with -map to get the layout geometry back as JSON."
    }
  ]
}
```

Keys

Key	Type	Default	Description
`anchor`	`anchor`		-
`angle`	`number | string`		Turn the block, in degrees counter-clockwise, about the CENTRE of its "position" box. A CONTAINER in the flow may also carry this: it becomes a GROUP that turns about its own measured box and reserves the bounding box of the turned result, and its children inherit the transform. Any other flow block ignores it with a warning -- wrap it in a container. The block lays out upright inside its box and the finished result is turned, so a "position.height" that truncates content truncates the same text it would unrotated, and the turned result is NOT clipped to its box: it may extend past it, exactly as an unrotated positioned block may. Not to be confused with the image block's "rotate", which is an integer QUARTER TURN CLOCKWISE applied to the picture INSIDE its box (and re-sizes that box to the turned orientation). The two compose -- an image can stand a sideways scan upright with "rotate" and tilt the whole frame with "angle". Same meaning as the watermark's "angle". A STRING is a data token, so each record of a template turns its own sticker: "{{s.angle}}" takes the record's number (or text of one, such as a CSV field); a record without it leaves the block upright, silently, and a value that is not a number is reported with the block's pointer (and the -merge record) and drawn upright. A string with no token is read as it stands.

Key	Type	Default	Description
bleed	number		Step and Repeat: how far this placed block's box reaches past its TRIM on every side, in points -- the box IS the bleed box, and the trim is the box inset by this much. A design tool sizes a card by it (the trim plus twice the bleed) and draws the trim line inside each copy; the engine draws nothing from it, and keeps the bleed only through the block's own clip. Only a block with position has a trim: on any other block the key is refused with a notice naming the block.
blend	string		How this block's OWN ink combines with the page under it -- the sixteen PDF blend modes (ISO 32000-1 11.3.5), written as /BM. On a container that means its background, border and stripe, never its children; for the subtree as one object see "groupBlend". Case and separators are folded, so "colorDodge", "color-dodge" and "COLORDODGE" are one mode. An unrecognized name is refused with a warning and the block paints normally. See the blend-modes sample. One of: normal, multiply, screen, overlay, darken, lighten, colorDodge, colorBurn, hardLight, softLight, difference, exclusion, hue, saturation, ...
blocks	blockGroup		Alias for "body".
body	blockGroup		The call-out's content blocks.
breakBefore	boolean		Pagination: open a new page before this block - a part divider, an appendix. Ignored when the page is already empty, so it never leaves a blank page behind.
changed	boolean		Revision / change bar: draw a vertical bar in the page margin beside this block's content to flag it as changed. A document-level changeBars object customizes the bar (width / color / side / distance); a condition tag declared with changeBar:true sets this automatically for tagged content.
clip	boolean		Clip this block's children (on a container) or its own ink to its box, with the style's corner "radius". Only bites where content is BIGGER than the box: a flow container grows to fit its children vertically, so give it "position" with a height, or a "minHeight", before expecting a vertical crop. Container-scoped, like "groupOpacity": on any other flow block it is refused with a warning. For a crop to a SHAPE rather than to the box, see "clipPath"; giving both INTERSECTS them.
clipPath	object		Crop this block to an arbitrary SHAPE stated inside its box -- a design tool's FRAME. Same vocabulary as the "shape" block and read by the same outline builder, so a shape and a clipPath given identical parameters describe the identical outline. Scoped exactly as "clip" is: a positioned block of any type (it needs a "position.height", because the shape is stated inside the box) or a container in the flow; anywhere else it is refused with a warning. Naming "clip" as well INTERSECTS the two. The shape turns, scales and skews with the block that carries it.
condition	string		Conditional text: render this block only when its condition-tag expression is shown under the document's conditions map. Whitespace/comma tag terms, each optionally '!'-negated, AND-combined; an unlisted tag defaults to shown. Empty means always shown.

Key	Type	Default	Description
`float`	`string`		Float this block out of the stacked flow to one side of its column; content that FOLLOWS it wraps into the space beside it. Works on ANY block type -- a table, a container, a chart, a code listing, a figure. "center" straddles the page content width, so text in every column wraps around it. The float box takes the block's own "width"; a block that declares none takes HALF the column, so a float always leaves room beside it. Ignored on a block that also sets "position" or "anchor" -- those have already chosen their placement. Not to be confused with "anchor": { "float": true }, a BOOLEAN about migrating an anchored frame to the next column that can hold it. One of: `left`, `right`, `center`.
`groupBlend`	`string`		Composite this block AND its children once, then blend the RESULT with the page. NOT the same key as "blend", and the difference is larger than it is for opacity: two stacked children that each multiply the backdrop multiply it TWICE, which is darker than the group multiplying once, and no per-child setting reproduces it. That is what a design tool's blend-mode control on a group means. Container-scoped: on any other flow block it is refused with a warning. Also the way to blend an element together with its drop shadow, which is drawn as stacked alpha layers and so keeps its own normal compositing. See the blend-modes sample. One of: `normal`, `multiply`, `screen`, `overlay`, `darken`, `lighten`, `colorDodge`, `colorBurn`, `hardLight`, `softLight`, `difference`, `exclusion`, `hue`, `saturation`, ...
`groupOpacity`	`number`		Composite this block AND its children once, then fade the result -- true GROUP opacity. NOT the same key as "opacity", which fades each element as it paints: two overlapping opaque children at opacity 0.5 each show a seam where they cross, and under groupOpacity 0.5 they do not. The two compose. Container-scoped: on any other flow block it is refused with a warning. See the clipped-groups sample.
`icon`	`string`		Optional glyph prefixed to the label (renders only if the font has it).
`keepTogether`	`boolean`	true	Keep the box on one page when it fits (default true).
`label`	`string`		Alias for "title".
`mirrorH`	`boolean | string`		Mirror the whole block left-to-right about the centre of its "position" box, under the same restriction as "angle". Identical to scale x -1, and the two compose. NOT "flipH": that key belongs to the image block, where it mirrors the PICTURE INSIDE its frame. The two are separate words so an image can do both. A STRING is a data token, so each record of a template mirrors its own sticker: "{{s.flip}}" takes the record's boolean, or text "true" / "false" in any case, or 1 / 0 (a number or text); a record without it draws the block unmirrored, silently, and any other value ("yes" included) is reported with the block's pointer (and the -merge record) and drawn unmirrored. A string with no token is read as it stands.

Key	Type	Default	Description
`mirrorV`	`boolean | string`		Mirror the whole block top-to-bottom about the centre of its "position" box, under the same restriction as "angle". Identical to scale y -1. NOT "flipV" -- see "mirrorH". A STRING is a data token, so each record of a template mirrors its own sticker: "{{s.flip}}" takes the record's boolean, or text "true" / "false" in any case, or 1 / 0 (a number or text); a record without it draws the block unmirrored, silently, and any other value ("yes" included) is reported with the block's pointer (and the -merge record) and drawn unmirrored. A string with no token is read as it stands.
`opacity`	`number`		Constant opacity for this block: 0 (invisible) to 1 (opaque, the default). REAL alpha -- a /CA /ca ExtGState in the PDF -- so a translucent panel over a photograph shows the photograph through it. Not to be confused with "style.background" {color, alpha}, which composites the tint toward WHITE at parse time and so only looks right over a white page. PER-ELEMENT, NOT GROUP: on a container it paints that container's own decoration -- background, border, accent bar, stripe -- and never its children, which draw at their own opacity (1 unless they set one). Two translucent shapes that overlap therefore composite twice, which is what per-element opacity means; group opacity, where a whole subtree is composited once and then faded, is a separate feature. Where a draw already carries its own alpha (an SVG fill-opacity, a gradient's alpha) the two MULTIPLY.
`repeatAt`	`array of object`		Imposition: draw this block ONCE and place the drawing again at each of these offsets, in points from the block's own position, y DOWNWARD as every `position` is. One entry = one extra copy, so a list of 7 puts 8 cards on a sheet. The drawing is captured into a Form XObject and each offset is a `cm` + `Do` of that one stream, so n copies cost one copy of the content. Translation only -- `angle` / `scale` / `skew` are not applied per placement. A REPEATED FORM MAY NOT CARRY TAGGED CONTENT: ISO 14289-1 7.20 allows a form's content ONE semantic parent, so repeating tagged text (or a shape with `alt`) fails PDF/UA -- veraPDF catches it, ap_addstruct's audit does not. The engine warns and draws it anyway. Repeat decoration: a shape or svg with no `alt`, or put the block in a master's `frames`, which is drawn as a page artifact. Only a block with `position` is drawn again: on a block without one (in the flow, or an anchored frame), or on a repeat / if / include block, the key is refused with a notice naming the block. Each copy takes the block's `groupOpacity` and `groupBlend`. The layout map (-map) reports the copies as `repeats` on the block's fragment, so a design tool can select and outline them.

Key	Type	Default	Description
`scale`	`number | array of number | object | string`		Scale the block about the same pivot as "angle" (the centre of its "position" box), and under the same restriction. A CONTAINER in the flow may also carry this: it becomes a GROUP that turns about its own measured box and reserves the bounding box of the turned result, and its children inherit the transform. Any other flow block ignores it with a warning -- wrap it in a container. A number scales both axes; [x, y] or {"x":, "y":} scales them separately. 1 = natural size. Applied AFTER layout, so a scaled block does NOT re-wrap its text at the scaled measure -- this is a design tool's scale handle, not a change of font size. A STRING is a data token, so each record of a template scales its own sticker: "{{s.scale}}" takes the record's number, [x, y] or {"x":, "y":} (members may be text of numbers), or text of one or two plain numbers such as a CSV field ("1.5", "1.5 0.8", "[1.5, 0.8]"); a record without it draws the block at its natural size, silently, and a value that is not a number or a pair is reported with the block's pointer (and the -merge record) and drawn at natural size. A string with no token is read as it stands.
`showLabel`	`boolean`	true	Draw the label line (default true).
`skew`	`number | array of number | object | string`		Shear the block, in DEGREES, about the same pivot as "angle", under the same restriction. [x, y] or {"x":, "y":}; a bare number skews x only. Positive x leans the top to the right and positive y lifts the right-hand side -- both counter-clockwise, matching "angle". Clamped to +/-89.9: a 90-degree skew is a singular matrix and draws nothing. A STRING is a data token, so each record of a template skews its own sticker: "{{s.skew}}" takes the record's number (x only), [x, y] or {"x":, "y":}, or text of one or two plain numbers; clamped as the written one is. A record without it draws the block unskewed, silently, and a value that is not a number or a pair is reported with the block's pointer (and the -merge record) and drawn unskewed. A string with no token is read as it stands.
`straddle`	`boolean`		Straddle across columns: when true and the body is multi-column, this block spans the FULL content width and text in every column wraps around it (above and below) - a spanning float generalized from center-floated images to any block. No effect in a single-column body, on a positioned / anchored block, or on an image/pdf already using float:"center" (the same mechanism). Does not balance the columns above it or act as a section divider.
`text`	`string`		Shorthand body: a single paragraph. Use "body" for richer content.
`title`	`string`		Overrides the default variant label (the bold colored heading of the box).
`variant`	`string`	note	Selects the accent color, tint and default label (default "note"). One of: `note`, `info`, `tip`, `success`, `important`, `warning`, `caution`, `danger`.

Key	Type	Default	Description
`verticalAlign`	`string`		Where the content sits in its "position" box when it is shorter than the box's height: "top" (the default), "middle" ("center" is the same) or "bottom". A paragraph, heading or pre moves as a whole (its own box with it); a container keeps its box where it is and moves its children inside it (a child with its own "position" stays put). The offset is the box height less the content's height as it is actually placed -- the same lines, shrinkToFit size and padding that are then drawn -- halved for middle. Content as tall as the box or taller does not move: it is clipped and reported exactly as without the key. Needs a position with a height: on a block without one, on any other kind of block, on text set along a textPath "d" curve (stated in fractions of the whole box, so drawn across all of it; the "arc" form aligns), and on vertical text whose columns fill the box (columns run the full height unless a paragraph's or a heading's boxHeight is shorter), it is refused with a notice and ignored. The layout map keeps reporting the box. One of: `top`, `middle`, `bottom`, `center`.
`z`	`number`		CANVAS MODE ONLY (see the top-level "pages"): where this element sits in the PAINT order of its page. Lower draws first, further back; unset means "wherever I am in the list", and ties keep author order. It moves the PAINT and not the READING: the structure tree keeps the document order, so a caption written first because it is read first can still be painted last because it sits on top. That separation is what a flattened scene graph cannot express. Ignored outside canvas mode, where the flow already fixes both orders.

12. Forms

Interactive AcroForm fields.

The field element

Type name: `field`.

An interactive AcroForm field: text box, checkbox, radio group, dropdown, list box, button or signature. The field is a real form widget in the output PDF - fillable in any viewer - with a label, tooltip, default value and tab order.

Example

```
{
  "type": "field",
  "field": "text",
  "name": "customer",
  "label": "Customer name",
  "width": "60%",
  "tooltip": "Full legal name"
}
```

A radio-button group

```
{
  "type": "field",
  "field": "radio",
  "name": "size",
  "label": "Size",
  "options": [
    "Small",
    "Medium",
    "Large"
  ],
  "value": "Medium"
}
```

A signature field

```
{
  "type": "field",
  "field": "signature",
  "name": "approval",
  "label": "Approved by",
  "width": "60%"
}
```

Keys

Key	Type	Default	Description
`anchor`	`anchor`		-
`angle`	`number | string`		Turn the block, in degrees counter-clockwise, about the CENTRE of its "position" box. A CONTAINER in the flow may also carry this: it becomes a GROUP that turns about its own measured box and reserves the bounding box of the turned result, and its children inherit the transform. Any other flow block ignores it with a warning -- wrap it in a container. The block lays out upright inside its box and the finished result is turned, so a "position.height" that truncates content truncates the same text it would unrotated, and the turned result is NOT clipped to its box: it may extend past it, exactly as an unrotated positioned block may. Not to be confused with the image block's "rotate", which is an integer QUARTER TURN CLOCKWISE applied to the picture INSIDE its box (and re-sizes that box to the turned orientation). The two compose -- an image can stand a sideways scan upright with "rotate" and tilt the whole frame with "angle". Same meaning as the watermark's "angle". A STRING is a data token, so each record of a template turns its own sticker: "{{s.angle}}" takes the record's number (or text of one, such as a CSV field); a record without it leaves the block upright, silently, and a value that is not a number is reported with the block's pointer (and the -merge record) and drawn upright. A string with no token is read as it stands.

Key	Type	Default	Description
bleed	number		Step and Repeat: how far this placed block's box reaches past its TRIM on every side, in points -- the box IS the bleed box, and the trim is the box inset by this much. A design tool sizes a card by it (the trim plus twice the bleed) and draws the trim line inside each copy; the engine draws nothing from it, and keeps the bleed only through the block's own clip. Only a block with position has a trim: on any other block the key is refused with a notice naming the block.
blend	string		How this block's OWN ink combines with the page under it -- the sixteen PDF blend modes (ISO 32000-1 11.3.5), written as /BM. On a container that means its background, border and stripe, never its children; for the subtree as one object see "groupBlend". Case and separators are folded, so "colorDodge", "color-dodge" and "COLORDODGE" are one mode. An unrecognized name is refused with a warning and the block paints normally. See the blend-modes sample. One of: normal, multiply, screen, overlay, darken, lighten, colorDodge, colorBurn, hardLight, softLight, difference, exclusion, hue, saturation, ...
breakBefore	boolean		Pagination: open a new page before this block - a part divider, an appendix. Ignored when the page is already empty, so it never leaves a blank page behind.
changed	boolean		Revision / change bar: draw a vertical bar in the page margin beside this block's content to flag it as changed. A document-level changeBars object customizes the bar (width / color / side / distance); a condition tag declared with changeBar:true sets this automatically for tagged content.
clip	boolean		Clip this block's children (on a container) or its own ink to its box, with the style's corner "radius". Only bites where content is BIGGER than the box: a flow container grows to fit its children vertically, so give it "position" with a height, or a "minHeight", before expecting a vertical crop. Container-scoped, like "groupOpacity": on any other flow block it is refused with a warning. For a crop to a SHAPE rather than to the box, see "clipPath"; giving both INTERSECTS them.
clipPath	object		Crop this block to an arbitrary SHAPE stated inside its box -- a design tool's FRAME. Same vocabulary as the "shape" block and read by the same outline builder, so a shape and a clipPath given identical parameters describe the identical outline. Scoped exactly as "clip" is: a positioned block of any type (it needs a "position.height", because the shape is stated inside the box) or a container in the flow; anywhere else it is refused with a warning. Naming "clip" as well INTERSECTS the two. The shape turns, scales and skews with the block that carries it.
condition	string		Conditional text: render this block only when its condition-tag expression is shown under the document's conditions map. Whitespace/comma tag terms, each optionally '!'-negated, AND-combined; an unlisted tag defaults to shown. Empty means always shown.

Key	Type	Default	Description
`float`	`string`		Float this block out of the stacked flow to one side of its column; content that FOLLOWS it wraps into the space beside it. Works on ANY block type -- a table, a container, a chart, a code listing, a figure. "center" straddles the page content width, so text in every column wraps around it. The float box takes the block's own "width"; a block that declares none takes HALF the column, so a float always leaves room beside it. Ignored on a block that also sets "position" or "anchor" -- those have already chosen their placement. Not to be confused with "anchor": { "float": true }, a BOOLEAN about migrating an anchored frame to the next column that can hold it. One of: `left`, `right`, `center`.
`groupBlend`	`string`		Composite this block AND its children once, then blend the RESULT with the page. NOT the same key as "blend", and the difference is larger than it is for opacity: two stacked children that each multiply the backdrop multiply it TWICE, which is darker than the group multiplying once, and no per-child setting reproduces it. That is what a design tool's blend-mode control on a group means. Container-scoped: on any other flow block it is refused with a warning. Also the way to blend an element together with its drop shadow, which is drawn as stacked alpha layers and so keeps its own normal compositing. See the blend-modes sample. One of: `normal`, `multiply`, `screen`, `overlay`, `darken`, `lighten`, `colorDodge`, `colorBurn`, `hardLight`, `softLight`, `difference`, `exclusion`, `hue`, `saturation`, ...
`groupOpacity`	`number`		Composite this block AND its children once, then fade the result -- true GROUP opacity. NOT the same key as "opacity", which fades each element as it paints: two overlapping opaque children at opacity 0.5 each show a seam where they cross, and under groupOpacity 0.5 they do not. The two compose. Container-scoped: on any other flow block it is refused with a warning. See the clipped-groups sample.
`height`	`number`		Height of the field control in points.
`label`	`string`		Visible label text next to the widget.
`labelPos`	`string`		Where the field's label sits relative to the control: before it, after it, or not drawn. One of: `before`, `after`, `none`.
`mirrorH`	`boolean | string`		Mirror the whole block left-to-right about the centre of its "position" box, under the same restriction as "angle". Identical to scale x -1, and the two compose. NOT "flipH": that key belongs to the image block, where it mirrors the PICTURE INSIDE its frame. The two are separate words so an image can do both. A STRING is a data token, so each record of a template mirrors its own sticker: "{{s.flip}}" takes the record's boolean, or text "true" / "false" in any case, or 1 / 0 (a number or text); a record without it draws the block unmirrored, silently, and any other value ("yes" included) is reported with the block's pointer (and the -merge record) and drawn unmirrored. A string with no token is read as it stands.

Key	Type	Default	Description
`mirrorV`	`boolean | string`		Mirror the whole block top-to-bottom about the centre of its "position" box, under the same restriction as "angle". Identical to scale y -1. NOT "flipV" -- see "mirrorH". A STRING is a data token, so each record of a template mirrors its own sticker: "{{s.flip}}" takes the record's boolean, or text "true" / "false" in any case, or 1 / 0 (a number or text); a record without it draws the block unmirrored, silently, and any other value ("yes" included) is reported with the block's pointer (and the -merge record) and drawn unmirrored. A string with no token is read as it stands.
`opacity`	`number`		Constant opacity for this block: 0 (invisible) to 1 (opaque, the default). REAL alpha -- a /CA /ca ExtGState in the PDF -- so a translucent panel over a photograph shows the photograph through it. Not to be confused with "style.background" {color, alpha}, which composites the tint toward WHITE at parse time and so only looks right over a white page. PER-ELEMENT, NOT GROUP: on a container it paints that container's own decoration -- background, border, accent bar, stripe -- and never its children, which draw at their own opacity (1 unless they set one). Two translucent shapes that overlap therefore composite twice, which is what per-element opacity means; group opacity, where a whole subtree is composited once and then faded, is a separate feature. Where a draw already carries its own alpha (an SVG fill-opacity, a gradient's alpha) the two MULTIPLY.
`repeatAt`	`array of object`		Imposition: draw this block ONCE and place the drawing again at each of these offsets, in points from the block's own position, y DOWNWARD as every `position` is. One entry = one extra copy, so a list of 7 puts 8 cards on a sheet. The drawing is captured into a Form XObject and each offset is a `cm` + `Do` of that one stream, so n copies cost one copy of the content. Translation only -- `angle` / `scale` / `skew` are not applied per placement. A REPEATED FORM MAY NOT CARRY TAGGED CONTENT: ISO 14289-1 7.20 allows a form's content ONE semantic parent, so repeating tagged text (or a shape with `alt`) fails PDF/UA -- veraPDF catches it, ap_addstruct's audit does not. The engine warns and draws it anyway. Repeat decoration: a shape or svg with no `alt`, or put the block in a master's `frames`, which is drawn as a page artifact. Only a block with `position` is drawn again: on a block without one (in the flow, or an anchored frame), or on a repeat / if / include block, the key is refused with a notice naming the block. Each copy takes the block's `groupOpacity` and `groupBlend`. The layout map (-map) reports the copies as `repeats` on the block's fragment, so a design tool can select and outline them.

Key	Type	Default	Description
`scale`	`number | array of number | object | string`		Scale the block about the same pivot as "angle" (the centre of its "position" box), and under the same restriction. A CONTAINER in the flow may also carry this: it becomes a GROUP that turns about its own measured box and reserves the bounding box of the turned result, and its children inherit the transform. Any other flow block ignores it with a warning -- wrap it in a container. A number scales both axes; [x, y] or {"x":, "y":} scales them separately. 1 = natural size. Applied AFTER layout, so a scaled block does NOT re-wrap its text at the scaled measure -- this is a design tool's scale handle, not a change of font size. A STRING is a data token, so each record of a template scales its own sticker: "{{s.scale}}" takes the record's number, [x, y] or {"x":, "y":} (members may be text of numbers), or text of one or two plain numbers such as a CSV field ("1.5", "1.5 0.8", "[1.5, 0.8]"); a record without it draws the block at its natural size, silently, and a value that is not a number or a pair is reported with the block's pointer (and the -merge record) and drawn at natural size. A string with no token is read as it stands.
`skew`	`number | array of number | object | string`		Shear the block, in DEGREES, about the same pivot as "angle", under the same restriction. [x, y] or {"x":, "y":}; a bare number skews x only. Positive x leans the top to the right and positive y lifts the right-hand side -- both counter-clockwise, matching "angle". Clamped to +/-89.9: a 90-degree skew is a singular matrix and draws nothing. A STRING is a data token, so each record of a template skews its own sticker: "{{s.skew}}" takes the record's number (x only), [x, y] or {"x":, "y":}, or text of one or two plain numbers; clamped as the written one is. A record without it draws the block unskewed, silently, and a value that is not a number or a pair is reported with the block's pointer (and the -merge record) and drawn unskewed. A string with no token is read as it stands.
`straddle`	`boolean`		Straddle across columns: when true and the body is multi-column, this block spans the FULL content width and text in every column wraps around it (above and below) - a spanning float generalized from center-floated images to any block. No effect in a single-column body, on a positioned / anchored block, or on an image/pdf already using float:"center" (the same mechanism). Does not balance the columns above it or act as a section divider.
`verticalAlign`	`string`		Where the content sits in its "position" box when it is shorter than the box's height: "top" (the default), "middle" ("center" is the same) or "bottom". A paragraph, heading or pre moves as a whole (its own box with it); a container keeps its box where it is and moves its children inside it (a child with its own "position" stays put). The offset is the box height less the content's height as it is actually placed -- the same lines, shrinkToFit size and padding that are then drawn -- halved for middle. Content as tall as the box or taller does not move: it is clipped and reported exactly as without the key. Needs a position with a height: on a block without one, on any other kind of block, on text set along a textPath "d" curve (stated in fractions of the whole box, so drawn across all of it; the "arc" form aligns), and on vertical text whose columns fill the box (columns run the full height unless a paragraph's or a heading's boxHeight is shorter), it is refused with a notice and ignored. The layout map keeps reporting the box. One of: `top`, `middle`, `bottom`, `center`.
`width`	`dimension`		Widget width (auto = fill the row after the label).

Key	Type	Default	Description
z	number		CANVAS MODE ONLY (see the top-level "pages"): where this element sits in the PAINT order of its page. Lower draws first, further back; unset means "wherever I am in the list", and ties keep author order. It moves the PAINT and not the READING: the structure tree keeps the document order, so a caption written first because it is read first can still be painted last because it sits on top. That separation is what a flattened scene graph cannot express. Ignored outside canvas mode, where the flow already fixes both orders.

13. Data-driven documents

Elements that build content from a data context.

The repeat element

Type name: `repeat`.

Builds content from data: names an array in the data context and clones its template blocks once per element, binding `{{...}}` placeholders as it goes. One template, N rendered sections.

Example

```
{
  "type": "repeat",
  "for": "lineItems",
  "as": "item",
  "blocks": [
    {
      "type": "paragraph",
      "text": "{{item.sku}}  -  {{item.qty}}"
    }
  ]
}
```

Keys

Key	Type	Default	Description
anchor	anchor		-

Key	Type	Default	Description
angle	number | string		Turn the block, in degrees counter-clockwise, about the CENTRE of its "position" box. A CONTAINER in the flow may also carry this: it becomes a GROUP that turns about its own measured box and reserves the bounding box of the turned result, and its children inherit the transform. Any other flow block ignores it with a warning -- wrap it in a container. The block lays out upright inside its box and the finished result is turned, so a "position.height" that truncates content truncates the same text it would unrotated, and the turned result is NOT clipped to its box: it may extend past it, exactly as an unrotated positioned block may. Not to be confused with the image block's "rotate", which is an integer QUARTER TURN CLOCKWISE applied to the picture INSIDE its box (and re-sizes that box to the turned orientation). The two compose -- an image can stand a sideways scan upright with "rotate" and tilt the whole frame with "angle". Same meaning as the watermark's "angle". A STRING is a data token, so each record of a template turns its own sticker: "{{s.angle}}" takes the record's number (or text of one, such as a CSV field); a record without it leaves the block upright, silently, and a value that is not a number is reported with the block's pointer (and the -merge record) and drawn upright. A string with no token is read as it stands.
as	string		Loop-variable name bound to each element (default "item"); a "loop" helper exposes index / number / count / first / last.
bleed	number		Step and Repeat: how far this placed block's box reaches past its TRIM on every side, in points -- the box IS the bleed box, and the trim is the box inset by this much. A design tool sizes a card by it (the trim plus twice the bleed) and draws the trim line inside each copy; the engine draws nothing from it, and keeps the bleed only through the block's own clip. Only a block with position has a trim: on any other block the key is refused with a notice naming the block.
blend	string		How this block's OWN ink combines with the page under it -- the sixteen PDF blend modes (ISO 32000-1 11.3.5), written as /BM. On a container that means its background, border and stripe, never its children; for the subtree as one object see "groupBlend". Case and separators are folded, so "colorDodge", "color-dodge" and "COLORDODGE" are one mode. An unrecognized name is refused with a warning and the block paints normally. See the blend-modes sample. One of: normal, multiply, screen, overlay, darken, lighten, colorDodge, colorBurn, hardLight, softLight, difference, exclusion, hue, saturation, ...
blocks	array of block		Template blocks cloned and bound once per array element.
breakBefore	boolean		Pagination: open a new page before this block - a part divider, an appendix. Ignored when the page is already empty, so it never leaves a blank page behind.
changed	boolean		Revision / change bar: draw a vertical bar in the page margin beside this block's content to flag it as changed. A document-level changeBars object customizes the bar (width / color / side / distance); a condition tag declared with changeBar:true sets this automatically for tagged content.

Key	Type	Default	Description
`clip`	`boolean`		Clip this block's children (on a container) or its own ink to its box, with the style's corner "radius". Only bites where content is BIGGER than the box: a flow container grows to fit its children vertically, so give it "position" with a height, or a "minHeight", before expecting a vertical crop. Container-scoped, like "groupOpacity": on any other flow block it is refused with a warning. For a crop to a SHAPE rather than to the box, see "clipPath"; giving both INTERSECTS them.
`clipPath`	`object`		Crop this block to an arbitrary SHAPE stated inside its box -- a design tool's FRAME. Same vocabulary as the "shape" block and read by the same outline builder, so a shape and a clipPath given identical parameters describe the identical outline. Scoped exactly as "clip" is: a positioned block of any type (it needs a "position.height", because the shape is stated inside the box) or a container in the flow; anywhere else it is refused with a warning. Naming "clip" as well INTERSECTS the two. The shape turns, scales and skews with the block that carries it.
`condition`	`string`		Conditional text: render this block only when its condition-tag expression is shown under the document's conditions map. Whitespace/comma tag terms, each optionally '!'-negated, AND-combined; an unlisted tag defaults to shown. Empty means always shown.
`float`	`string`		Float this block out of the stacked flow to one side of its column; content that FOLLOWS it wraps into the space beside it. Works on ANY block type -- a table, a container, a chart, a code listing, a figure. "center" straddles the page content width, so text in every column wraps around it. The float box takes the block's own "width"; a block that declares none takes HALF the column, so a float always leaves room beside it. Ignored on a block that also sets "position" or "anchor" -- those have already chosen their placement. Not to be confused with "anchor": { "float": true }, a BOOLEAN about migrating an anchored frame to the next column that can hold it. One of: `left`, `right`, `center`.
`for`	`string`		**Required.** Data-driven repeat: path to the array to iterate (e.g. "items", "order.lines"). The "blocks" template is cloned once per element.
`groupBlend`	`string`		Composite this block AND its children once, then blend the RESULT with the page. NOT the same key as "blend", and the difference is larger than it is for opacity: two stacked children that each multiply the backdrop multiply it TWICE, which is darker than the group multiplying once, and no per-child setting reproduces it. That is what a design tool's blend-mode control on a group means. Container-scoped: on any other flow block it is refused with a warning. Also the way to blend an element together with its drop shadow, which is drawn as stacked alpha layers and so keeps its own normal compositing. See the blend-modes sample. One of: `normal`, `multiply`, `screen`, `overlay`, `darken`, `lighten`, `colorDodge`, `colorBurn`, `hardLight`, `softLight`, `difference`, `exclusion`, `hue`, `saturation`, ...

Key	Type	Default	Description
groupOpacity	number		Composite this block AND its children once, then fade the result -- true GROUP opacity. NOT the same key as "opacity", which fades each element as it paints: two overlapping opaque children at opacity 0.5 each show a seam where they cross, and under groupOpacity 0.5 they do not. The two compose. Container-scoped: on any other flow block it is refused with a warning. See the clipped-groups sample.
mirrorH	boolean | string		Mirror the whole block left-to-right about the centre of its "position" box, under the same restriction as "angle". Identical to scale x -1, and the two compose. NOT "flipH": that key belongs to the image block, where it mirrors the PICTURE INSIDE its frame. The two are separate words so an image can do both. A STRING is a data token, so each record of a template mirrors its own sticker: "{{s.flip}}" takes the record's boolean, or text "true" / "false" in any case, or 1 / 0 (a number or text); a record without it draws the block unmirrored, silently, and any other value ("yes" included) is reported with the block's pointer (and the -merge record) and drawn unmirrored. A string with no token is read as it stands.
mirrorV	boolean | string		Mirror the whole block top-to-bottom about the centre of its "position" box, under the same restriction as "angle". Identical to scale y -1. NOT "flipV" -- see "mirrorH". A STRING is a data token, so each record of a template mirrors its own sticker: "{{s.flip}}" takes the record's boolean, or text "true" / "false" in any case, or 1 / 0 (a number or text); a record without it draws the block unmirrored, silently, and any other value ("yes" included) is reported with the block's pointer (and the -merge record) and drawn unmirrored. A string with no token is read as it stands.
opacity	number		Constant opacity for this block: 0 (invisible) to 1 (opaque, the default). REAL alpha -- a /CA /ca ExtGState in the PDF -- so a translucent panel over a photograph shows the photograph through it. Not to be confused with "style.background" {color, alpha}, which composites the tint toward WHITE at parse time and so only looks right over a white page. PER-ELEMENT, NOT GROUP: on a container it paints that container's own decoration -- background, border, accent bar, stripe -- and never its children, which draw at their own opacity (1 unless they set one). Two translucent shapes that overlap therefore composite twice, which is what per-element opacity means; group opacity, where a whole subtree is composited once and then faded, is a separate feature. Where a draw already carries its own alpha (an SVG fill-opacity, a gradient's alpha) the two MULTIPLY.

Key	Type	Default	Description
`repeatAt`	`array of object`		Imposition: draw this block ONCE and place the drawing again at each of these offsets, in points from the block's own position, y DOWNWARD as every `position` is. One entry = one extra copy, so a list of 7 puts 8 cards on a sheet. The drawing is captured into a Form XObject and each offset is a `cm` + `Do` of that one stream, so n copies cost one copy of the content. Translation only -- `angle` / `scale` / `skew` are not applied per placement. A REPEATED FORM MAY NOT CARRY TAGGED CONTENT: ISO 14289-1 7.20 allows a form's content ONE semantic parent, so repeating tagged text (or a shape with `alt`) fails PDF/UA -- veraPDF catches it, ap_addstruct's audit does not. The engine warns and draws it anyway. Repeat decoration: a shape or svg with no `alt`, or put the block in a master's `frames`, which is drawn as a page artifact. Only a block with `position` is drawn again: on a block without one (in the flow, or an anchored frame), or on a repeat / if / include block, the key is refused with a notice naming the block. Each copy takes the block's `groupOpacity` and `groupBlend`. The layout map (-map) reports the copies as `repeats` on the block's fragment, so a design tool can select and outline them.
`scale`	`number | array of number | object | string`		Scale the block about the same pivot as "angle" (the centre of its "position" box), and under the same restriction. A CONTAINER in the flow may also carry this: it becomes a GROUP that turns about its own measured box and reserves the bounding box of the turned result, and its children inherit the transform. Any other flow block ignores it with a warning -- wrap it in a container. A number scales both axes; [x, y] or {"x":, "y":} scales them separately. 1 = natural size. Applied AFTER layout, so a scaled block does NOT re-wrap its text at the scaled measure -- this is a design tool's scale handle, not a change of font size. A STRING is a data token, so each record of a template scales its own sticker: "{{s.scale}}" takes the record's number, [x, y] or {"x":, "y":} (members may be text of numbers), or text of one or two plain numbers such as a CSV field ("1.5", "1.5 0.8", "[1.5, 0.8]"); a record without it draws the block at its natural size, silently, and a value that is not a number or a pair is reported with the block's pointer (and the -merge record) and drawn at natural size. A string with no token is read as it stands.
`skew`	`number | array of number | object | string`		Shear the block, in DEGREES, about the same pivot as "angle", under the same restriction. [x, y] or {"x":, "y":}; a bare number skews x only. Positive x leans the top to the right and positive y lifts the right-hand side -- both counter-clockwise, matching "angle". Clamped to +/-89.9: a 90-degree skew is a singular matrix and draws nothing. A STRING is a data token, so each record of a template skews its own sticker: "{{s.skew}}" takes the record's number (x only), [x, y] or {"x":, "y":}, or text of one or two plain numbers; clamped as the written one is. A record without it draws the block unskewed, silently, and a value that is not a number or a pair is reported with the block's pointer (and the -merge record) and drawn unskewed. A string with no token is read as it stands.

Key	Type	Default	Description
straddle	boolean		Straddle across columns: when true and the body is multi-column, this block spans the FULL content width and text in every column wraps around it (above and below) - a spanning float generalized from center-floated images to any block. No effect in a single-column body, on a positioned / anchored block, or on an image/pdf already using float:"center" (the same mechanism). Does not balance the columns above it or act as a section divider.
verticalAlign	string		Where the content sits in its "position" box when it is shorter than the box's height: "top" (the default), "middle" ("center" is the same) or "bottom". A paragraph, heading or pre moves as a whole (its own box with it); a container keeps its box where it is and moves its children inside it (a child with its own "position" stays put). The offset is the box height less the content's height as it is actually placed -- the same lines, shrinkToFit size and padding that are then drawn -- halved for middle. Content as tall as the box or taller does not move: it is clipped and reported exactly as without the key. Needs a position with a height: on a block without one, on any other kind of block, on text set along a textPath "d" curve (stated in fractions of the whole box, so drawn across all of it; the "arc" form aligns), and on vertical text whose columns fill the box (columns run the full height unless a paragraph's or a heading's boxHeight is shorter), it is refused with a notice and ignored. The layout map keeps reporting the box. One of: top, middle, bottom, center.
z	number		CANVAS MODE ONLY (see the top-level "pages"): where this element sits in the PAINT order of its page. Lower draws first, further back; unset means "wherever I am in the list", and ties keep author order. It moves the PAINT and not the READING: the structure tree keeps the document order, so a caption written first because it is read first can still be painted last because it sits on top. That separation is what a flattened scene graph cannot express. Ignored outside canvas mode, where the flow already fixes both orders.

The if element

Type name: `if`.

Includes its blocks only when a data predicate holds. The same test is available on *any* block as the `when` and `unless` keys; the `if` element is the form to reach for when you want to gate a whole group at once.

Example

```
{
  "type": "if",
  "when": "total > 1000",
  "blocks": [
    {
      "type": "paragraph",
      "text": "Volume discount applied."
    }
  ]
}
```

Keys

Key	Type	Default	Description
anchor	anchor		-
angle	number | string		Turn the block, in degrees counter-clockwise, about the CENTRE of its "position" box. A CONTAINER in the flow may also carry this: it becomes a GROUP that turns about its own measured box and reserves the bounding box of the turned result, and its children inherit the transform. Any other flow block ignores it with a warning -- wrap it in a container. The block lays out upright inside its box and the finished result is turned, so a "position.height" that truncates content truncates the same text it would unrotated, and the turned result is NOT clipped to its box: it may extend past it, exactly as an unrotated positioned block may. Not to be confused with the image block's "rotate", which is an integer QUARTER TURN CLOCKWISE applied to the picture INSIDE its box (and re-sizes that box to the turned orientation). The two compose -- an image can stand a sideways scan upright with "rotate" and tilt the whole frame with "angle". Same meaning as the watermark's "angle". A STRING is a data token, so each record of a template turns its own sticker: "{{s.angle}}" takes the record's number (or text of one, such as a CSV field); a record without it leaves the block upright, silently, and a value that is not a number is reported with the block's pointer (and the -merge record) and drawn upright. A string with no token is read as it stands.
bleed	number		Step and Repeat: how far this placed block's box reaches past its TRIM on every side, in points -- the box IS the bleed box, and the trim is the box inset by this much. A design tool sizes a card by it (the trim plus twice the bleed) and draws the trim line inside each copy; the engine draws nothing from it, and keeps the bleed only through the block's own clip. Only a block with position has a trim: on any other block the key is refused with a notice naming the block.
blend	string		How this block's OWN ink combines with the page under it -- the sixteen PDF blend modes (ISO 32000-1 11.3.5), written as /BM. On a container that means its background, border and stripe, never its children; for the subtree as one object see "groupBlend". Case and separators are folded, so "colorDodge", "color-dodge" and "COLORDODGE" are one mode. An unrecognized name is refused with a warning and the block paints normally. See the blend-modes sample. One of: normal, multiply, screen, overlay, darken, lighten, colorDodge, colorBurn, hardLight, softLight, difference, exclusion, hue, saturation, ...
blocks	array of block		Emitted when the test is true.
breakBefore	boolean		Pagination: open a new page before this block - a part divider, an appendix. Ignored when the page is already empty, so it never leaves a blank page behind.
changed	boolean		Revision / change bar: draw a vertical bar in the page margin beside this block's content to flag it as changed. A document-level changeBars object customizes the bar (width / color / side / distance); a condition tag declared with changeBar:true sets this automatically for tagged content.

Key	Type	Default	Description
`clip`	`boolean`		Clip this block's children (on a container) or its own ink to its box, with the style's corner "radius". Only bites where content is BIGGER than the box: a flow container grows to fit its children vertically, so give it "position" with a height, or a "minHeight", before expecting a vertical crop. Container-scoped, like "groupOpacity": on any other flow block it is refused with a warning. For a crop to a SHAPE rather than to the box, see "clipPath"; giving both INTERSECTS them.
`clipPath`	`object`		Crop this block to an arbitrary SHAPE stated inside its box -- a design tool's FRAME. Same vocabulary as the "shape" block and read by the same outline builder, so a shape and a clipPath given identical parameters describe the identical outline. Scoped exactly as "clip" is: a positioned block of any type (it needs a "position.height", because the shape is stated inside the box) or a container in the flow; anywhere else it is refused with a warning. Naming "clip" as well INTERSECTS the two. The shape turns, scales and skews with the block that carries it.
`condition`	`string`		Conditional text: render this block only when its condition-tag expression is shown under the document's conditions map. Whitespace/comma tag terms, each optionally '!'-negated, AND-combined; an unlisted tag defaults to shown. Empty means always shown.
`else`	`array of block`		Emitted when the test is false.
`float`	`string`		Float this block out of the stacked flow to one side of its column; content that FOLLOWS it wraps into the space beside it. Works on ANY block type -- a table, a container, a chart, a code listing, a figure. "center" straddles the page content width, so text in every column wraps around it. The float box takes the block's own "width"; a block that declares none takes HALF the column, so a float always leaves room beside it. Ignored on a block that also sets "position" or "anchor" -- those have already chosen their placement. Not to be confused with "anchor": { "float": true }, a BOOLEAN about migrating an anchored frame to the next column that can hold it. One of: `left`, `right`, `center`.
`groupBlend`	`string`		Composite this block AND its children once, then blend the RESULT with the page. NOT the same key as "blend", and the difference is larger than it is for opacity: two stacked children that each multiply the backdrop multiply it TWICE, which is darker than the group multiplying once, and no per-child setting reproduces it. That is what a design tool's blend-mode control on a group means. Container-scoped: on any other flow block it is refused with a warning. Also the way to blend an element together with its drop shadow, which is drawn as stacked alpha layers and so keeps its own normal compositing. See the blend-modes sample. One of: `normal`, `multiply`, `screen`, `overlay`, `darken`, `lighten`, `colorDodge`, `colorBurn`, `hardLight`, `softLight`, `difference`, `exclusion`, `hue`, `saturation`, ...

Key	Type	Default	Description
groupOpacity	number		Composite this block AND its children once, then fade the result -- true GROUP opacity. NOT the same key as "opacity", which fades each element as it paints: two overlapping opaque children at opacity 0.5 each show a seam where they cross, and under groupOpacity 0.5 they do not. The two compose. Container-scoped: on any other flow block it is refused with a warning. See the clipped-groups sample.
mirrorH	boolean | string		Mirror the whole block left-to-right about the centre of its "position" box, under the same restriction as "angle". Identical to scale x -1, and the two compose. NOT "flipH": that key belongs to the image block, where it mirrors the PICTURE INSIDE its frame. The two are separate words so an image can do both. A STRING is a data token, so each record of a template mirrors its own sticker: "{{s.flip}}" takes the record's boolean, or text "true" / "false" in any case, or 1 / 0 (a number or text); a record without it draws the block unmirrored, silently, and any other value ("yes" included) is reported with the block's pointer (and the -merge record) and drawn unmirrored. A string with no token is read as it stands.
mirrorV	boolean | string		Mirror the whole block top-to-bottom about the centre of its "position" box, under the same restriction as "angle". Identical to scale y -1. NOT "flipV" -- see "mirrorH". A STRING is a data token, so each record of a template mirrors its own sticker: "{{s.flip}}" takes the record's boolean, or text "true" / "false" in any case, or 1 / 0 (a number or text); a record without it draws the block unmirrored, silently, and any other value ("yes" included) is reported with the block's pointer (and the -merge record) and drawn unmirrored. A string with no token is read as it stands.
opacity	number		Constant opacity for this block: 0 (invisible) to 1 (opaque, the default). REAL alpha -- a /CA /ca ExtGState in the PDF -- so a translucent panel over a photograph shows the photograph through it. Not to be confused with "style.background" {color, alpha}, which composites the tint toward WHITE at parse time and so only looks right over a white page. PER-ELEMENT, NOT GROUP: on a container it paints that container's own decoration -- background, border, accent bar, stripe -- and never its children, which draw at their own opacity (1 unless they set one). Two translucent shapes that overlap therefore composite twice, which is what per-element opacity means; group opacity, where a whole subtree is composited once and then faded, is a separate feature. Where a draw already carries its own alpha (an SVG fill-opacity, a gradient's alpha) the two MULTIPLY.

Key	Type	Default	Description
repeatAt	array of object		Imposition: draw this block ONCE and place the drawing again at each of these offsets, in points from the block's own position, y DOWNWARD as every position is. One entry = one extra copy, so a list of 7 puts 8 cards on a sheet. The drawing is captured into a Form XObject and each offset is a cm + Do of that one stream, so n copies cost one copy of the content. Translation only -- angle / scale / skew are not applied per placement. A REPEATED FORM MAY NOT CARRY TAGGED CONTENT: ISO 14289-1 7.20 allows a form's content ONE semantic parent, so repeating tagged text (or a shape with alt) fails PDF/UA -- veraPDF catches it, ap_addstruct's audit does not. The engine warns and draws it anyway. Repeat decoration: a shape or svg with no alt, or put the block in a master's frames, which is drawn as a page artifact. Only a block with position is drawn again: on a block without one (in the flow, or an anchored frame), or on a repeat / if / include block, the key is refused with a notice naming the block. Each copy takes the block's groupOpacity and groupBlend. The layout map (-map) reports the copies as repeats on the block's fragment, so a design tool can select and outline them.
scale	number | array of number | object | string		Scale the block about the same pivot as "angle" (the centre of its "position" box), and under the same restriction. A CONTAINER in the flow may also carry this: it becomes a GROUP that turns about its own measured box and reserves the bounding box of the turned result, and its children inherit the transform. Any other flow block ignores it with a warning -- wrap it in a container. A number scales both axes; [x, y] or {"x":, "y":} scales them separately. 1 = natural size. Applied AFTER layout, so a scaled block does NOT re-wrap its text at the scaled measure -- this is a design tool's scale handle, not a change of font size. A STRING is a data token, so each record of a template scales its own sticker: "{{s.scale}}" takes the record's number, [x, y] or {"x":, "y":} (members may be text of numbers), or text of one or two plain numbers such as a CSV field ("1.5", "1.5 0.8", "[1.5, 0.8]"); a record without it draws the block at its natural size, silently, and a value that is not a number or a pair is reported with the block's pointer (and the -merge record) and drawn at natural size. A string with no token is read as it stands.
skew	number | array of number | object | string		Shear the block, in DEGREES, about the same pivot as "angle", under the same restriction. [x, y] or {"x":, "y":}; a bare number skews x only. Positive x leans the top to the right and positive y lifts the right-hand side -- both counter-clockwise, matching "angle". Clamped to +/-89.9: a 90-degree skew is a singular matrix and draws nothing. A STRING is a data token, so each record of a template skews its own sticker: "{{s.skew}}" takes the record's number (x only), [x, y] or {"x":, "y":}, or text of one or two plain numbers; clamped as the written one is. A record without it draws the block unskewed, silently, and a value that is not a number or a pair is reported with the block's pointer (and the -merge record) and drawn unskewed. A string with no token is read as it stands.

Key	Type	Default	Description
straddle	boolean		Straddle across columns: when true and the body is multi-column, this block spans the FULL content width and text in every column wraps around it (above and below) - a spanning float generalized from center-floated images to any block. No effect in a single-column body, on a positioned / anchored block, or on an image/pdf already using float:"center" (the same mechanism). Does not balance the columns above it or act as a section divider.
test	string		Predicate: a path (truthiness) or "path OP value" (== != < > <= >=), optional leading !.
verticalAlign	string		Where the content sits in its "position" box when it is shorter than the box's height: "top" (the default), "middle" ("center" is the same) or "bottom". A paragraph, heading or pre moves as a whole (its own box with it); a container keeps its box where it is and moves its children inside it (a child with its own "position" stays put). The offset is the box height less the content's height as it is actually placed -- the same lines, shrinkToFit size and padding that are then drawn -- halved for middle. Content as tall as the box or taller does not move: it is clipped and reported exactly as without the key. Needs a position with a height: on a block without one, on any other kind of block, on text set along a textPath "d" curve (stated in fractions of the whole box, so drawn across all of it; the "arc" form aligns), and on vertical text whose columns fill the box (columns run the full height unless a paragraph's or a heading's boxHeight is shorter), it is refused with a notice and ignored. The layout map keeps reporting the box. One of: top, middle, bottom, center.
z	number		CANVAS MODE ONLY (see the top-level "pages"): where this element sits in the PAINT order of its page. Lower draws first, further back; unset means "wherever I am in the list", and ties keep author order. It moves the PAINT and not the READING: the structure tree keeps the document order, so a caption written first because it is read first can still be painted last because it sits on top. That separation is what a flattened scene graph cannot express. Ignored outside canvas mode, where the flow already fixes both orders.

The include element

Type name: include.
Pulls in another spec file as a reusable partial, so shared front matter, a standard cover or a boilerplate appendix lives in one place.

Example

```
{
  "type": "include",
  "src": "shared/cover.json"
}
```

Keys

Key	Type	Default	Description
anchor	anchor		-
angle	number | string		Turn the block, in degrees counter-clockwise, about the CENTRE of its "position" box. A CONTAINER in the flow may also carry this: it becomes a GROUP that turns about its own measured box and reserves the bounding box of the turned result, and its children inherit the transform. Any other flow block ignores it with a warning -- wrap it in a container. The block lays out upright inside its box and the finished result is turned, so a "position.height" that truncates content truncates the same text it would unrotated, and the turned result is NOT clipped to its box: it may extend past it, exactly as an unrotated positioned block may. Not to be confused with the image block's "rotate", which is an integer QUARTER TURN CLOCKWISE applied to the picture INSIDE its box (and re-sizes that box to the turned orientation). The two compose -- an image can stand a sideways scan upright with "rotate" and tilt the whole frame with "angle". Same meaning as the watermark's "angle". A STRING is a data token, so each record of a template turns its own sticker: "{{s.angle}}" takes the record's number (or text of one, such as a CSV field); a record without it leaves the block upright, silently, and a value that is not a number is reported with the block's pointer (and the -merge record) and drawn upright. A string with no token is read as it stands.
bleed	number		Step and Repeat: how far this placed block's box reaches past its TRIM on every side, in points -- the box IS the bleed box, and the trim is the box inset by this much. A design tool sizes a card by it (the trim plus twice the bleed) and draws the trim line inside each copy; the engine draws nothing from it, and keeps the bleed only through the block's own clip. Only a block with position has a trim: on any other block the key is refused with a notice naming the block.
blend	string		How this block's OWN ink combines with the page under it -- the sixteen PDF blend modes (ISO 32000-1 11.3.5), written as /BM. On a container that means its background, border and stripe, never its children; for the subtree as one object see "groupBlend". Case and separators are folded, so "colorDodge", "color-dodge" and "COLORDODGE" are one mode. An unrecognized name is refused with a warning and the block paints normally. See the blend-modes sample. One of: normal, multiply, screen, overlay, darken, lighten, colorDodge, colorBurn, hardLight, softLight, difference, exclusion, hue, saturation, ...
breakBefore	boolean		Pagination: open a new page before this block - a part divider, an appendix. Ignored when the page is already empty, so it never leaves a blank page behind.
changed	boolean		Revision / change bar: draw a vertical bar in the page margin beside this block's content to flag it as changed. A document-level changeBars object customizes the bar (width / color / side / distance); a condition tag declared with changeBar:true sets this automatically for tagged content.

Key	Type	Default	Description
`clip`	`boolean`		Clip this block's children (on a container) or its own ink to its box, with the style's corner "radius". Only bites where content is BIGGER than the box: a flow container grows to fit its children vertically, so give it "position" with a height, or a "minHeight", before expecting a vertical crop. Container-scoped, like "groupOpacity": on any other flow block it is refused with a warning. For a crop to a SHAPE rather than to the box, see "clipPath"; giving both INTERSECTS them.
`clipPath`	`object`		Crop this block to an arbitrary SHAPE stated inside its box -- a design tool's FRAME. Same vocabulary as the "shape" block and read by the same outline builder, so a shape and a clipPath given identical parameters describe the identical outline. Scoped exactly as "clip" is: a positioned block of any type (it needs a "position.height", because the shape is stated inside the box) or a container in the flow; anywhere else it is refused with a warning. Naming "clip" as well INTERSECTS the two. The shape turns, scales and skews with the block that carries it.
`condition`	`string`		Conditional text: render this block only when its condition-tag expression is shown under the document's conditions map. Whitespace/comma tag terms, each optionally '!'-negated, AND-combined; an unlisted tag defaults to shown. Empty means always shown.
`float`	`string`		Float this block out of the stacked flow to one side of its column; content that FOLLOWS it wraps into the space beside it. Works on ANY block type -- a table, a container, a chart, a code listing, a figure. "center" straddles the page content width, so text in every column wraps around it. The float box takes the block's own "width"; a block that declares none takes HALF the column, so a float always leaves room beside it. Ignored on a block that also sets "position" or "anchor" -- those have already chosen their placement. Not to be confused with "anchor": { "float": true }, a BOOLEAN about migrating an anchored frame to the next column that can hold it. One of: `left`, `right`, `center`.
`groupBlend`	`string`		Composite this block AND its children once, then blend the RESULT with the page. NOT the same key as "blend", and the difference is larger than it is for opacity: two stacked children that each multiply the backdrop multiply it TWICE, which is darker than the group multiplying once, and no per-child setting reproduces it. That is what a design tool's blend-mode control on a group means. Container-scoped: on any other flow block it is refused with a warning. Also the way to blend an element together with its drop shadow, which is drawn as stacked alpha layers and so keeps its own normal compositing. See the blend-modes sample. One of: `normal`, `multiply`, `screen`, `overlay`, `darken`, `lighten`, `colorDodge`, `colorBurn`, `hardLight`, `softLight`, `difference`, `exclusion`, `hue`, `saturation`, ...
`groupOpacity`	`number`		Composite this block AND its children once, then fade the result -- true GROUP opacity. NOT the same key as "opacity", which fades each element as it paints: two overlapping opaque children at opacity 0.5 each show a seam where they cross, and under groupOpacity 0.5 they do not. The two compose. Container-scoped: on any other flow block it is refused with a warning. See the clipped-groups sample.

Key	Type	Default	Description
`mirrorH`	`boolean | string`		Mirror the whole block left-to-right about the centre of its "position" box, under the same restriction as "angle". Identical to scale x -1, and the two compose. NOT "flipH": that key belongs to the image block, where it mirrors the PICTURE INSIDE its frame. The two are separate words so an image can do both. A STRING is a data token, so each record of a template mirrors its own sticker: "{{s.flip}}" takes the record's boolean, or text "true" / "false" in any case, or 1 / 0 (a number or text); a record without it draws the block unmirrored, silently, and any other value ("yes" included) is reported with the block's pointer (and the -merge record) and drawn unmirrored. A string with no token is read as it stands.
`mirrorV`	`boolean | string`		Mirror the whole block top-to-bottom about the centre of its "position" box, under the same restriction as "angle". Identical to scale y -1. NOT "flipV" -- see "mirrorH". A STRING is a data token, so each record of a template mirrors its own sticker: "{{s.flip}}" takes the record's boolean, or text "true" / "false" in any case, or 1 / 0 (a number or text); a record without it draws the block unmirrored, silently, and any other value ("yes" included) is reported with the block's pointer (and the -merge record) and drawn unmirrored. A string with no token is read as it stands.
`opacity`	`number`		Constant opacity for this block: 0 (invisible) to 1 (opaque, the default). REAL alpha -- a /CA /ca ExtGState in the PDF -- so a translucent panel over a photograph shows the photograph through it. Not to be confused with "style.background" {color, alpha}, which composites the tint toward WHITE at parse time and so only looks right over a white page. PER-ELEMENT, NOT GROUP: on a container it paints that container's own decoration -- background, border, accent bar, stripe -- and never its children, which draw at their own opacity (1 unless they set one). Two translucent shapes that overlap therefore composite twice, which is what per-element opacity means; group opacity, where a whole subtree is composited once and then faded, is a separate feature. Where a draw already carries its own alpha (an SVG fill-opacity, a gradient's alpha) the two MULTIPLY.
`repeatAt`	`array of object`		Imposition: draw this block ONCE and place the drawing again at each of these offsets, in points from the block's own position, y DOWNWARD as every `position` is. One entry = one extra copy, so a list of 7 puts 8 cards on a sheet. The drawing is captured into a Form XObject and each offset is a `cm` + `Do` of that one stream, so n copies cost one copy of the content. Translation only -- `angle` / `scale` / `skew` are not applied per placement. A REPEATED FORM MAY NOT CARRY TAGGED CONTENT: ISO 14289-1 7.20 allows a form's content ONE semantic parent, so repeating tagged text (or a shape with `alt`) fails PDF/UA -- veraPDF catches it, ap_addstruct's audit does not. The engine warns and draws it anyway. Repeat decoration: a shape or svg with no `alt`, or put the block in a master's `frames`, which is drawn as a page artifact. Only a block with `position` is drawn again: on a block without one (in the flow, or an anchored frame), or on a repeat / if / include block, the key is refused with a notice naming the block. Each copy takes the block's `groupOpacity` and `groupBlend`. The layout map (-map) reports the copies as `repeats` on the block's fragment, so a design tool can select and outline them.

Key	Type	Default	Description
`scale`	`number | array of number | object | string`		Scale the block about the same pivot as "angle" (the centre of its "position" box), and under the same restriction. A CONTAINER in the flow may also carry this: it becomes a GROUP that turns about its own measured box and reserves the bounding box of the turned result, and its children inherit the transform. Any other flow block ignores it with a warning -- wrap it in a container. A number scales both axes; [x, y] or {"x":, "y":} scales them separately. 1 = natural size. Applied AFTER layout, so a scaled block does NOT re-wrap its text at the scaled measure -- this is a design tool's scale handle, not a change of font size. A STRING is a data token, so each record of a template scales its own sticker: "{{s.scale}}" takes the record's number, [x, y] or {"x":, "y":} (members may be text of numbers), or text of one or two plain numbers such as a CSV field ("1.5", "1.5 0.8", "[1.5, 0.8]"); a record without it draws the block at its natural size, silently, and a value that is not a number or a pair is reported with the block's pointer (and the -merge record) and drawn at natural size. A string with no token is read as it stands.
`skew`	`number | array of number | object | string`		Shear the block, in DEGREES, about the same pivot as "angle", under the same restriction. [x, y] or {"x":, "y":}; a bare number skews x only. Positive x leans the top to the right and positive y lifts the right-hand side -- both counter-clockwise, matching "angle". Clamped to +/-89.9: a 90-degree skew is a singular matrix and draws nothing. A STRING is a data token, so each record of a template skews its own sticker: "{{s.skew}}" takes the record's number (x only), [x, y] or {"x":, "y":}, or text of one or two plain numbers; clamped as the written one is. A record without it draws the block unskewed, silently, and a value that is not a number or a pair is reported with the block's pointer (and the -merge record) and drawn unskewed. A string with no token is read as it stands.
`src`	`string`		**Required.** Path (relative to the spec) to a partial: a block, an array of blocks, or {blocks:[...]}.
`straddle`	`boolean`		Straddle across columns: when true and the body is multi-column, this block spans the FULL content width and text in every column wraps around it (above and below) - a spanning float generalized from center-floated images to any block. No effect in a single-column body, on a positioned / anchored block, or on an image/pdf already using float:"center" (the same mechanism). Does not balance the columns above it or act as a section divider.
`verticalAlign`	`string`		Where the content sits in its "position" box when it is shorter than the box's height: "top" (the default), "middle" ("center" is the same) or "bottom". A paragraph, heading or pre moves as a whole (its own box with it); a container keeps its box where it is and moves its children inside it (a child with its own "position" stays put). The offset is the box height less the content's height as it is actually placed -- the same lines, shrinkToFit size and padding that are then drawn -- halved for middle. Content as tall as the box or taller does not move: it is clipped and reported exactly as without the key. Needs a position with a height: on a block without one, on any other kind of block, on text set along a textPath "d" curve (stated in fractions of the whole box, so drawn across all of it; the "arc" form aligns), and on vertical text whose columns fill the box (columns run the full height unless a paragraph's or a heading's boxHeight is shorter), it is refused with a notice and ignored. The layout map keeps reporting the box. One of: `top`, `middle`, `bottom`, `center`.

Key	Type	Default	Description
`with`	`object`		Fields added/overridden in the included scope. A value that is a single "{{path}}" token resolves to the whole value at that path.
`z`	`number`		CANVAS MODE ONLY (see the top-level "pages"): where this element sits in the PAINT order of its page. Lower draws first, further back; unset means "wherever I am in the list", and ties keep author order. It moves the PAINT and not the READING: the structure tree keeps the document order, so a caption written first because it is read first can still be painted last because it sits on top. That separation is what a flattened scene graph cannot express. Ignored outside canvas mode, where the flow already fixes both orders.

The marker element

Type name: `marker`.

Plants an invisible named text value at a point in the flow, for running headers and footers to pick up with the `{{marker:name}}` building block - the dictionary-header idiom, where each page's header shows the last marker seen. Draws nothing itself.

Keys

Key	Type	Default	Description
`changed`	`boolean`		Revision / change bar: draw a vertical bar in the page margin beside this block's content to flag it as changed. A document-level changeBars object customizes the bar (width / color / side / distance); a condition tag declared with changeBar:true sets this automatically for tagged content.
`condition`	`string`		Conditional text: render this block only when its condition-tag expression is shown under the document's conditions map. Whitespace/comma tag terms, each optionally '!'-negated, AND-combined; an unlisted tag defaults to shown. Empty means always shown.
`name`	`string`		The marker class. A running band's {{marker:Name}} / {{firstMarker:Name}} / {{lastMarker:Name}} select on this name.
`text`	`string`		The value the running band prints for this marker.

14. Story threading

Text that flows between frames.

The frame element

Type name: `frame`.

A frame in a threaded story. Text poured into a story flows from frame to frame in the order you thread them, jumping pages if you like, with automatic "continued on page N" lines. The magazine and newsletter idiom.

Example

```
{
  "type": "frame",
  "story": "feature",
  "columns": 2,
  "columnGap": 18
}
```

Keys

Key	Type	Default	Description
`anchor`	`anchor`		-
`angle`	`number | string`		Turn the block, in degrees counter-clockwise, about the CENTRE of its "position" box. A CONTAINER in the flow may also carry this: it becomes a GROUP that turns about its own measured box and reserves the bounding box of the turned result, and its children inherit the transform. Any other flow block ignores it with a warning -- wrap it in a container. The block lays out upright inside its box and the finished result is turned, so a "position.height" that truncates content truncates the same text it would unrotated, and the turned result is NOT clipped to its box: it may extend past it, exactly as an unrotated positioned block may. Not to be confused with the image block's "rotate", which is an integer QUARTER TURN CLOCKWISE applied to the picture INSIDE its box (and re-sizes that box to the turned orientation). The two compose -- an image can stand a sideways scan upright with "rotate" and tilt the whole frame with "angle". Same meaning as the watermark's "angle". A STRING is a data token, so each record of a template turns its own sticker: "{{s.angle}}" takes the record's number (or text of one, such as a CSV field); a record without it leaves the block upright, silently, and a value that is not a number is reported with the block's pointer (and the -merge record) and drawn upright. A string with no token is read as it stands.
`bleed`	`number`		Step and Repeat: how far this placed block's box reaches past its TRIM on every side, in points -- the box IS the bleed box, and the trim is the box inset by this much. A design tool sizes a card by it (the trim plus twice the bleed) and draws the trim line inside each copy; the engine draws nothing from it, and keeps the bleed only through the block's own `clip`. Only a block with `position` has a trim: on any other block the key is refused with a notice naming the block.

Key	Type	Default	Description
blend	string		How this block's OWN ink combines with the page under it -- the sixteen PDF blend modes (ISO 32000-1 11.3.5), written as /BM. On a container that means its background, border and stripe, never its children; for the subtree as one object see "groupBlend". Case and separators are folded, so "colorDodge", "color-dodge" and "COLORDODGE" are one mode. An unrecognized name is refused with a warning and the block paints normally. See the blend-modes sample. One of: normal, multiply, screen, overlay, darken, lighten, colorDodge, colorBurn, hardLight, softLight, difference, exclusion, hue, saturation, ...
breakBefore	boolean		Pagination: open a new page before this block - a part divider, an appendix. Ignored when the page is already empty, so it never leaves a blank page behind.
changed	boolean		Revision / change bar: draw a vertical bar in the page margin beside this block's content to flag it as changed. A document-level changeBars object customizes the bar (width / color / side / distance); a condition tag declared with changeBar:true sets this automatically for tagged content.
clip	boolean		Clip this block's children (on a container) or its own ink to its box, with the style's corner "radius". Only bites where content is BIGGER than the box: a flow container grows to fit its children vertically, so give it "position" with a height, or a "minHeight", before expecting a vertical crop. Container-scoped, like "groupOpacity": on any other flow block it is refused with a warning. For a crop to a SHAPE rather than to the box, see "clipPath"; giving both INTERSECTS them.
clipPath	object		Crop this block to an arbitrary SHAPE stated inside its box -- a design tool's FRAME. Same vocabulary as the "shape" block and read by the same outline builder, so a shape and a clipPath given identical parameters describe the identical outline. Scoped exactly as "clip" is: a positioned block of any type (it needs a "position.height", because the shape is stated inside the box) or a container in the flow; anywhere else it is refused with a warning. Naming "clip" as well INTERSECTS the two. The shape turns, scales and skews with the block that carries it.
columnGap	number		Gutter between frame columns, in points.
columns	integer		Text columns inside the frame.
condition	string		Conditional text: render this block only when its condition-tag expression is shown under the document's conditions map. Whitespace/comma tag terms, each optionally '!'-negated, AND-combined; an unlisted tag defaults to shown. Empty means always shown.
continuation	boolean		Draw "(Continued from page N)" at the top (N = the previous frame in the thread).
continued	boolean		Draw "(Continued on page N)" at the bottom when the story overflows this frame.

Key	Type	Default	Description
`float`	`string`		Float this block out of the stacked flow to one side of its column; content that FOLLOWS it wraps into the space beside it. Works on ANY block type -- a table, a container, a chart, a code listing, a figure. "center" straddles the page content width, so text in every column wraps around it. The float box takes the block's own "width"; a block that declares none takes HALF the column, so a float always leaves room beside it. Ignored on a block that also sets "position" or "anchor" -- those have already chosen their placement. Not to be confused with "anchor": { "float": true }, a BOOLEAN about migrating an anchored frame to the next column that can hold it. One of: `left`, `right`, `center`.
`fromText`	`string`		Override the "Continued from" line; "{{page}}" is replaced by the page number.
`groupBlend`	`string`		Composite this block AND its children once, then blend the RESULT with the page. NOT the same key as "blend", and the difference is larger than it is for opacity: two stacked children that each multiply the backdrop multiply it TWICE, which is darker than the group multiplying once, and no per-child setting reproduces it. That is what a design tool's blend-mode control on a group means. Container-scoped: on any other flow block it is refused with a warning. Also the way to blend an element together with its drop shadow, which is drawn as stacked alpha layers and so keeps its own normal compositing. See the blend-modes sample. One of: `normal`, `multiply`, `screen`, `overlay`, `darken`, `lighten`, `colorDodge`, `colorBurn`, `hardLight`, `softLight`, `difference`, `exclusion`, `hue`, `saturation`, ...
`groupOpacity`	`number`		Composite this block AND its children once, then fade the result -- true GROUP opacity. NOT the same key as "opacity", which fades each element as it paints: two overlapping opaque children at opacity 0.5 each show a seam where they cross, and under groupOpacity 0.5 they do not. The two compose. Container-scoped: on any other flow block it is refused with a warning. See the clipped-groups sample.
`mirrorH`	`boolean | string`		Mirror the whole block left-to-right about the centre of its "position" box, under the same restriction as "angle". Identical to scale x -1, and the two compose. NOT "flipH": that key belongs to the image block, where it mirrors the PICTURE INSIDE its frame. The two are separate words so an image can do both. A STRING is a data token, so each record of a template mirrors its own sticker: "{{s.flip}}" takes the record's boolean, or text "true" / "false" in any case, or 1 / 0 (a number or text); a record without it draws the block unmirrored, silently, and any other value ("yes" included) is reported with the block's pointer (and the -merge record) and drawn unmirrored. A string with no token is read as it stands.
`mirrorV`	`boolean | string`		Mirror the whole block top-to-bottom about the centre of its "position" box, under the same restriction as "angle". Identical to scale y -1. NOT "flipV" -- see "mirrorH". A STRING is a data token, so each record of a template mirrors its own sticker: "{{s.flip}}" takes the record's boolean, or text "true" / "false" in any case, or 1 / 0 (a number or text); a record without it draws the block unmirrored, silently, and any other value ("yes" included) is reported with the block's pointer (and the -merge record) and drawn unmirrored. A string with no token is read as it stands.

Key	Type	Default	Description
`opacity`	`number`		Constant opacity for this block: 0 (invisible) to 1 (opaque, the default). REAL alpha -- a /CA /ca ExtGState in the PDF -- so a translucent panel over a photograph shows the photograph through it. Not to be confused with "style.background" {color, alpha}, which composites the tint toward WHITE at parse time and so only looks right over a white page. PER-ELEMENT, NOT GROUP: on a container it paints that container's own decoration -- background, border, accent bar, stripe -- and never its children, which draw at their own opacity (1 unless they set one). Two translucent shapes that overlap therefore composite twice, which is what per-element opacity means; group opacity, where a whole subtree is composited once and then faded, is a separate feature. Where a draw already carries its own alpha (an SVG fill-opacity, a gradient's alpha) the two MULTIPLY.
`repeatAt`	`array of object`		Imposition: draw this block ONCE and place the drawing again at each of these offsets, in points from the block's own position, y DOWNWARD as every `position` is. One entry = one extra copy, so a list of 7 puts 8 cards on a sheet. The drawing is captured into a Form XObject and each offset is a `cm` + `Do` of that one stream, so n copies cost one copy of the content. Translation only -- `angle` / `scale` / `skew` are not applied per placement. A REPEATED FORM MAY NOT CARRY TAGGED CONTENT: ISO 14289-1 7.20 allows a form's content ONE semantic parent, so repeating tagged text (or a shape with `alt`) fails PDF/UA -- veraPDF catches it, ap_addstruct's audit does not. The engine warns and draws it anyway. Repeat decoration: a shape or svg with no `alt`, or put the block in a master's `frames`, which is drawn as a page artifact. Only a block with `position` is drawn again: on a block without one (in the flow, or an anchored frame), or on a repeat / if / include block, the key is refused with a notice naming the block. Each copy takes the block's `groupOpacity` and `groupBlend`. The layout map (-map) reports the copies as `repeats` on the block's fragment, so a design tool can select and outline them.
`scale`	`number | array of number | object | string`		Scale the block about the same pivot as "angle" (the centre of its "position" box), and under the same restriction. A CONTAINER in the flow may also carry this: it becomes a GROUP that turns about its own measured box and reserves the bounding box of the turned result, and its children inherit the transform. Any other flow block ignores it with a warning -- wrap it in a container. A number scales both axes; [x, y] or {"x":, "y":} scales them separately. 1 = natural size. Applied AFTER layout, so a scaled block does NOT re-wrap its text at the scaled measure -- this is a design tool's scale handle, not a change of font size. A STRING is a data token, so each record of a template scales its own sticker: "{{s.scale}}" takes the record's number, [x, y] or {"x":, "y":} (members may be text of numbers), or text of one or two plain numbers such as a CSV field ("1.5", "1.5 0.8", "[1.5, 0.8]"); a record without it draws the block at its natural size, silently, and a value that is not a number or a pair is reported with the block's pointer (and the -merge record) and drawn at natural size. A string with no token is read as it stands.

Key	Type	Default	Description
skew	number | array of number | object | string		Shear the block, in DEGREES, about the same pivot as "angle", under the same restriction. [x, y] or {"x":, "y":}; a bare number skews x only. Positive x leans the top to the right and positive y lifts the right-hand side -- both counter-clockwise, matching "angle". Clamped to +/-89.9: a 90-degree skew is a singular matrix and draws nothing. A STRING is a data token, so each record of a template skews its own sticker: "{{s.skew}}" takes the record's number (x only), [x, y] or {"x":, "y":}, or text of one or two plain numbers; clamped as the written one is. A record without it draws the block unskewed, silently, and a value that is not a number or a pair is reported with the block's pointer (and the -merge record) and drawn unskewed. A string with no token is read as it stands.
story	string		Id of the story (top-level "stories") this frame pours its text from. Overflow threads into the next frame referencing the same story.
straddle	boolean		Straddle across columns: when true and the body is multi-column, this block spans the FULL content width and text in every column wraps around it (above and below) - a spanning float generalized from center-floated images to any block. No effect in a single-column body, on a positioned / anchored block, or on an image/pdf already using float:"center" (the same mechanism). Does not balance the columns above it or act as a section divider.
toText	string		Override the "Continued on" line; "{{page}}" is replaced by the page number.
verticalAlign	string		Where the content sits in its "position" box when it is shorter than the box's height: "top" (the default), "middle" ("center" is the same) or "bottom". A paragraph, heading or pre moves as a whole (its own box with it); a container keeps its box where it is and moves its children inside it (a child with its own "position" stays put). The offset is the box height less the content's height as it is actually placed -- the same lines, shrinkToFit size and padding that are then drawn -- halved for middle. Content as tall as the box or taller does not move: it is clipped and reported exactly as without the key. Needs a position with a height: on a block without one, on any other kind of block, on text set along a textPath "d" curve (stated in fractions of the whole box, so drawn across all of it; the "arc" form aligns), and on vertical text whose columns fill the box (columns run the full height unless a paragraph's or a heading's boxHeight is shorter), it is refused with a notice and ignored. The layout map keeps reporting the box. One of: top, middle, bottom, center.
z	number		CANVAS MODE ONLY (see the top-level "pages"): where this element sits in the PAINT order of its page. Lower draws first, further back; unset means "wherever I am in the list", and ties keep author order. It moves the PAINT and not the READING: the structure tree keeps the document order, so a caption written first because it is read first can still be painted last because it sits on top. That separation is what a flattened scene graph cannot express. Ignored outside canvas mode, where the flow already fixes both orders.

15. Meters

Small inline indicators.

The bar element

Type name: `bar`.

A small horizontal gauge - a rounded track with a proportional fill. Useful in dashboard tables and summary cards where a number wants a visual weight beside it.

Example

```
{
  "type": "bar",
  "value": 72,
  "max": 100,
  "width": "40%"
}
```

Keys

Key	Type	Default	Description
`anchor`	`anchor`		-
`angle`	`number | string`		Turn the block, in degrees counter-clockwise, about the CENTRE of its "position" box. A CONTAINER in the flow may also carry this: it becomes a GROUP that turns about its own measured box and reserves the bounding box of the turned result, and its children inherit the transform. Any other flow block ignores it with a warning -- wrap it in a container. The block lays out upright inside its box and the finished result is turned, so a "position.height" that truncates content truncates the same text it would unrotated, and the turned result is NOT clipped to its box: it may extend past it, exactly as an unrotated positioned block may. Not to be confused with the image block's "rotate", which is an integer QUARTER TURN CLOCKWISE applied to the picture INSIDE its box (and re-sizes that box to the turned orientation). The two compose -- an image can stand a sideways scan upright with "rotate" and tilt the whole frame with "angle". Same meaning as the watermark's "angle". A STRING is a data token, so each record of a template turns its own sticker: "{{s.angle}}" takes the record's number (or text of one, such as a CSV field); a record without it leaves the block upright, silently, and a value that is not a number is reported with the block's pointer (and the -merge record) and drawn upright. A string with no token is read as it stands.
`bleed`	`number`		Step and Repeat: how far this placed block's box reaches past its TRIM on every side, in points -- the box IS the bleed box, and the trim is the box inset by this much. A design tool sizes a card by it (the trim plus twice the bleed) and draws the trim line inside each copy; the engine draws nothing from it, and keeps the bleed only through the block's own `clip`. Only a block with `position` has a trim: on any other block the key is refused with a notice naming the block.

Key	Type	Default	Description
`blend`	`string`		How this block's OWN ink combines with the page under it -- the sixteen PDF blend modes (ISO 32000-1 11.3.5), written as /BM. On a container that means its background, border and stripe, never its children; for the subtree as one object see "groupBlend". Case and separators are folded, so "colorDodge", "color-dodge" and "COLORDODGE" are one mode. An unrecognized name is refused with a warning and the block paints normally. See the blend-modes sample. One of: `normal`, `multiply`, `screen`, `overlay`, `darken`, `lighten`, `colorDodge`, `colorBurn`, `hardLight`, `softLight`, `difference`, `exclusion`, `hue`, `saturation`, ...
`breakBefore`	`boolean`		Pagination: open a new page before this block - a part divider, an appendix. Ignored when the page is already empty, so it never leaves a blank page behind.
`changed`	`boolean`		Revision / change bar: draw a vertical bar in the page margin beside this block's content to flag it as changed. A document-level changeBars object customizes the bar (width / color / side / distance); a condition tag declared with changeBar:true sets this automatically for tagged content.
`clip`	`boolean`		Clip this block's children (on a container) or its own ink to its box, with the style's corner "radius". Only bites where content is BIGGER than the box: a flow container grows to fit its children vertically, so give it "position" with a height, or a "minHeight", before expecting a vertical crop. Container-scoped, like "groupOpacity": on any other flow block it is refused with a warning. For a crop to a SHAPE rather than to the box, see "clipPath"; giving both INTERSECTS them.
`clipPath`	`object`		Crop this block to an arbitrary SHAPE stated inside its box -- a design tool's FRAME. Same vocabulary as the "shape" block and read by the same outline builder, so a shape and a clipPath given identical parameters describe the identical outline. Scoped exactly as "clip" is: a positioned block of any type (it needs a "position.height", because the shape is stated inside the box) or a container in the flow; anywhere else it is refused with a warning. Naming "clip" as well INTERSECTS the two. The shape turns, scales and skews with the block that carries it.
`condition`	`string`		Conditional text: render this block only when its condition-tag expression is shown under the document's conditions map. Whitespace/comma tag terms, each optionally '!'-negated, AND-combined; an unlisted tag defaults to shown. Empty means always shown.
`fill`	`color | object`		Color of the filled portion of the bar.

Key	Type	Default	Description
`float`	`string`		Float this block out of the stacked flow to one side of its column; content that FOLLOWS it wraps into the space beside it. Works on ANY block type -- a table, a container, a chart, a code listing, a figure. "center" straddles the page content width, so text in every column wraps around it. The float box takes the block's own "width"; a block that declares none takes HALF the column, so a float always leaves room beside it. Ignored on a block that also sets "position" or "anchor" -- those have already chosen their placement. Not to be confused with "anchor": { "float": true }, a BOOLEAN about migrating an anchored frame to the next column that can hold it. One of: `left`, `right`, `center`.
`groupBlend`	`string`		Composite this block AND its children once, then blend the RESULT with the page. NOT the same key as "blend", and the difference is larger than it is for opacity: two stacked children that each multiply the backdrop multiply it TWICE, which is darker than the group multiplying once, and no per-child setting reproduces it. That is what a design tool's blend-mode control on a group means. Container-scoped: on any other flow block it is refused with a warning. Also the way to blend an element together with its drop shadow, which is drawn as stacked alpha layers and so keeps its own normal compositing. See the blend-modes sample. One of: `normal`, `multiply`, `screen`, `overlay`, `darken`, `lighten`, `colorDodge`, `colorBurn`, `hardLight`, `softLight`, `difference`, `exclusion`, `hue`, `saturation`, ...
`groupOpacity`	`number`		Composite this block AND its children once, then fade the result -- true GROUP opacity. NOT the same key as "opacity", which fades each element as it paints: two overlapping opaque children at opacity 0.5 each show a seam where they cross, and under groupOpacity 0.5 they do not. The two compose. Container-scoped: on any other flow block it is refused with a warning. See the clipped-groups sample.
`height`	`number`		Thickness of the bar, in points.
`max`	`number`		Value that corresponds to a completely full bar. Default 1, so a 0-1 value reads as a fraction.
`maxValue`	`number`		Alias for "max".
`mirrorH`	`boolean | string`		Mirror the whole block left-to-right about the centre of its "position" box, under the same restriction as "angle". Identical to scale x -1, and the two compose. NOT "flipH": that key belongs to the image block, where it mirrors the PICTURE INSIDE its frame. The two are separate words so an image can do both. A STRING is a data token, so each record of a template mirrors its own sticker: "{{s.flip}}" takes the record's boolean, or text "true" / "false" in any case, or 1 / 0 (a number or text); a record without it draws the block unmirrored, silently, and any other value ("yes" included) is reported with the block's pointer (and the -merge record) and drawn unmirrored. A string with no token is read as it stands.

Key	Type	Default	Description
`mirrorV`	`boolean | string`		Mirror the whole block top-to-bottom about the centre of its "position" box, under the same restriction as "angle". Identical to scale y -1. NOT "flipV" -- see "mirrorH". A STRING is a data token, so each record of a template mirrors its own sticker: "{{s.flip}}" takes the record's boolean, or text "true" / "false" in any case, or 1 / 0 (a number or text); a record without it draws the block unmirrored, silently, and any other value ("yes" included) is reported with the block's pointer (and the -merge record) and drawn unmirrored. A string with no token is read as it stands.
`opacity`	`number`		Constant opacity for this block: 0 (invisible) to 1 (opaque, the default). REAL alpha -- a /CA /ca ExtGState in the PDF -- so a translucent panel over a photograph shows the photograph through it. Not to be confused with "style.background" {color, alpha}, which composites the tint toward WHITE at parse time and so only looks right over a white page. PER-ELEMENT, NOT GROUP: on a container it paints that container's own decoration -- background, border, accent bar, stripe -- and never its children, which draw at their own opacity (1 unless they set one). Two translucent shapes that overlap therefore composite twice, which is what per-element opacity means; group opacity, where a whole subtree is composited once and then faded, is a separate feature. Where a draw already carries its own alpha (an SVG fill-opacity, a gradient's alpha) the two MULTIPLY.
`radius`	`number`		Corner radius of the bar and its track, in points. Use half the height for a fully rounded capsule.
`repeatAt`	`array of object`		Imposition: draw this block ONCE and place the drawing again at each of these offsets, in points from the block's own position, y DOWNWARD as every `position` is. One entry = one extra copy, so a list of 7 puts 8 cards on a sheet. The drawing is captured into a Form XObject and each offset is a `cm` + `Do` of that one stream, so n copies cost one copy of the content. Translation only -- `angle` / `scale` / `skew` are not applied per placement. A REPEATED FORM MAY NOT CARRY TAGGED CONTENT: ISO 14289-1 7.20 allows a form's content ONE semantic parent, so repeating tagged text (or a shape with `alt`) fails PDF/UA -- veraPDF catches it, ap_addstruct's audit does not. The engine warns and draws it anyway. Repeat decoration: a shape or svg with no `alt`, or put the block in a master's `frames`, which is drawn as a page artifact. Only a block with `position` is drawn again: on a block without one (in the flow, or an anchored frame), or on a repeat / if / include block, the key is refused with a notice naming the block. Each copy takes the block's `groupOpacity` and `groupBlend`. The layout map (-map) reports the copies as `repeats` on the block's fragment, so a design tool can select and outline them.

Key	Type	Default	Description
scale	number | array of number | object | string		Scale the block about the same pivot as "angle" (the centre of its "position" box), and under the same restriction. A CONTAINER in the flow may also carry this: it becomes a GROUP that turns about its own measured box and reserves the bounding box of the turned result, and its children inherit the transform. Any other flow block ignores it with a warning -- wrap it in a container. A number scales both axes; [x, y] or {"x":, "y":} scales them separately. 1 = natural size. Applied AFTER layout, so a scaled block does NOT re-wrap its text at the scaled measure -- this is a design tool's scale handle, not a change of font size. A STRING is a data token, so each record of a template scales its own sticker: "{{s.scale}}" takes the record's number, [x, y] or {"x":, "y":} (members may be text of numbers), or text of one or two plain numbers such as a CSV field ("1.5", "1.5 0.8", "[1.5, 0.8]"); a record without it draws the block at its natural size, silently, and a value that is not a number or a pair is reported with the block's pointer (and the -merge record) and drawn at natural size. A string with no token is read as it stands.
skew	number | array of number | object | string		Shear the block, in DEGREES, about the same pivot as "angle", under the same restriction. [x, y] or {"x":, "y":}; a bare number skews x only. Positive x leans the top to the right and positive y lifts the right-hand side -- both counter-clockwise, matching "angle". Clamped to +/-89.9: a 90-degree skew is a singular matrix and draws nothing. A STRING is a data token, so each record of a template skews its own sticker: "{{s.skew}}" takes the record's number (x only), [x, y] or {"x":, "y":}, or text of one or two plain numbers; clamped as the written one is. A record without it draws the block unskewed, silently, and a value that is not a number or a pair is reported with the block's pointer (and the -merge record) and drawn unskewed. A string with no token is read as it stands.
straddle	boolean		Straddle across columns: when true and the body is multi-column, this block spans the FULL content width and text in every column wraps around it (above and below) - a spanning float generalized from center-floated images to any block. No effect in a single-column body, on a positioned / anchored block, or on an image/pdf already using float:"center" (the same mechanism). Does not balance the columns above it or act as a section divider.
track	color		Color of the unfilled groove behind the bar.
value	number		The value to show, measured against "max".

Key	Type	Default	Description
`verticalAlign`	`string`		Where the content sits in its "position" box when it is shorter than the box's height: "top" (the default), "middle" ("center" is the same) or "bottom". A paragraph, heading or pre moves as a whole (its own box with it); a container keeps its box where it is and moves its children inside it (a child with its own "position" stays put). The offset is the box height less the content's height as it is actually placed -- the same lines, shrinkToFit size and padding that are then drawn -- halved for middle. Content as tall as the box or taller does not move: it is clipped and reported exactly as without the key. Needs a position with a height: on a block without one, on any other kind of block, on text set along a textPath "d" curve (stated in fractions of the whole box, so drawn across all of it; the "arc" form aligns), and on vertical text whose columns fill the box (columns run the full height unless a paragraph's or a heading's boxHeight is shorter), it is refused with a notice and ignored. The layout map keeps reporting the box. One of: `top`, `middle`, `bottom`, `center`.
`width`	`number | string`		Drawn track width - points, a percentage of the column, or "auto"/absent for the full content width. The colored fill inside the track is value/max of this.
`z`	`number`		CANVAS MODE ONLY (see the top-level "pages"): where this element sits in the PAINT order of its page. Lower draws first, further back; unset means "wherever I am in the list", and ties keep author order. It moves the PAINT and not the READING: the structure tree keeps the document order, so a caption written first because it is read first can still be painted last because it sits on top. That separation is what a flattened scene graph cannot express. Ignored outside canvas mode, where the flow already fixes both orders.

Part IV — Shared vocabularies

16. Shared definitions

These are the building blocks referred to from many elements. Where an element's key table gives a type such as `style` or `color`, this is where that type is defined.

Style

Schema definition: `style`. Type and paragraph properties every block accepts.
A style: text properties (face, size, color, tracking) cascade to nested content; box properties (padding, border, background, radius, spacing) apply only to the block that sets them.

A justified body style

```
{
  "fontFamily": "Helvetica",
  "fontSize": 11,
  "color": "#222222",
  "align": "justify",
  "lineHeight": 1.4,
  "firstLineIndent": 24,
  "spaceAfter": 6
}
```

Decimal tab stops with a dot leader

```
{
  "tabs": [
    {
      "at": 60,
      "align": "left"
    },
    {
      "at": 470,
      "align": "decimal",
      "leader": "dot"
    }
  ]
}
```

Key	Type	Default	Description
`accentBar`	`border`		A solid colored bar drawn on the LEFT edge of a box (a textbook call-out's spine). Uses the border width as the bar thickness. Painted by container, frame, table cell and, since the box became a text-block property, paragraph and heading; a list or pre still ignores it.
`align`	`string`		Horizontal alignment of the text within its column. One of: `left`, `center`, `right`, `justify`, `start`, `end`.

Key	Type	Default	Description
`background`	`color | object`		A solid fill color, or { color, alpha } composited over white to a solid tint.
`basedOn`	`string`		Start from this named "styles" entry; the object's own keys override it.
`blockAlign`	`string`		Set the paragraph's lines as ONE BLOCK: every line keeps its own "align", but against a box as wide as the paragraph's longest line, and that box sits at the left, centre or right of the content width -- so an address typed flush left stays flush left line to line while the group sits centred in its box. The line breaks never change (the block is measured from the lines already broken), a line's indent counts toward its width, and each line moves by (content width - longest line) x (block fraction - align fraction), with fractions 0 / 0.5 / 1 for left / center / right: so centre-aligned text under "center", and a justified paragraph (whose lines fill the width), do not move. Absent, each line aligns across the full width as always -- for left-aligned text the same as "left", which is a real request on right-aligned text (it pulls the block to the left edge). Applied at the size drawn (after shrinkToFit), with keepIndent, tabs (a line's tab stops move with it) and in columns, on paragraphs, headings and pre blocks, in the flow or in a positioned box. Not applied where text flows around floats, through threaded frames, beside a run-in heading or a drop cap: those lines align across the width as before. Vertical text and text on a textPath curve have no lines to set as a block and are left as they are. A BLOCK property: read from the block being laid out, so setting it on a RUN changes nothing. Inherited by nested blocks. One of: `left`, `center`, `right`.
`bold`	`boolean`		Shorthand for weight: "bold".
`border`	`border`		Rule drawn around the block's box. A box property - not inherited.
`color`	`color`		Text color. Inherited by nested content. The special value "none" fills nothing, leaving hollow letterforms that show only the "strokeColor" outline - with no outline to fall back on it is ignored, so text is never silently made invisible.

Key	Type	Default	Description
direction	string		Which way this block's text RUNS, for bidirectional scripts (Hebrew, Arabic). "auto" (the default) applies the Unicode bidirectional algorithm's own rule - the first strong character in the paragraph decides - so a Hebrew paragraph inside an English report resolves right-to-left on its own and an English one inside a Hebrew report resolves left-to-right, with nothing declared. Set it explicitly only to override that: a paragraph of digits or punctuation has no strong character to read, and an interface label may need to run right-to-left regardless of its content. Numbers inside right-to-left text still run left-to-right, and brackets take the direction of what they enclose, because the algorithm says so. NOT the same key as pageSetup.binding, which is the edge the BOOK is bound on: a Hebrew book wants both, an English book quoting one Hebrew sentence wants neither. WHAT IT MOVES: text alignment (align "start"/"end"), list bullets and numbers, table columns (the first column sits at the right edge, spans and rules follow), footnote and endnote markers, bibliography labels, and the titles of those blocks. Direction is INHERITED, so setting it on a container or on the document carries to everything inside - including a table cell of nothing but digits, which has no strong character of its own to detect from. WHAT IT DOES NOT MOVE, deliberately: `float`, `imageAlign`, `tableAlign` and a pdfContactSheet grid are PHYSICAL placements - left means the left of the page, the way a printer means it - and the left/center/right slots of a running header or footer are named for the page edges they sit at (pageSetup mirroring already swaps those on verso pages). A `pre` block stays left-aligned because code is read left-to-right whatever surrounds it. Table of contents leaders and index columns do not mirror YET. One of: `auto`, `ltr`, `rtl`, `l2r`, `r2l`, `left-to-right`, `right-to-left`.
features	array of string		OpenType features to set this text with, on an EMBEDDED face that defines them: `liga` standard ligatures (fi, ffi), `dlig` discretionary ligatures (ct, st), `onum` oldstyle figures / `lnum` lining figures, `tnum` tabular figures / `pnum` proportional figures, `smcp` true small capitals. Inherited, and the list REPLACES the inherited one rather than merging with it; an EMPTY list means none, which is how a table turns figures back to lining inside a document that set them oldstyle. Applied in a fixed order (smcp, liga, dlig, then the figure sets) rather than the order listed, because each lookup sees the glyphs the last one left. A feature the face does not define substitutes nothing and is not an error. Measured and drawn from the same glyphs, so line breaking accounts for the substituted widths. Base-14 faces carry no GSUB and are unaffected. A substituted glyph is placed by glyph id, and the engine writes the glyph-to-text map itself rather than leaving the PDF library to derive it from glyph names - so the characters survive for every feature listed here, including `pnum`, whose `one.fitted` figures used to come back as Private Use characters and could not be copied at all.
firstLineIndent	number		Points the paragraph's FIRST line is indented; negative = hanging (first line pulled back left of the rest). Not inherited.
fitWidth	boolean		Heading: solve the font size so the (single-line) text fills the column.
fontFamily	string		Font family name, resolved and embedded by name. Inherited by nested content.

Key	Type	Default	Description
fontSize	number		Type size in points. Inherited by nested content.
italic	boolean		Use the italic face of the family. Inherited by nested content.
keepIndent	boolean		Keep the spaces a line starts with -- at the start of the text and after each line break -- as that line's indent, the width of that many of the paragraph's spaces (wordSpacing included), instead of collapsing them. Only leading spaces: every other run of spaces still collapses to one, a line of nothing but spaces stays empty, and tabs keep their own meaning (they align to "tabs" stops when those are set). The indent is part of the line, like firstLineIndent, so justification never stretches it and it is never underlined; it gives way rather than push the line's first word out of the column, and tab stops stay measured from the paragraph's edge. Like firstLineIndent, it is applied at the left edge in right-to-left text too. Not applied where text flows around floats, through threaded frames, beside a run-in heading or a drop cap. A BLOCK property: it is read from the block being laid out, so setting it on a RUN changes nothing. Inherited by nested blocks.
letterSpacing	number		Extra points between glyphs (inherited).
lineHeight	number		Multiple of font size (1.0 = single).
minFontSize	number	6	shrinkToFit's floor, in points: shrinking never sets any run of the text below this, and text written smaller keeps its own size. Not inherited.
minHeight	number		The box grows to at least this height in points (uniform tiles in a grid, a deep chapter banner on a heading). Honored by container, frame, table cell, paragraph and heading.
overline	boolean		Draw a rule above the text, clear of the ascender (in the text color, same weight as the underline). Inherited; consecutive decorated words bridge into one continuous rule.
padding	insets		Padding between the block's box edge and its content. A box property - not inherited. Honored by container, frame, table cell, paragraph and heading; on a paragraph or a heading it insets the text on all four sides, so a negative firstLineIndent hangs against it. A list or pre ignores it.
radius	number		Rounded-corner radius in points for a container / card box (fill, border, and edge stripe follow it).
shadow	shadow		-
shrinkToFit	boolean		Paragraph or heading with a position height: shrink the text until its lines fit that box, in 0.25 pt steps of its largest size -- every size scaled by one ratio, but no run set below minFontSize (a run written smaller keeps its own size). Only a positioned block with a height, set in horizontal lines, is shrunk -- not a dropCap or textPath paragraph, one a run-in heading is drawn into, or a side head; anywhere else the key does nothing. Text that still does not fit at minFontSize, or that its padding leaves no room for, is clipped and reported. Not inherited.

Key	Type	Default	Description
smallCaps	boolean		Render lowercase letters as smaller uppercase (two-tier small caps): capitals/digits stay full size, lowercase becomes uppercase at about 0.75x the size. Synthetic from the regular font; the original text is preserved via /ActualText so search and assistive tech still read the real words. Inherited. When the same style ALSO lists smcp in features and the embedded face actually defines that feature, the drawn small capitals win and this synthesis is skipped - never both, because the synthetic pass uppercases the text and smcp maps from the lowercase, so running both would leave the feature nothing to substitute. A face without smcp keeps these synthetic capitals.
spaceAfter	number		Blank space below the block, in points. A box property: it applies to this block only and is NOT inherited by nested content.
spaceBefore	number		Blank space above the block, in points. A box property: it applies to this block only and is NOT inherited by nested content.
stretch	number	100	Horizontal glyph scale as a PERCENT: 100 is normal, below 100 condenses, above 100 extends (FrameMaker's FP_Stretch, the PDF Tz operator). Scales the letterforms and their advances but NOT the font size, so the baseline and line height are unchanged and line breaking accounts for the new widths. Inherited.
strike	boolean		Strike through the text.
strikethrough	boolean		Alias for "strike".
stripe	object		A colored bar along one edge of a box (generalizes accentBar; rounds only the two corners on its own edge). Painted by container, frame, table cell and, since the box became a text-block property, paragraph and heading; a list or pre still ignores it.
strokeColor	color		Color of the glyph OUTLINE, drawn in addition to (or instead of) the fill. Inherited by nested content, so a run inside a stroked paragraph keeps the outline unless it overrides it. Needs "strokeWidth" above 0 to show; on its own it does nothing.
strokeWidth	number		Thickness of the glyph outline, in POINTS. Needs "strokeColor" to show. A PDF stroke straddles the outline - half inside the letterform, half outside - so stroking always thickens the type somewhat; a value tuned for display sizes is far too heavy at body sizes. Set 0 to drop an outline inherited from a parent.
tabs	array of object		Paragraph tab stops. A Tab (\t) in the text advances to the next stop and fills the gap with its leader (Name ..... 42).
textEffect	textEffect		-
textFill	textFill		-
tracking	number		Alias for letterSpacing.

Key	Type	Default	Description
underline	boolean | string		Underline the text, drawn in the text color. true is a single rule and false is none - the spelling every existing document uses, and it keeps working. The named values add FrameMaker's other two: "double" draws a second rule below the first; "numeric" is carried distinctly but DRAWS AS SINGLE today - the FDK defines the constant (FV_CB_NUMERIC_UNDERLINE) and neither of its manuals says what it looks like, so the value is preserved rather than guessed at. One of: none, single, double, numeric.
weight	string		Font weight - "bold" for the bold face of the family. Inherited by nested content. One of: normal, regular, bold.
wordSpacing	number		Extra points between words. A BLOCK property: it is read from the block being laid out, so setting it on a RUN parses and resolves but changes nothing. Inherited by nested blocks.

Inline runs

Schema definition: runObject. The pieces a paragraph or heading is built from.

An inline run object: text + inline properties (style, link, lang, footnote, or an inline form field).

A styled run

```
{
  "text": "18 N·m",
  "style": {
    "weight": "bold"
  }
}
```

A run that links out

```
{
  "text": "the engine",
  "link": "https://appligent.com"
}
```

A run with a footnote

```
{
  "text": "apogee",
  "footnote": "The highest point of the flight."
}
```

A run with an endnote (gathered at the next `endnotes` block)

```
{
  "text": "apogee",
  "endnote": "The highest point of the flight."
}
```

A superscript run (ordinals, ® / ™, exponents)

```
{
  "text": "nd",
  "superscript": true
}
```

A cross-reference to a block id

```
{
  "ref": "assembly"
}
```

Key	Type	Default	Description
`alt`	`string`		What a screen reader is given in place of the graphic or equation this run draws: the /Alt of the inline <Figure>, or of the inline <Formula> (there an alias for "mathAlt"). Falls back to the run's own text. REQUIRED on a graphic, which has no other accessible text - without it the engine warns and tags the placeholder /Alt "graphic".
`as`	`string`	page	What a "ref" run renders: "page" the page number, "number" the bare number ("3-2"), "label" the label word + number ("Figure 3-2"), "title" the target's heading / caption text without its number, "full" the target named and located ("Figure 3-2 on page 47", or "Centre of Pressure on page 47" where nothing is numbered). "number" and "label" need a numbering scheme and degrade to the page label without one; "title" and "full" do not, and only a target with neither a number nor a title falls back to the page. Ignored without "ref". One of: `page`, `number`, `label`, `title`, `full`.
`badge`	`string`		An inline badge / pill: this text is drawn inside a rounded, padded, filled label. `style` colors the label text; `badgeStyle` carries the box.
`badgeStyle`	`object`		The badge box: fill / radius / padding (radius >= 999 renders a full pill).
`baselineShift`	`number`		Points to move this run off its baseline: positive raises it, negative lowers it. It is added to the superscript or subscript position, so a small superscript ™ can be lifted to the cap height or a subscript set lower; on a plain run it moves that run alone. Only the ink moves: the line keeps its height, so a large shift can reach the line above or below.

Key	Type	Default	Description
cite	string		Cite one or more works from the document's bibliography block, by key. Several keys are separated by a comma or a space ("knuth1984, lamport1986"). The run's own text is IGNORED - the engine composes the marker from the bibliography's style ([3] or (Knuth 1984)), numbering entries by order of first citation, and links the marker forward to its entry (the entry's own label links back to the first citation of it). In a run-form caption too, numbered where the caption stands in reading order -- when the caption is drawn.
citePages	string		A locator inside the cited work, drawn inside the marker: "pp. 12-14" gives [3, pp. 12-14]. Ignored without cite.
condition	string		Conditional text: render this run only when its condition-tag expression is shown under the document's conditions map. Same grammar as a block condition.
content	string		Inline SVG source, drawn baseline-aligned in place of the text. Alias for "svg".
endnote	string		Endnote text: a superscript marker here + a note gathered into the next endnotes block (chapter / document end). The marker links to its note and back (to the first place the marker is drawn, never to a contents row's copy of it). Independent of the footnote stream. In a run-form caption too, numbered where the caption stands in reading order (a table's caption before its rows) -- when the caption is drawn: one with no text is not, and its endnote takes no number (the engine says so).
footnote	string		Footnote text: a superscript marker here + an auto-numbered note at the page bottom. Not drawn in a caption: a caption reserves no room at the page foot, so the note is dropped, takes no number from the footnotes after it, and the engine says so, naming the block (use an endnote there). Not drawn in vertical (直排) text either -- a column has no page foot along it -- with the same result: no number, and the engine says so. Drawn on every other form of heading, run-in and curved included.
height	number		Inline box height in points: a field widget (with "field"), or a graphic (with "image" / "svg"). Absent on a graphic, the height is the run's own font size - an unsized icon is one em tall, so it sits on the line like a letter.
image	string		An inline GRAPHIC: an image file drawn baseline-aligned in place of the text, tagged <Figure> carrying "alt". The inline analogue of the "image" block - an icon, a logo, a signature, a symbol no font carries. Sized by "width" / "height"; unsized it is one em tall and takes its width from the image's own aspect, so it sits on the line like a letter and follows the text when the style changes.
label	string		Inline field tooltip fallback (with "field").
lang	string		Run language -> a <Span> with /Lang in tagged output, and the language whose rules divide this run's words when hyphenate is on. A run beats its block, which beats the document.
link	string		URL (external) or #anchor (internal) link target.
mathAlt	string		An inline equation: a spoken reading (the inline <Formula> /Alt).

Key	Type	Default	Description
mathML	string		Alias for "mathml".
mathSvg	string		An inline equation: its pre-rendered SVG (MathJax, fontCache:'none'), drawn baseline-aligned in place of text.
mathml	string		An inline equation: its Presentation MathML source, embedded as a /Supplement associated file on the inline <Formula>. Pair with "mathSvg".
ref	string		Cross-reference: renders a live property of the block whose "id" matches, as an internal link. The run's own text is ignored; "as" picks which property (default the page number).
refTemplate	string		Custom "ref" text over the placeholders {number} {word} {label} {title} {page} - e.g. "see {label} ({title}) on p.{page}". Overrides "as". Ignored without "ref".
sidenote	string		Margin-note text drawn in the reserved side band, top-aligned with the line this run STARTS on. Unlike a footnote or endnote it leaves NO marker in the text - a margin note refers to its text by sitting beside it. Requires the band: set pageSetup.sidenoteWidth (alias sideHeadWidth) and a single-column body. Notes stack in the band rather than overlapping, sharing the cursor with side heads. Tagged as a Note element. Not drawn in a caption, which has no margin band, nor beside vertical (直排) text, which has none along its columns: the engine says so, naming the block.
style	style		Style for this run only. Inherited text properties still apply underneath it. TEXT properties only: a box property written here (background, padding, border, radius, stripe, shadow, minHeight, align, lineHeight, firstLineIndent, tabs, spaceBefore/After, fitWidth) is accepted and ignored, because a run is not a box -- to tint words behind the type use "textEffect": "background".
subscript	boolean		Drop this run to subscript position - for chemical formulae (H2O) and mathematical indices. The run is DROPPED but not resized; set a smaller "style.fontSize" for the conventional look. Ignored when "superscript" is also set.
superscript	boolean		Raise this run to superscript position - for ordinals (1st), trademark marks (® / ™) and exponents. The run is RAISED but not resized; set a smaller "style.fontSize" for the conventional look.
svg	string		Inline SVG source, drawn baseline-aligned in place of the text. Beside "mathml" or "mathSvg" this is the EQUATION's SVG (tagged <Formula>, with the MathML as a /Supplement associated file). On its own it is a GRAPHIC (tagged <Figure>, described by "alt"), sized by "width" / "height" or, unsized, one em tall with its width from the viewBox aspect.
text	string		The run's text - the characters this run contributes to the line.
width	number		Inline box width in points: a field widget (with "field"), or a graphic (with "image" / "svg"). Absent on a graphic, the width is derived from the height and the graphic's intrinsic aspect. Giving BOTH width and height sets the box: an image is stretched to it; an svg keeps its viewBox's shape inside it, as its root preserveAspectRatio says (default xMidYMid meet: fitted whole and centred), and is stretched only when that says "none".

Color

Schema definition: `color`. How a color is written.

A color in one of four color spaces. The PDF backend emits the true operator for each (rg / g / k / scn).

Hex, the everyday form

```
#1f4e79
```

DeviceCMYK for print

```
{
  "cmyk": [
    0.75,
    0.0,
    0.62,
    0.06
  ]
}
```

A spot ink with a tint and a process alternate

```
{
  "spot": "PANTONE 185 C",
  "tint": 0.6,
  "cmyk": [
    0,
    0.91,
    0.76,
    0
  ]
}
```

Key	Type	Default	Description
`cmyk`	`array of number`		The alternate -- what the ink approximates at full tint.
`color`	`string`		RGB alternate when no "cmyk" given (display only).
`gray`	`number`		Single gray component: 0 is black, 1 is white. A value above 1 is read as a 0-100 percentage.
`rgb`	`array of number`		[red, green, blue]. Components run 0-1, but if any value exceeds 1 the whole array is read as 0-100 percentages.
`spot`	`string`		The colorant name written into the PDF (e.g. "PANTONE 185 C").
`tint`	`number`		Ink strength 0..1 (or 0..100 percent). Default 1.

Dimensions

Schema definition: `dimension`. How a length is written.
A length: points (number), "NN%" of the available width, "auto", or "NN".

A percentage of the column

```
50%
```

Absolute points

```
180
```

Inches

```
3in
```

Absolute position

Schema definition: `position`. Taking a block out of the flow.
Absolute placement: fixed coordinates (points, top-left origin), measured from the trim page's top-left -- or, when the nearest enclosing container that writes "childOrigin" says "container", from that container's border box's top-left, wherever it lands. Either way everything that reports geometry (the layout map, notices) reports PAGE coordinates. The block leaves the flow and consumes no flow height; content past an explicit height is not drawn, and the engine says so -- an "aplayout:" notice naming the block, and "overset" on its layout-map entry. A paragraph or heading can set "shrinkToFit" to shrink its text into the box instead. Pairs with pageSetup.template for form overlays onto existing PDFs. Bindable from data, so each record of a template places its own sticker: any of x, y, width, height may be a "{{path}}" string, and so may the whole position ("{{s.box}}", a record's {x, y, width, height}). A value that is exactly one token takes the whole value at that path -- a number, or text of one such as a CSV field (points; there are no units). A record without a width or height sizes the box to its content, silently, as an unwritten one does; a record without a usable x or y is reported with the block's pointer (and the -merge record) and the block is NOT drawn, because there is nowhere it was meant to go. Text that is not a number, and a negative width or height, are reported. A string with no token is read as it stands, in a document with no data. The whole position from data: "{{s.box}}" resolves to the record's {x, y, width, height} object ("w" / "h" accepted), each member a number or text of one.

Example

```
{
  "x": 320,
  "y": 60,
  "width": 180,
  "height": 90
}
```

Key	Type	Default	Description
h	number | string		Alias for "height", used only when "height" is absent.
height	number | string		Height of the block's box, in points. Optional - most blocks size to their content; vertical text with no boxHeight, which has no size of its own, takes the room from where it begins down to the flow's foot, above any footnote band (the page's foot, where it begins in the bottom margin or within a character of the flow's), and at most the text area's full height - in a positioned paragraph or heading, or in the flow of a container with no height, where what precedes the text is drawn as before and what follows it is not drawn, and said; in a cell of a positioned columns block, table or cardGrid, every such text shares the room: each is given the same box, the tallest with which the whole block ends at that foot (or, where the block's horizontal content runs past the foot anyway, with which the text does), every row still drawn (since 2026-10-15). With a height, a table drops its rows past it (said) and a columns block's or cardGrid's vertical text with no boxHeight stands a text area tall past it; and in a vertical section positioned vertical text is not bounded - give it a boxHeight. A STRING is a data token (or text holding one, or a number as text), resolved per record: a record without it sizes the box to its content.
w	number | string		Alias for "width", used only when "width" is absent.
width	number | string		Width of the block's box, in points. A positioned block is taken out of the normal flow and reserves no vertical space unless it also sets "textWrap". A STRING is a data token (or text holding one, or a number as text), resolved per record: a record without it sizes the box as if no width were written.
x	number | string		**Required.** Distance from the origin's left edge to the block's left edge, in points: the trim page's left edge, or -- when the nearest enclosing container that writes "childOrigin" says "container" -- that container's box's left edge. A STRING is a data token (or text holding one, or a number as text), resolved per record: a record without it is reported and the block is not drawn.
y	number | string		**Required.** Distance from the origin's TOP edge to the block's top edge, in points: the trim page's top edge, or -- when the nearest enclosing container that writes "childOrigin" says "container" -- that container's box's top edge. A STRING is a data token (or text holding one, or a number as text), resolved per record: a record without it is reported and the block is not drawn.

Borders

Schema definition: border. Stroke width and color.

A border: {width, color}, or a color (hairline) / number (black) shorthand. The color may be an ink object such as {"cmyk": [0, 1, 1, 0]}; "none" (or "transparent") draws no border.

Example

```
{
  "width": 0.75,
  "color": "#2a4d7a"
}
```

Key	Type	Default	Description
color	color		Line color. "none" (or "transparent") draws no line at any width.
dash	array of number | number		Dash pattern: alternating on/off run lengths in points, repeating. A single number means equal on and off. Omit or [] for a solid line. A pattern whose lengths are all zero is refused with a warning and the line draws solid -- an all-zero array is a division by zero in most renderers and an invisible line in the rest.
dashPhase	number		How far INTO the dash pattern the line starts, in points. Shifts the rhythm without changing it -- two parallel dashed rules with different phases read as a weave rather than as a grid.
lineCap	string		The shape of a stroke's free ends, and of each dash. "butt" (the default) stops square at the endpoint; "round" and "square" both extend half the stroke width past it. A dotted line is a dash of [0, gap] with a ROUND cap -- with butt ends the same pattern draws nothing at all. One of: butt, round, square.
lineJoin	string		The shape of a stroke's corners. "miter" (the default) extends both edges to a point, "round" arcs them, "bevel" cuts the point off. Only meaningful where one PATH turns a corner: a polygon, a polyline, a path, or a rectangle's border. One of: miter, round, bevel .
miterLimit	number		How far a mitered corner may spike before it is cut to a bevel, as a multiple of the stroke width. PDF's default is 10 (SVG's too). A sharp angle at a heavy weight can throw a spike many times the line width, which is what this bounds.
width	number		Line thickness, in points. Leave it out and give a color, and the line is a 0.75pt hairline, as the color shorthand draws.

Insets and padding

Schema definition: insets. Four-edge measurements.

Edge insets: one number for all four edges, [top, right, bottom, left], or per-edge keys.

One value for all four edges

```
10
```

Per edge

```
{
  "top": 6,
  "right": 10,
  "bottom": 6,
  "left": 10
}
```

Key	Type	Default	Description
`bottom`	`number`		Inset on the bottom edge, in points.
`left`	`number`		Inset on the left edge, in points.
`right`	`number`		Inset on the right edge, in points.
`top`	`number`		Inset on the top edge, in points.

Gradients

Schema definition: `gradient`. Axial and radial fills.
A gradient fill: a color ramp along an axis (linear / axial) or out from a center (radial). Rendered as a true PDF axial (Type 2) or radial (Type 3) shading.

A diagonal linear ramp

```
{
  "type": "linear",
  "angle": 45,
  "stops": [
    {
      "offset": 0,
      "color": "#1f4e79"
    },
    {
      "offset": 1,
      "color": "#7ba44f"
    }
  ]
}
```

A radial badge

```
{
  "type": "radial",
  "center": [
    0.4,
    0.4
  ],
  "radius": 0.7,
  "stops": [
    {
      "offset": 0,
      "color": "#ffffff"
    },
    {
      "offset": 1,
      "color": "#1a4fa0"
    }
  ]
}
```

Key	Type	Default	Description
`angle`	`number`		Linear gradient direction in degrees: 0 = left->right, 90 = top->bottom. Shorthand for from/to.
`center`	`array of number`		Radial center [x, y] as box fractions 0..1 (default [0.5, 0.5]).
`from`	`array of number`		Linear axis start [x, y] as box fractions 0..1 (default [0, 0.5]).
`radius`	`number`		Radial outer radius as a fraction of the box's longer side (default 0.5).
`stops`	`array of object`		**Required.** Two or more color stops, each { offset: 0..1, color: <color> }. Each stop sits at its offset: before the first stop the ramp is that stop's color, after the last stop the last stop's color, linear between consecutive stops (stops at 0.3 and 0.7 are solid to 30% and from 70%). Sorted by offset, keeping the written order of stops at the same offset, so two stops at one offset make a hard edge.
`to`	`array of number`		Linear axis end [x, y] as box fractions 0..1 (default [1, 0.5]).
`type`	`string`		linear (axial, default) or radial. One of: `linear`, `radial`.

Form-field spec

Schema definition: `fieldSpec`. Keys shared by every interactive field.
Interactive AcroForm field keys, shared by the "field" block and inline run fields.

Example

```
{
  "field": "text",
  "name": "email",
  "caption": "Email address",
  "tooltip": "We never share it",
  "required": true
}
```

Key	Type	Default	Description
`caption`	`string`		Push-button caption.
`checked`	`boolean`		Initial state of a checkbox or radio button.
`combo`	`boolean`		true = a dropdown combo (choice fields).
`field`	`string`		The field kind. One of: `text`, `checkbox`, `check`, `radio`, `choice`, `combo`, `list`, `button`, `signature`, `sig`.
`fontSize`	`number`		Point size for the field's own text. 0 means auto-size to the field height.
`list`	`boolean`		true = a list box rather than a dropdown (choice fields).
`multiline`	`boolean`		Let a text field hold more than one line.
`name`	`string`		Fully-qualified field name (radios sharing a name form one group).
`options`	`array of string | object`		Choice options: strings, or { value, label } pairs.
`password`	`boolean`		Mask typed characters, for a password entry field.
`readOnly`	`boolean`		Make the field read-only: it renders and prints but cannot be edited.
`readonly`	`boolean`		All-lowercase alias for "readOnly".
`required`	`boolean`		Mark the field required in the PDF form.
`tooltip`	`string`		/TU alternate description (also the PDF/UA accessible name).
`value`	`string`		Initial value (text/choice), or export value (checkbox/radio).

Page setup

Schema definition: `pageSetup`. Page size, margins, columns, running bands.

Page geometry and print setup: size, orientation, margins, columns, running heads, page numbering, and the print-production extras (bleed, marks, color bars, letterhead).

Example

```
{
  "size": "Letter",
  "margins": 64,
  "columns": 2,
  "header": {
    "center": "{{title}}"
  },
  "footer": {
    "right": "{{page}} / {{pages}}"
  }
}
```

Key	Type	Default	Description
`background`	`color`		Page background color, painted to the full bleed extent as an artifact.
`bleed`	`insets`		Bleed allowance in points: one number for all edges or per-edge insets. The MediaBox grows by it; TrimBox stays the trim page and BleedBox covers the bleed.
`colorBars`	`boolean`		Process color bars (K tints + C/M/Y solids) in the bottom slug area.
`columnGap`	`number`		Gutter between columns, in points.
`columns`	`integer`		Number of text columns the content box is divided into. Default 1.
`embedFont`	`string`		System font to embed (subset) instead of the Base-14 path. Forced to "Helvetica" for tagged / PDF/A docs with no value. If the family cannot be embedded on the rendering machine the document is REFUSED, not substituted and not written -- unembedded text is invisible off the machine that made it and cannot honestly carry a PDF/UA or PDF/A claim. Preflight with `aplayoutapp -auditfonts <spec>`; `-allowunembeddedfonts` downgrades the refusal to a warning for diagnosis.
`footer`	`runningHF`		Running footer drawn at the bottom of every page in range.
`footerMargin`	`number`		Distance from the page edge to the footer band (< 0 => sit on the bottom margin).
`header`	`runningHF`		Running header drawn at the top of every page in range.
`headerMargin`	`number`		Distance from the page edge to the header band (< 0 => sit on the top margin).
`margin`	`insets`		Alias for "margins".
`margins`	`insets`		Page margins. The area inside them is the content box that text flows through.

Key	Type	Default	Description
marks	boolean		Draw printer's marks (corner crop marks + registration targets) outside the bleed. Registration marks use the /All separation so they appear on every plate.
mirrorMargins	boolean		Swap left/right margins on verso (even) pages - the book gutter convention. Content width is unchanged.
numbering	numbering		How page numbers are formatted and where they restart. (This is PAGE numbering - folios. The document-level "numbering" is a different key: it counts headings, figures, tables, listings and equations.)
orientation	string		Page orientation. Landscape swaps the width and height of the chosen size. One of: portrait, landscape.
pageSize	pageSize		Alias for "size".
rotate	integer		Page rotation: a display /Rotate on the page (a multiple of 90; normalized). Turns the whole page for viewing/printing WITHOUT changing layout coordinates - a landscape fold-out in a portrait book is a wide (landscape) page plus rotate:90. Distinct from orientation, which swaps the MediaBox.
sideHeadGap	number		Side-head area: gap in points between the side-head band and the body column (default 18).
sideHeadSide	string		Side-head area: which side the band sits on. left/right are absolute physical sides; outside/inside are binding-relative and flip per recto/verso (outside = outer edge, inside = spine edge), independent of mirrorMargins. One of: left, right, outside, inside.
sideHeadWidth	number		Side-head area: reserve a band this many points wide beside the body column for margin headings. When > 0 the body column is narrowed by (sideHeadWidth + sideHeadGap) document-wide, and a heading with "sidehead": true is drawn in the band with its top aligned to the body that follows (which flows beside it). 0 (default) = no side-head area.
sidenoteGap	number		Alias for sideHeadGap - the gap between the band and the body column.
sidenoteSide	string		Alias for sideHeadSide. One of: left, right, outside, inside.
sidenoteWidth	number		Alias for sideHeadWidth. Side heads and margin notes (run key "sidenote") share ONE reserved band; use whichever spelling matches what the band is for.
size	pageSize		Page size - a named size such as "Letter" or "A4", or explicit dimensions.
slug	boolean | string		Slug line in the top slug area: true for "{{title}} - {{date}} - p. {{page}}", or a custom template using those variables.
template	pageTemplate		Page template - repeating furniture such as a letterhead PDF drawn under each page.

Key	Type	Default	Description
`verticalAlign`	`string`		Vertical justification: position this section's body content in the page. "top" (default). "center" / "bottom" ("middle" = center) shift the whole content block, applied only when the section fits on ONE single-column page (a title page / opener / colophon), so nothing splits and the shift is exact. "justify" feathers a MULTI-page single-column flow - spreading each page's leftover slack across its inter-paragraph gaps so the bottoms line up flush (every page but the last). All modes fall back to top on a multi-column section, a baseline grid, or (for justify) any float / side head / straddle / anchored frame / footnote / tall inline atom. One of: `top`, `center`, `bottom`, `middle`, `justify`.
`watermark`	`watermark`		Watermark drawn behind the content of every page in range.
`writingMode`	`string`		Default writing mode for the document's blocks: a block that does not set its own takes this, and a section can override it. "vertical-rl" sets the prose in 直排 (tategaki) columns; "horizontal-tb" is the default. The page's own geometry - margins, columns, running bands - is unaffected. One of: `horizontal-tb`, `horizontal`, `lr-tb`, `yokogaki`, 横排, `vertical-rl`, `vertical`, `tb-rl`, `tategaki`, 直排, 縦書き, `vertical-lr`, `tb-lr`.

Running bands

Schema definition: `runningBand`. Headers and footers.

One running band. Slots may use the {{page}}, {{pages}}, {{title}}, {{date}} variables.

Example

```
{
  "height": 24,
  "style": {
    "fontFamily": "Helvetica",
    "fontSize": 9
  },
  "left": [
    {
      "text": "Appligent, Inc."
    }
  ],
  "right": [
    {
      "text": "Page {{page}}"
    }
  ],
  "rule": {
    "width": 0.5,
    "color": "#cdd4de"
  }
}
```

Key	Type	Default	Description
`center`	`slotRuns`		Content for the center slot.
`height`	`number`		Height reserved for the band, in points.
`image`	`object`		A logo / image drawn in the band (e.g. a letterhead mark): placed horizontally by align, vertically centered; the band grows to fit it.
`left`	`slotRuns`		Content for the left-hand slot.
`right`	`slotRuns`		Content for the right-hand slot.
`style`	`style`		Default style for every slot in the band.

Page numbering

Schema definition: `numbering`. Folio start value and numeral style.
PAGE numbering for a section - the folio sequence. Not to be confused with the document-level "numbering", which counts headings, figures, tables, listings and equations.

Lower-roman front matter

```
{
  "startAt": 1,
  "format": "lower-roman"
}
```

Zero-padded Bates numbers

```
{
  "format": "arabic",
  "startAt": 1,
  "pad": 6
}
```

Key	Type	Default	Description
`format`	`string`		Numeral style for the page number - arabic, roman or alphabetic in either case, or Chinese numerals ("hanzi", or "hanzi-traditional" for 萬 rather than 万). PDF has no page-label style for Chinese numerals, so such a page carries its whole rendered label as the /PageLabels /P prefix. One of: `arabic`, `decimal`, `1`, `roman`, `lower-roman`, `i`, `ROMAN`, `upper-roman`, `I`, `alpha`, `lower-alpha`, `a`, `ALPHA`, `upper-alpha`, ...
`pad`	`integer`		Zero-pad arabic page numbers to N digits (the Bates idiom); set startAt to continue across documents.
`startAt`	`integer`		First page number in the sequence. (This is PAGE numbering - folios. The document-level "numbering" is a different key: it counts headings, figures, tables, listings and equations.)

Automatic numbering

Schema definition: `captionNumbering`. Numbering headings, captions and equations.

Automatic numbering for one counter (figures, tables, listings or equations). true takes the defaults; an object tunes them. The label is composed as labelWord + " " + prefix + [chapter + prefixSep] + number + suffix - prepended to the caption for a captioned block, or drawn in the right gutter for an equation.

Example

```
{
  "enabled": true,
  "labelWord": "Figure",
  "prefixWithHeading": true,
  "prefixLevel": 1,
  "prefixSep": "."
}
```

Key	Type	Default	Description
`enabled`	`boolean`	true	Turn this counter off. Supplying the object at all implies on, so this is only needed to disable a counter a template switched on.
`format`	`string`	arabic	Numeral style for the number itself. One of: `arabic`, `decimal`, `1`, `roman`, `lower-roman`, `i`, `ROMAN`, `upper-roman`, `I`, `alpha`, `lower-alpha`, `a`, `ALPHA`, `upper-alpha`, ...
`from`	`integer`	1	Value given to the first caption.
`labelWord`	`string`		"Figure" / "Table" / "Listing"; empty string gives a bare number. Equations default to empty - their label IS the wrapped number. A "{}" in the word says where the number goes instead of the default trailing space, which is how a Chinese caption reads 图1-1: "图{}".
`prefix`	`string`	""	Wraps the number's left side. Empty for captions; "(" for equations, whose label is "(3.2)".
`prefixLevel`	`integer`	1	Heading level that supplies the chapter number.
`prefixSep`	`string`	-	Between chapter and ordinal.
`prefixWithHeading`	`boolean`	false	Chapter-relative numbering: "Figure 3-2". The counter restarts at each heading of prefixLevel.
`suffix`	`string`	.	Between the number and the caption text.

Watermarks

Schema definition: `watermark`. Stamps drawn on every page.

A page watermark / stamp repeated on the section's pages: text (or an image) drawn diagonally / at an angle / centered / top / bottom, solid or outline, under or over the body, with constant alpha. Always a Pagination/Watermark artifact.

Example

```
{
  "text": "DRAFT",
  "angle": 45,
  "mode": "dual",
  "opacity": 0.12,
  "color": "#c0202a"
}
```

Key	Type	Default	Description
`angle`	`number`		Degrees counter-clockwise, with position "angle" (setting it implies the position).
`autoScale`	`number`		Font size as a percentage of the page height (omitted = fit to the extent).
`color`	`color`		Watermark color.
`image`	`string`		Image file stamped instead of text (PNG/JPEG/GIF/TIFF/WebP, or HEIC/HEIF on macOS).
`justify`	`string`		Multi-line alignment: left / center (default) / right. For top/bottom it also anchors the block on the page; for diagonal / diagonal-down / center / angle it justifies each line within the rotated text block. One of: `left`, `center`, `right`.
`layer`	`string`		Under or over the body content (dual draws both). One of: `under`, `over`.
`mode`	`string`		solid fill, hollow outline, or dual (solid under + outline over - reads on text and image pages alike). One of: `solid`, `outline`, `dual`.
`offsetX`	`number`		Horizontal nudge from the chosen position, in points.
`offsetY`	`number`		Vertical nudge from the chosen position, in points.
`opacity`	`number`		Opacity from 0 (invisible) to 1 (solid). A watermark is drawn as a page artifact, so it stays out of the reading order.
`pages`	`string`		Which pages to stamp (first/rest are relative to the section). One of: `all`, `even`, `odd`, `first`, `rest`.
`position`	`string`		Where the watermark sits and how it is angled. "angle" uses the explicit "angle" property instead of a preset. One of: `diagonal`, `diagonal-down`, `center`, `top`, `bottom`, `angle`.
`scale`	`number`		Image width as a percentage of the page width (default 50).
`size`	`number`		Explicit font size in points (0/omitted = automatic).
`src`	`string`		Alias for "image".
`text`	`string | array of string`		Watermark text; an array gives one line per entry (or use \n).

Page templates

Schema definition: `pageTemplate`. Reusable page furniture.

Letterhead / stationery: one page of an external PDF drawn to the full trim page under the body (a Page artifact). For PDF/UA the source's fonts must be embedded.

Example

```
{
  "src": "letterhead.pdf",
  "page": 1,
  "on": "first"
}
```

Key	Type	Default	Description
`on`	`string`		Which pages get the underlay: every page, only the first, or every page after the first. One of: `all`, `first`, `rest`.
`page`	`integer`		Which page of "src" to draw, 1-based. Default 1.
`src`	`string`		External PDF whose page is drawn as a full-page underlay (vector Form XObject).

Table cells

Schema definition: `tableCell`. One cell of a table row.

One table cell: a bare array of blocks, or an object with span/padding/valign/style + blocks.

Example

```
{
  "colspan": 2,
  "valign": "middle",
  "style": {
    "background": "#eef2f8"
  },
  "blocks": [
    {
      "type": "paragraph",
      "text": "A merged, tinted header cell"
    }
  ]
}
```

Key	Type	Default	Description
`blocks`	`array of block`		Full block content for the cell when a plain string is not enough: a paragraph with runs, a list, a nested table, an image.
`colspan`	`integer`		How many columns this cell spans. Default 1.
`header`	`boolean`		Tag THIS ONE cell as a header (TH), whatever its row and column are. A row's "header" makes every cell in the row a header and "headerCols" makes the leading N columns headers in every row; neither can say "this label cell heads its row and the values beside it are data", which is the shape of a subtotal row. Scopes as a row header, or Both inside a header row.
`headers`	`string | array of string`		PDF/UA: id(s) of the header cells that govern this data cell (/Headers). With `rowGroups` the engine DERIVES this for every cell -- the row header immediately to its left plus its column header -- so an authored list is only needed for a table that does not use row groups.
`id`	`string`		PDF/UA cell identifier, referenced by other cells' "headers".
`padding`	`insets`		Padding inside this one cell, overriding the table's "cellPadding".
`rotate`	`string`		Draw this cell's text label VERTICALLY (a narrow-column header). 90 = read bottom-to-top, 270 = read top-to-bottom; any other value = upright. The cell's height grows to the label's length and its column stays narrow. The rotation rides the text matrix, so the TH tag and PDF/UA stay intact. Scope: a text label (rich block content in a rotated cell is not laid out rotated). One of: `0`, `90`, `270`.
`rowspan`	`integer`		How many rows this cell spans. Default 1.
`style`	`style`		Style for this cell - including "background", which is how a single cell is tinted.
`valign`	`string`		Vertical alignment of the cell's content within the row height. One of: `top`, `middle`, `center`, `bottom`.

Chart axes

Schema definition: `chartAxis`. Axis labels, ticks and gridlines.

One chart axis: its title, bounds, tick spacing, gridlines and scale. These all describe a NUMERIC axis. When the chart supplies `categories`, its x axis becomes a category axis instead - one tick per category, no numeric rounding - and the keys below behave as each notes.

Example

```
{
  "label": "Revenue",
  "unit": "K",
  "min": 0,
  "max": 120,
  "majorTick": 20,
  "grid": true
}
```

Key	Type	Default	Description
`grid`	`boolean`		Hairline gridlines at the major ticks. On a category axis, one per LABELED category, on the slot center with its tick - rule, tick and name are one decision, so the axis never carries a mark it cannot explain.
`label`	`string`		Axis title, drawn alongside the axis.
`logScale`	`boolean`		Plot this axis logarithmically (base 10). Major ticks land on decades with minor ticks at 2x..9x, and labels read as plain decimals while they stay short (0.001 to 10000) and as 1e5 / 1e-6 beyond. The axis never starts at zero, so `min` is taken from the smallest positive value in the data unless you set it; values at or below that floor are drawn on it rather than dropped. Honored on the cartesian chart types - line, bar, scatter, area and combo - on both `xAxis` and `yAxis`. Not honored on waterfall, candlestick, box-plot, gauge or radar, whose value axes are linear by construction, nor on `yAxis2`. Not honored on an x axis carrying `categories` either: a category index is ordinal, not a magnitude, and a log scale would map two adjacent slots onto the same pixel.
`majorTick`	`number`		Explicit major-tick spacing; omit for automatic nice 1/2/5x10^n. On a category axis this reads as a whole-number stride - label every Nth category - since a fractional step would put a tick where no category exists. Omit it there and the stride is measured from the widest label so the names cannot collide.
`max`	`number`		Fixed upper bound. Omit to take it from the data. On a category axis, see `min`.
`min`	`number`		Fixed lower bound. Omit to take it from the data. On a category axis, omitting it is normal: the range is pinned to the categories with half a slot of air at each end, so the outermost bar sits inside the frame. Setting it pins that edge instead, and any category falling outside the plot is not drawn.
`minorTicks`	`integer`		Minor subdivisions per major interval. Ignored on a category axis (there is nothing between two adjacent slots to subdivide) and on a log axis, which always draws its own 2x..9x minors.
`unit`	`string`		Suffix appended to every tick label ("K", "%", "ms", ...). Ignored on a category axis, where the label is the category name.

Index terms

Schema definition: `indexTerm`. What the index scanner looks for.
One index term: a bare word/phrase, or an object with an optional match pattern and one level of sub-entries.

Example

```
{
  "term": "stability",
  "subentries": [
    {
      "term": "static margin"
    },
    {
      "term": "dynamic"
    }
  ]
}
```

Key	Type	Default	Description
`label`	`string`		Alias for "term".
`lang`	`string`		BCP-47 language tag of this entry's label (e.g. "de-DE" for {"term": "Weltanschauung", "lang": "de-DE"}), emitted as /Lang on the entry so a screen reader pronounces it in that language. Without it an entry whose label is the body text it matches reads in the language of the term's first occurrence, and one whose label is your own wording (a "match" of other words, or a label-only group header) reads as the index; this wins over both. It names the label, not the index: the page numbers ride inside the entry as they always do, and a Hanzi page number or the engine's "(continued)" still reads as the index. A sub-entry takes its own. "de_DE" is written de-DE; a value that is no language tag ("x!!bad") is ignored, and the entry reads as it would without it; either is reported once per spelling, naming the term. Tagged documents only.
`match`	`string`		The text matched in the body when it differs from the label. An explicit empty string makes the entry label-only: never scanned, rendered as a group header when its subentries have hits.
`sortAs`	`string`		The key this entry sorts and files under, replacing the label. Wins in every collation, and is the escape hatch a Chinese index needs: no reading table can be right for every proper noun or 多音字 - a character whose reading depends on the word - so 长江 takes "changjiang" (Chángjing, not zhng) and 重庆 takes "chongqing". It is also how a Latin-ordered index files a Chinese term. InDesign spells this "Yomi", makeindex spells it key@display.
`subentries`	`array of indexTerm`		Sub-entries nested one level under this term.
`term`	`string`		The displayed entry label (also the match text when no "match").

Part V — The document object

17. Document-level keys

The keys of the top-level object: metadata that lands in the PDF's `/Info` dictionary and XMP, the standards switches, the typography defaults, and the two content keys.

Key	Type	Default	Description
`attachments`	`array of object`		Associated files embedded as PDF/A-3 attachments (catalog /AF + /Names /EmbeddedFiles). Any attachment promotes a PDF/A-2 conformance level to the matching PDF/A-3 level.
`author`	`string`		/Info Author + XMP dc:creator.
`baselineGrid`	`number`		Baseline grid pitch in points (0 = off): text-line baselines snap to a page-absolute grid (measured from the content top) so columns and facing pages align. Set it to your body leading (fontSize x lineHeight).
`binding`	`string`		Which edge the book is bound on. "left" (default) is the Western binding: the spine is on the left of a recto and the pages progress left to right. "right" is the 右綴じ binding of a vertical Chinese or Japanese book - the spine on the right, the pages read back to front. A right binding writes /ViewerPreferences /Direction /R2L, so a two-up viewer lays the spreads out in reading order, and it mirrors every binding-relative side: with mirrorMargins the wider spine margin moves to the other edge, and inside/outside side heads and margin frames follow. Independent of writingMode - a right-bound book of horizontal text is a real thing, and one vertical block never re-binds the document. One of: `left`, `right`, `ltr`, `rtl`, `l2r`, `r2l`.
`blocks`	`array of block`		The document body as a single implicit section. Use "sections" instead for multi-section documents.
`canvas`	`boolean`		Read-only echo of canvas mode, which is turned on by the presence of "pages". Setting it without "pages" does nothing -- there would be no designed page to apply it to.
`changeBars`	`object`		Revision / change bars: the appearance of the vertical bar drawn in the page margin beside any content marked changed (a block's changed:true, or a block carrying a condition tag declared with changeBar:true). This object only styles the bar; it is not an on/off switch - a bar draws wherever changed content exists. The bar is a PDF artifact, out of the tagged reading order.
`chapters`	`array of string | object`		Native book model: assemble N chapter spec files into ONE PDF. Each entry is a chapter file path (relative to this book spec), or an object {src, restart}. Chapters are loaded through the full per-chapter pipeline (their images/includes/data resolve relative to the chapter's own directory) and merged as sections, so the TOC/index/list-of-figures, cross-file cross-references and the tagged-PDF structure tree span every chapter. This document's own top-level `blocks` become front matter (a cover, a master TOC) before chapter 1, and its pageSetup / running header+footer / numbering / defaultStyle are the book defaults each chapter inherits (and may override).

Key	Type	Default	Description
colorMode	string		"cmyk" renders the whole document in DeviceCMYK (RGB inputs are converted, so the file carries no DeviceRGB) and a PDF/A document then gets a CMYK OutputIntent (the bundled U.S. Web Coated SWOP v2, or outputProfile) instead of sRGB. Default "rgb". One of: rgb, cmyk, RGB, CMYK.
conditionIndicators	boolean		Conditional text author view: when true, conditional content is NOT dropped - every conditional block/run is tinted in its tag's color and the hidden ones are struck through, so an editor can see what is conditional. Off (default) = final build.
conditions	object		Conditional text (FrameMaker condition tags): maps a condition-tag name to its state - a bare boolean (true = shown), or an object { show, color }. A block / run / table-row "condition" expression (a boolean grammar over tag names: AND / OR / NOT / parentheses, with juxtaposition = AND) is evaluated against this map; an unlisted tag defaults to shown. In a final build hidden content is removed in a pre-pass before layout; see conditionIndicators for the author view.
creator	string		/Info Creator + XMP xmp:CreatorTool (default "PageWright").
data	object | array		Data-driven binding context: a record (object) whose fields are substituted into every {{path}} token in the document before layout, e.g. {{customer.name}} or {{items.0.sku}}. Reserved page variables ({{page}} {{pages}} {{date}} {{time}} {{datetime}} {{title}}) are not consumed here. Mutually exclusive with dataFile.
dataFile	string		Path (relative to the spec) to an external record used as the data binding context when no inline data is given. JSON, or CSV when the path ends in .csv - a CSV's first line names the fields, one data row binds as the record itself and two or more bind as an array.
declarationGate	string		How the WCAG gate reacts when this document contradicts a WCAG level it declares in declarations. "warn" (the default whenever a WCAG declaration is present) reports every finding on stderr and still writes the file - the claim is the author's. "strict" refuses to write the file at all, so no PDF can ship carrying a claim the engine just contradicted. "off" skips the checking. A document that declares no WCAG level is never gated. The gate decides only what a layout model can decide - alt-text presence (not accuracy), structure and table headers, text contrast, title, link purpose, heading order and language; a clean run means nothing checkable contradicts the claim, NOT that the document is WCAG conformant. One of: off, warn, strict.
declarations	array of object | string		PDF Declarations (PDF Association): machine-readable claims, written to the XMP pdfd:declarations bag, that this document conforms to a standard PDF has no identifier of its own for - WCAG 2.1/2.2 at level A/AA/AAA, Well-Tagged PDF, ISO/TS 32005, or a profile you define. PDF/A, PDF/UA and PDF/X have their own identifiers and are declared with pdfa, pdfua and pdfx instead. A declaration is a CLAIM, not a validation result, and it is yours: the engine writes exactly what you list here, and never adds a declaration or fills in a claimant on your behalf. An entry may be a bare alias string as shorthand for { "conformsTo": ... }.
defaultStyle	style		Document-wide default style - the base every block inherits before its own style applies.

Key	Type	Default	Description
`fallbackFont`	`string`		CJK fallback font family for the engine's auto-generated blocks (toc / lof / glossary / index). Those blocks flatten source text (headings, captions, terms) into one block font; when it is Latin, CJK characters render as blank boxes. Set a CJK-capable family here and each wide/CJK segment is drawn in it while the rest stays in the block font. It also reaches body text: any character the run's own face does not have (an emoji, a script the face lacks) is drawn in this family, so an emoji face such as NotoColorEmoji here makes emoji typed in ordinary text print in colour. A character neither face has still prints as .notdef, and one the fallback face lacks is reported. Empty = off.
`features`	`array of string`		The document-wide OpenType feature set - what every run is set with unless its own style names a different one. The same vocabulary as the per-style `features` key, and the general form of `ligatures` above (which is folded in as `liga`). Use this for a decision that really is document-wide, such as oldstyle figures throughout a book, and the style key for the exceptions.
`fontDirs`	`array of string`		Extra font FOLDERS this document may name faces from, searched before the installed ones, so a spec can travel with a face that is not installed on the rendering machine. Relative paths resolve against the spec file's own directory. Folders, not files: the same list is given to the engine's font index and to the PDF library that embeds the face, and the library takes directories only. Under -batch the union over every spec in the job list is used, and a book's chapters contribute theirs; on the apDocLib server a document's folders reach the resolver but not the embedder, which is fixed at startup by the server's own -fontdir.
`hangingPunctuation`	`boolean`		CJK hanging punctuation (ぶら下げ): a trailing ideographic comma/full stop (、 。) or its full-width Latin form (， ．) hangs past the line edge into the margin instead of wrapping to the next line.
`hyphenate`	`boolean`		Hyphenate Latin words at line ends (Knuth/Liang patterns; en, fr, de and es are embedded). Each word divides by the rules of ITS OWN language -- the nearest `lang` on the run, the block or the document -- so a quoted German passage inside an English report breaks as German. A language with no embedded pattern set is left undivided rather than guessed at in English.
`images`	`object`		Output policy for placed raster images, and a low-resolution report (minPpi).

Key	Type	Default	Description
`impose`	`object`		How a print copy of this document is laid out on sheets: a saddle-stitched booklet, two pages to a sheet side, for sheets printed on both sides, folded once and stapled on the fold. The key changes nothing in the document itself. The print copy is a second PDF, made from the first one's pages only when one is asked for (`pagewrightapp spec.json out.pdf -impose booklet.pdf`, or Export Booklet PDF in PageWright Studio's File menu). It is untagged and claims no conformance, because a sheet has no reading order: the reading PDF stays the accessible one. Its /Subject says how to print it, and its viewer preferences preset two-sided, actual-size printing on the sheet it was laid out for. A right `binding` mirrors every sheet. Every render of a document that carries the key says what its page count costs the booklet: a count that is not a multiple of four is padded with blank pages.
`kashida`	`boolean`	false	Kashida (tatweel) justification for Arabic: a JUSTIFIED Arabic line spends its slack elongating the connections between letters - inserting the face's own tatweel glyph (U+0640) at the joints where a letter joins onward - and only what the tatweels cannot absorb goes to the word gaps. That is what Arabic typography does; Latin justification applied to Arabic leaves the words floating in holes. Off by default, like every other typographic switch here, because turning it on changes the ink on every justified Arabic line. Needs an EMBEDDED face that shaped through its GSUB (see the Arabic samples) - a run drawn from Unicode presentation forms has no joints this can reach - and has no effect on any other script, or on Arabic that is ragged rather than justified. The quantum is coarse (a tatweel is about 0.195 em), so the line still ends exactly on the margin by handing the remainder back to the word gaps. The engine spreads the elongation evenly across the line; it does not implement the priority order classical Arabic typography uses among candidate joints.
`kerning`	`boolean`		Apply pair kerning: tightens pairs such as AV, To, Wa, F. and P. The built-in Helvetica kerns from the canonical Adobe AFM tables; any other embedded font kerns from its own OpenType kern and GPOS pair tables, located by name in the system font folders - including faces housed inside TrueType collections (.ttc). Measured and drawn alike, so justification and line breaks stay exact. A font whose file cannot be found simply renders unkerned.
`keywords`	`string`		/Info Keywords + XMP pdf:Keywords.
`lang`	`string`		Document language, BCP-47 (e.g. "en-US"). Default when tagged, and the hyphenation language for text that declares none of its own.

Key	Type	Default	Description
`ligatures`	`boolean`	false	Apply the embedded face's OpenType standard ligatures (the `liga` feature): fi, fl, ff, ffi and ffl are drawn as the single glyph the type designer cut for them instead of as colliding letters. Off by default, because turning it on reflows every line that gains a ligature and that is the author's call. Measured and drawn from the same glyphs, so justification and line breaks stay exact - a ligature is narrower than the letters it replaces. Base-14 faces carry no GSUB and are unaffected, as is any embedded face whose `liga` is absent, so a document with mixed faces is safe. The letters remain the stored text: a ligature glyph declares the characters it stands for through /ToUnicode, so copy, search and a screen reader all still see "office". Shorthand for `"features": ["liga"]`; setting both is safe, since this key can only add.
`linkStyle`	`boolean | object`		Make links LOOK clickable. A PDF link annotation is invisible by default, so without this nothing distinguishes linked text and a reader can only find it by hovering. true uses the defaults (a conventional link blue, underlined); an object overrides them. Only fills what a run left unset - a run that sets its own color or underline keeps it. Off by default: enabling it would restyle every existing document.
`master`	`string`		The document-wide DEFAULT master page (a key of "masters"). Applied to every section - and the implicit top-level section - that names no master of its own.
`masters`	`object`		Master pages: named, reusable page templates. Each master is an ordinary pageSetup object (size/margins/columns/header/footer/numbering/background/watermark/template/bleed+marks) PLUS an optional "frames" array of positioned background blocks stamped on every page that uses the master, as page artifacts. A master's decoration is merged into each using section's pageSetup (doc default -> master -> section overrides).
`meta`	`object`		Optional metadata object; its fields override the top-level ones. One field lives only here: rights, a copyright or licence notice written to the XMP as dc:rights.
`namedDestinations`	`boolean`	true	Publish every block "id" as a PDF named destination (catalog /Names /Dests), so `guide.pdf#nameddest=chapter3` opens at that block and other tools can deep-link into the document. Additive and invisible, so it is on by default; set false to omit the name tree.
`notes`	`object`		Footnote / endnote streams: the marker glyph style and, for footnotes, the counter-restart scope. Absent => numeric markers with footnotes counted continuously through the document (unchanged from before).
`numbering`	`object`		Automatic content numbering. Absent => nothing is numbered.
`opticalMargins`	`boolean`		Latin optical margins: trailing punctuation (. , ; : - - - and quotes) hangs slightly past the line edge so the body's optical right edge stays flush (most visible in justified text).
`optimalBreaks`	`boolean`		Knuth-Plass optimal line breaking for justified paragraphs: choose breakpoints that minimize the total badness across the whole paragraph (more even spacing, fewer rivers) instead of the default greedy line-by-line fit.

Key	Type	Default	Description
`outputProfile`	`string | object`		The CMYK press this document is printed on, as an ICC output profile file - in place of the built-in U.S. Web Coated (SWOP) v2. In cmyk color mode (colorMode "cmyk", or pdfx) every RGB and gray color, raster image, gradient stop and emoji is converted to it, and a PDF/X or PDF/A file embeds it as the OutputIntent's DestOutputProfile, identified as "Custom" with the profile's own description as Info unless you give a registered identifier. The path resolves against the spec file's directory. The profile must be a CMYK printer ('prtr') profile, ICC version 2 to 4, that the color engine can convert to; anything else is refused with one notice and the document converts to and declares the built-in SWOP. Ignored, with a notice, in rgb mode. Colors already in CMYK (cmyk colors, spot alternates, CMYK images) are not converted. Either a path string or an object.
`overprint`	`boolean`		Spot-ink (separation) fills and strokes overprint instead of knocking out (/OP /op true, /OPM 1). For print workflows where a spot accent must not knock a hole in the underlying process color.
`pageSetup`	`pageSetup`		Page geometry and print setup for the document: size, orientation, margins, columns, running heads and the print-production extras. A section can override it.
`pages`	`array of object`		CANVAS MODE: the document is a stack of DESIGNED pages rather than a flow. Each entry is one page and the elements on it, and every element needs a "position". A canvas page IS a section that makes exactly one page, so page size, margins, masters, running bands, page labels, links and tagging all work here unchanged. Present => canvas mode is on, which adds the two things a design needs and a flow does not: elements paint in "z" order and each occupies ONE z slot, so a shape declared after a text element draws OVER it (in a flow it cannot -- all text is written after all graphics); and the structure tree keeps the DOCUMENT order, so the reading order is the one you wrote while the paint order is the one you designed. Use "blocks" or "sections" for a flowing document.
`palette`	`object`		Document color palette. The one edit that re-skins every chart, bar and table zebra in the document. Every field is optional and every consumer keeps the constant it has always used when a field is absent, so a document with no palette renders exactly as it did before this existed. The order is always: a color written on the block wins, then the palette, then the engine default.
`pdfa`	`string | boolean`		PDF/A archival conformance: "2b"|"2u"|"2a" (PDF/A-2), "3b"|"3u"|"3a" (PDF/A-3, which also permits embedded files), or "4"|"4e"|"4f" (PDF/A-4, the PDF 2.0 revision; "4f" permits embedded files), case-insensitive; or true for a smart default (2a when tagged, else 2u). An "a" level ("2a", "3a") forces tagged. Adding `attachments` promotes a PDF/A-2 level to the PDF/A-3 level with the same letter ("2u" becomes "3u"), and "4" or "4e" to "4f".

Key	Type	Default	Description
`pdfua`	`boolean | string`		PDF/UA target. true (or 1) = PDF/UA-1 (same as "tagged"); 2 = PDF/UA-2 (ISO 14289-2 on PDF 2.0: namespaced structure, structure destinations, pdfuaid:part 2 + rev 2024). Any positive value forces tagging. FALSE IS NOT THE SAME AS ABSENT: it keeps the structure tree but writes no pdfuaid claim at all, for a document that is tagged without having been verified conformant. A structure tree is necessary for PDF/UA and nowhere near sufficient -- the fonts must all be embedded, no tagged content may sit inside an artifact -- so use this rather than ship a claim nothing has checked. One of: `1`, `2`.
`pdfx`	`string | boolean`		PDF/X print-production conformance. "x-4" (case-insensitive; any spelling containing 4) or true badges the file as PDF/X-4 (ISO 15930-7): it forces cmyk color mode, embeds all fonts, writes a GTS_PDFX output intent (U.S. Web Coated SWOP v2, or the document's `outputProfile`) with the pdfxid metadata, gives each page a TrimBox, sets PDF 1.6, and skips encryption. Only X-4 is supported - it permits live transparency and layers, so nothing is flattened.
`producer`	`string`		/Info Producer + XMP pdf:Producer (default "Appligent PageWright").
`recordIndex`	`integer`	1	Which record of "records" this document binds (1-based). Clamped to the record count. Ignored when the CLI supplies -data.
`records`	`string | array`		A record SET for mail merge: a path (relative to the spec) to a JSON array or a CSV whose header row names the fields, or an inline array of record objects. The document renders bound to one record (see recordIndex); the CLI's -data renders one document per record and overrides this.
`renderingIntent`	`renderingIntent`		How colors outside the destination's gamut are mapped, for the whole document. "relative" (relative colorimetric, the default): colors the press can print come through exactly and the rest are clipped to the nearest printable one -- right for flat brand colors, rules and charts. "perceptual": the whole gamut is compressed so a photograph's gradations survive, at the cost of shifting colors that were already printable. "saturation": vividness over accuracy (business graphics). "absolute": relative colorimetric that also simulates the paper's white (proofing). In cmyk color mode (colorMode "cmyk", or pdfx) every RGB-to-CMYK conversion runs at it -- vector colors, images, gradients, emoji, placed PDF pages -- into SWOP or `outputProfile`. In every mode a value other than "relative" is also written into each page (the PDF `ri` operator), so a RIP or viewer converting the file onward uses it too. An image block may name its own. The PDF names ("Perceptual", "RelativeColorimetric", "Saturation", "AbsoluteColorimetric") are accepted too.
`restart`	`boolean`		Book model: restart page numbering at 1 for every chapter (per-chapter folios). Default false = continuous pagination across the whole book. A chapter's own `restart` overrides this.

Key	Type	Default	Description
sectionTags	string	auto	How much <Sect> GROUPING the structure tree carries. "auto" (default, and what this engine has always emitted) opens a nested <Sect> for every heading and wraps each document section in one more. "pages" keeps only the per-section wrapper. "none" emits no <Sect> at all, so headings and content sit directly under <Document>. A FLATTER TREE IS OFTEN THE BETTER ONE. ISO 32000-1 14.8.4.3.5 calls a Sect-grouped tree "strongly structured" and PDF/UA permits either, so no validator has an opinion -- all three modes are veraPDF UA-1 clean. What decides it is how assistive technology reads the result, and professional remediators commonly strip <Sect> for exactly that reason: every level is one more container between the reader and the heading. Measured on this engine's marketing brochure: "auto" 63 Sect elements and 12 levels at its deepest, "none" 0 and 10. The ink is identical in every mode -- this changes structure only. An unknown value refuses the document. One of: auto, pages, none.
sections	array of object		Explicit sections. Each inherits the document pageSetup, then applies its own overrides.
security	object		Standard PDF security (encryption + permissions) applied via aplib APEncrypt. Requires an owner (permissions) password; an empty/absent user password lets the file open with no password. Mutually exclusive with PDF/A.
stories	array of object | object		Free-standing block lists poured into the thread of "frame" blocks that reference them by id (article threading). Either an array of {id, blocks:[...]} or an object { id: {blocks:[...]} | [...] }. Frames in one thread should share a column width.
styles	object		Named stylesheet: each entry is a style object (optionally "basedOn" another entry). Any "style" below may reference an entry by name.
subject	string		/Info Subject + XMP dc:description.
tagged	boolean		Emit a tagged PDF (logical structure tree). Also writes a PDF/UA-1 claim into the XMP unless "pdfua" is false. How much <Sect> GROUPING that tree carries is a separate choice -- see sectionTags.
title	string		Document title: /Info Title, XMP dc:title, and the {{title}} variable.
variables	object		Variable catalog: named values referenced with {{var.name}} anywhere the data binding reaches (text runs, links, image/pdf src, running-band templates, metadata). Defined once, edited in one place; static and page-independent, unlike the render-time {{page}}/{{section}} tokens. Folded into the binding context under var, so a real data record still wins on its own keys. Nested objects are allowed ({{var.company.name}}).
viewer	object		Initial view: catalog /PageLayout, /PageMode and the /OpenAction destination.

Worked document examples

Complete top-level objects, each one turning on a single document-level capability. Paste any of them into a file and render it as it stands.

Sections, each with its own page setup and folio style

```
{
  "title": "Report",
  "tagged": true,
  "sections": [
    {
      "size": "Letter",
      "numbering": {
        "format": "lower-roman"
      },
      "blocks": [
        {
          "type": "heading",
          "level": 1,
          "text": "Preface"
        }
      ]
    },
    {
      "size": "Letter",
      "columns": 2,
      "restartNumbering": true,
      "numbering": {
        "format": "arabic"
      },
      "blocks": [
        {
          "type": "heading",
          "level": 1,
          "text": "Body"
        }
      ]
    }
  ]
}
```

A tagged, accessible document (PDF/UA-1)

```
{
  "title": "Accessible report",
  "tagged": true,
  "lang": "en-US",
  "blocks": [
    {
      "type": "heading",
      "level": 1,
      "text": "Summary"
    },
    {
      "type": "image",
      "src": "chart.png",
      "alt": "Revenue rose nine per cent."
    }
  ]
}
```

Archival PDF/A-2a with an sRGB output intent

```
{
  "title": "Archive",
  "tagged": true,
  "pdfa": "2A",
  "lang": "en-US",
  "blocks": [
    {
      "type": "paragraph",
      "text": "Self-contained by construction."
    }
  ]
}
```

Press-ready PDF/X-4 in DeviceCMYK

```
{
  "title": "Press",
  "pdfx": "x-4",
  "pageSetup": {
    "size": "Letter",
    "bleed": 9,
    "marks": true
  },
  "blocks": [
    {
      "type": "paragraph",
      "text": "Every color converts to CMYK; fonts embed; transparency stays live."
    }
  ]
}
```

PDF/UA-2 with deep-linkable named destinations

```
{
  "title": "UA-2",
  "pdfua": 2,
  "tagged": true,
  "lang": "en-US",
  "namedDestinations": true,
  "blocks": [
    {
      "type": "heading",
      "level": 1,
      "id": "intro",
      "text": "Introduction"
    }
  ]
}
```

Viewer folios from section numbering (a /PageLabels tree)

```
{
  "title": "Folios",
  "sections": [
    {
      "numbering": {
        "format": "lower-roman"
      },
      "blocks": [
        {
          "type": "heading",
          "level": 1,
          "text": "Preface"
        }
      ]
    },
    {
      "restartNumbering": true,
      "numbering": {
        "format": "arabic"
      },
      "blocks": [
        {
          "type": "heading",
          "level": 1,
          "text": "Chapter 1"
        }
      ]
    }
  ]
}
```

Viewer preferences: open on the outline, fit the width

```
{
  "title": "Prefs",
  "viewer": {
    "layout": "single",
    "mode": "outlines",
    "fit": "fitWidth"
  },
  "blocks": [
    {
      "type": "paragraph",
      "text": "Honored by every conforming reader."
    }
  ]
}
```

Endnotes gathered at the document end, with symbol footnote markers

```
{
  "title": "Notes",
  "tagged": true,
  "lang": "en-US",
  "notes": {
    "footnoteMarker": "symbols",
    "footnoteRestart": "section"
  },
  "blocks": [
    {
      "type": "paragraph",
      "runs": [
        {
          "text": "The rocket equation",
          "endnote": "Tsiolkovsky, 1903."
        },
        {
          "text": " sets the ceiling on performance."
        }
      ]
    },
    {
      "type": "endnotes",
      "title": "Notes"
    }
  ]
}
```

Appendices

A. Every element at a glance

Every type name the engine accepts, and where it is documented.

Element	Aliases	Group	Size
`paragraph`	-	Text	32 keys
`heading`	-	Text	36 keys
`list`	-	Text	29 keys
`pre`	`code`	Text	33 keys
`container`	-	Structure and spacing	27 keys
`columns`	-	Structure and spacing	24 keys
`cardGrid`	`cardgrid`	Structure and spacing	26 keys
`spacer`	-	Structure and spacing	22 keys
`rule`	-	Structure and spacing	24 keys
`pagebreak`	`page-break`	Structure and spacing	2 keys
`table`	-	Tables	42 keys
`image`	-	Pictures and drawings	45 keys
`shape`	-	Pictures and drawings	39 keys
`svg`	-	Pictures and drawings	29 keys
`chart`	-	Pictures and drawings	60 keys
`barcode`	-	Pictures and drawings	30 keys
`math`	`equation`	Pictures and drawings	31 keys
`mermaid`	`diagram`	Pictures and drawings	28 keys
`pdf`	-	Placing existing PDFs	33 keys
`pdf-contact-sheet`	`contact-sheet`	Placing existing PDFs	29 keys
`toc`	`lof`, `lot`, `lol`, `loc`	Navigation and apparatus	30 keys
`toc`	`lof`, `lot`, `lol`, `loc`	Navigation and apparatus	30 keys
`index`	-	Navigation and apparatus	30 keys
`glossary`	-	Navigation and apparatus	24 keys
`bibliography`	`references`	Navigation and apparatus	6 keys
`endnotes`	-	Navigation and apparatus	5 keys

Element	Aliases	Group	Size
`callout`	`admonition`, `note`	Notes and asides	30 keys
`field`	-	Forms	25 keys
`repeat`	-	Data-driven documents	24 keys
`if`	-	Data-driven documents	24 keys
`include`	-	Data-driven documents	23 keys
`marker`	-	Data-driven documents	4 keys
`frame`	-	Story threading	28 keys
`bar`	-	Meters	29 keys

B. Enumerations

Every fixed set of values in the language, gathered in one place.

Key	Accepted values
`align`	`center`, `decimal`, `left`, `right`
`align`	`center`, `end`, `justify`, `left`, `right`, `start`
`align`	`center`, `end`, `start`
`align`	`center`, `left`, `right`
`angleUnit`	`deg`, `rad`
`appendixFormat`	`1`, `A`, `ALPHA`, `I`, `ROMAN`, `a`, `alpha`, `arabic`, `chinese`, `chinese-traditional`, `cjk-decimal`, `decimal`, `hanzi`, `hanzi-traditional`, `hanzi-tw`, `i`, `lower-alpha`, `lower-roman`, `roman`, `upper-alpha`, `upper-roman`
`arrow`	`both`, `end`, `none`, `start`
`as`	`full`, `label`, `number`, `page`, `title`
`axis`	`left`, `right`, `secondary`, `y2`
`binding`	`l2r`, `left`, `ltr`, `r2l`, `right`, `rtl`
`blanks`	`beforeLast`, `end`
`blend`	`color`, `colorBurn`, `colorDodge`, `darken`, `difference`, `exclusion`, `hardLight`, `hue`, `lighten`, `luminosity`, `multiply`, `normal`, `overlay`, `saturation`, `screen`, `softLight`
`blockAlign`	`center`, `left`, `right`
`boxAlign`	`center`, `left`, `right`
`cellVAlign`	`bottom`, `center`, `middle`, `top`

Key	Accepted values
chainFormat	1, A, ALPHA, I, ROMAN, a, alpha, arabic, chinese, chinese-traditional, cjk-decimal, decimal, hanzi, hanzi-traditional, hanzi-tw, i, lower-alpha, lower-roman, roman, upper-alpha, upper-roman
chartType	area, bar, box, boxplot, bubble, candlestick, combo, donut, doughnut, funnel, gauge, heatmap, line, ohlc, pie, polar, radar, sankey, scatter, spider, treemap, waterfall
childOrigin	container, page
cipher	aes, rc4
code	128, 39, 93, auspost, ausredirect, ausreply, ausroute, azrune, aztec, bc412, c25iata, c25ind, c25inter, c25logic, c25matrix, c25standard, cepnet, channel, codabar, codablockf, code11, code128, code128ab, code128b, code16k, code32, code39, code39ext, code3of9, code49, code93, codeone, daft, databar, databarexpanded, databarlimited, datamatrix, dbarexp, dbarexpcc, dbarexpstk, dbarexpstkcc, dbarltd, dbarltdcc, dbaromn, dbaromncc, dbaromnstk, dbaromnstkcc, dbarstk, dbarstkcc, dotcode, dpd, dpident, dpleit, dxfilmedge, ean, ean128, ean13, ean14, ean8, eanx, eanxcc, eanxchk, excode39, fim, flat, gridmatrix, gs1128, gs1128cc, hanxin, hibc128, hibc39, hibcaztec, hibcblockf, hibcdm, hibcmicpdf, hibcpdf, hibcqr, i2of5, intelligentmail, interleaved2of5, isbn, isbnx, itf14, japanpost, kix, koreapost, logmars, mailmark, mailmark2d, mailmark4s, maxicode, micropdf417, microqr, msi, msiplessey, nve18, onecode, pdf417, pdf417comp, pharma, pharmatwo, planet, plessey, postnet, pzn, qr, qrcode, rm4scc, rmqr, royalmail, rss14, telepen, telepennum, ultra, ultracode, upca, upcacc, upcachk, upce, upcecc, upcechk, upnqr, upus10, uspsimail, vin
collation	pinyin, strokes, unicode
colorMode	CMYK, RGB, cmyk, rgb
colorProfiles	embed, ignore
declarationGate	off, strict, warn
direction	auto, l2r, left-to-right, ltr, r2l, right-to-left, rtl
duplex	longEdge, shortEdge
edge	bottom, left, right, top
endnoteMarker	numeric, symbols
extent	block, line
field	button, check, checkbox, choice, combo, list, radio, sig, signature, text
fit	actual, fit, fitHeight, fitWidth
fit	contain, cover, fill, none

Key	Accepted values
fit	contain, cover, stretch
float	center, left, right
footnoteMarker	numeric, symbols
footnoteRestart	document, section
format	1, A, ALPHA, I, ROMAN, a, alpha, arabic, chinese, chinese-traditional, cjk-decimal, decimal, hanzi, hanzi-traditional, hanzi-tw, i, lower-alpha, lower-roman, roman, upper-alpha, upper-roman
format	auto, flate, jpeg, png
groupBlend	color, colorBurn, colorDodge, darken, difference, exclusion, hardLight, hue, lighten, luminosity, multiply, normal, overlay, saturation, screen, softLight
justify	center, left, right
keyLength	128, 256, 40
labelPos	after, before, none
layer	over, under
layout	oneColumn, single, twoColumnLeft, twoColumnRight, twoPageLeft, twoPageRight
leader	dash, dot, dots, hyphen, none, underline, underscore
legendPos	bl, bottom, br, left, right, tl, top, tr
lineCap	butt, round, square
lineJoin	bevel, miter, round
marker	1, A, I, a, bullet, dash, decimal, i, lower-alpha, lower-roman, upper-alpha, upper-roman
mode	attachments, fullscreen, none, outlines, thumbs
mode	dual, outline, solid
on	all, first, rest
orientation	landscape, portrait
pages	all, even, first, odd, rest
pdfua	1, 2
position	angle, bottom, center, diagonal, diagonal-down, top
relationship	Alternative, Data, EncryptedPayload, FormData, Schema, Source, Supplement, Unspecified
role	aside, blockquote, div, none, section
rotate	0, 270, 90

Key	Accepted values
scale	actual, fit
scope	document, group, section
sectionTags	auto, none, pages
shape	circle, ellipse, line, path, polygon, rect
shape	circle, ellipse, path, polygon, rect
sheet	11x17, 12x18, A3, A4, Ledger, Letter, Tabloid, auto
side	"", both, recto, verso
side	inside, left, outside, right
sideHeadSide	inside, left, outside, right
sidenoteSide	inside, left, outside, right
startOn	fromFile, nextAvailable, recto, verso
style	author-year, numeric
tableAlign	center, left, right
textWrap	around, none, topAndBottom
to	below, below-line, bottom-of-column, bottom-of-page, col-bottom, col-top, column-bottom, column-top, in-flow, margin, outside, outside-column, outside-text-frame, page-bottom, page-top, top-of-column, top-of-page
type	admonition, callout, note
type	area, bar, line, scatter
type	background, echo, glow, lift, neon, none, shadow
type	bibliography, references
type	cardGrid, cardgrid
type	code, pre
type	contact-sheet, pdf-contact-sheet
type	diagram, mermaid
type	equation, math
type	gradient, image, none
type	linear, radial
type	loc, lof, lol, lot, toc
type	page-break, pagebreak
underline	double, none, numeric, single

Key	Accepted values
valign	bottom, center, middle, top
valign	bottom, center, stretch, top
variant	caution, danger, important, info, note, success, tip, warning
verticalAlign	bottom, center, justify, middle, top
verticalAlign	bottom, center, middle, top
weight	bold, normal, regular
writingMode	horizontal, horizontal-tb, inherit, lr-tb, tategaki, tb-lr, tb-rl, vertical, vertical-lr, vertical-rl, yokogaki, 横排, 直排, 縦書き
writingMode	horizontal, horizontal-tb, lr-tb, tategaki, tb-lr, tb-rl, vertical, vertical-lr, vertical-rl, yokogaki, 横排, 直排, 縦書き

Glossary

block One element in the document flow — a paragraph, table, image and so on. The unit the engine lays out and paginates.

counting pass A layout pass that measures without drawing, so the engine can resolve page numbers that depend on the pagination they change.

flow The sequence in which blocks are placed down the page, and onward to the next page or column when the current one fills.

run A styled piece of a paragraph or heading. One paragraph is built from any number of runs, each with its own style, link or footnote.

section A stretch of the document with its own page setup — size, orientation, margins, columns and folio style.

spec The JSON file describing a document.

tagged PDF A PDF carrying a logical structure tree, so assistive technology can read it in a meaningful order.

Index

A

alt 12, 19, 27, 30, 33, 40, 46, 52, 57, 62, 69, 72, 74–75, 77 80, 83, 87, 89–90, 93–94, 100–101, 104, 107, 110, 112 116, 118, 121, 123–124, 127, 129, 132, 138, 144, 149 155, 162, 168, 174, 179, 183, 189, 194, 203–205

appendix 9, 14–16, 24, 30, 37, 43, 49, 54, 59, 64, 68, 76 84, 90, 101, 108, 113, 118, 124, 130, 135, 141, 147 152, 158, 160, 166, 171, 176, 180–181, 187, 192

B

barcode 107, 235

barcode 107, 235

blocks 1, 5, 14, 22, 26, 35, 37, 42, 44, 64, 160, 170–172 175–176, 184, 197–200, 202, 208, 215, 219–220, 223 225, 227–228, 230

border 5, 9, 12, 15, 18, 24, 26, 30, 33, 35, 37, 39, 43, 45 49, 51, 54, 56, 59, 61, 64, 68–72, 76, 79, 83, 85–86, 90 92, 100, 104, 108, 110, 113, 115, 118, 120, 124, 126 130, 132, 135, 138, 141, 143, 147, 149, 152, 154 157–158, 160, 162, 166, 168, 171, 173, 176, 178, 181 183, 187, 189, 192, 194, 197–198, 200, 205, 207–209

C

callout 19, 39–40, 82, 158, 236, 239

caption 14, 22, 28, 30, 33, 35, 40, 42, 47, 53, 58, 63, 69 71, 74, 77, 79, 82, 88, 90, 92, 94, 101, 104, 106, 112 117, 122, 124, 126, 128, 134, 139–140, 145, 151, 156 164, 170, 175, 180, 185, 190, 196, 203–205, 212, 217

cardGrid 9, 16, 48, 208, 235, 239

chart 10, 17, 25, 31, 38, 44, 50, 55, 60, 70, 78, 85, 91, 94 100–103, 105–106, 109, 114, 119, 125, 130, 136, 142 148, 153, 161, 167, 172, 177, 182, 188, 193, 220–221 228, 235

color 10, 12, 16, 18, 24, 26, 31–34, 37, 39, 43, 45, 49, 51 54, 56, 58–61, 64, 68–73, 76–77, 79, 81–84, 86, 88, 90 92, 101–105, 108, 110, 113, 115, 118, 120, 124, 126 132, 135, 138, 141, 143, 147, 149, 152, 154, 157–158 160, 162–163, 166, 168, 171, 173, 176, 178, 181, 183 185, 187, 189, 192, 194–195, 197–198, 200–202, 206 208–211, 213, 218, 224, 227–229, 236, 238

columns 7, 9–10, 13–14, 16–17, 19–21, 28, 34–35 41–42, 44, 47, 51, 53, 58, 63–64, 69–70, 72–74, 81, 88 94, 105–106, 111–112, 117, 122, 128, 130, 133, 139 145, 148, 150–151, 156, 163–164, 169, 175, 180, 184 187, 190, 195–196, 198–199, 205, 208, 213, 215, 220 223, 228, 235

columns 7, 9–10, 13–14, 16–17, 19–21, 28, 34–35 41–42, 44, 47, 51, 53, 58, 63–64, 69–70, 72–74, 81, 88 94, 105–106, 111–112, 117, 122, 128, 130, 133, 139 145, 148, 150–151, 156, 163–164, 169, 175, 180, 184 187, 190, 195–196, 198–199, 205, 208, 213, 215, 220 223, 228, 235

container 5, 8–10, 12–13, 15–18, 20–21, 24–28, 30–31 33–47, 49–64, 68–71, 73–81, 83–94, 100–102 104–130, 132–133, 135–136, 138–169, 171–184 186–197, 199–201, 207–208, 230, 235, 237

cross-reference 5, 7, 12, 14–15, 19, 77, 112, 124, 203 205

D

data 8, 11–13, 15, 18, 20, 24, 26–27, 30, 32, 34, 36 39–41, 43, 45–46, 49, 51–52, 54, 56–57, 59, 61–62, 68 71, 73, 75, 78–84, 86–87, 89, 92–94, 100–101 103–105, 107, 109–111, 113, 115–116, 118, 120–121 123, 125–129, 131, 133, 135, 137–140, 143–146 148–150, 152, 154–155, 159, 161–163, 165, 167–171 173–176, 178–179, 181, 183–184, 186, 188–191 193–195, 207–208, 220–221, 224, 229–230, 238

E

equation 112, 114–115, 203–205, 217, 235, 239

F

field 8, 12, 15, 20, 24, 27, 30, 34, 36, 41, 43, 46, 49, 52 54, 57, 59, 62, 68, 73, 75, 78, 80, 82–83, 87, 89, 93 100, 105, 107, 111, 113, 116, 118, 121, 123, 125 127–129, 133, 135, 138, 140, 144, 146, 150, 152, 155 157, 159, 163–165, 167, 169, 171, 174, 176, 179, 181 184, 186, 189, 191, 195, 202, 204–205, 207, 212 227–228, 236–237

float 10, 13, 16–17, 20–21, 25, 28, 31, 34, 37–38, 41–42 44, 47, 50, 53, 55, 58, 60, 63, 70, 73–74, 78, 81, 85, 88 91, 94, 102, 105, 109, 111, 114, 117, 119, 122, 125 128, 130, 136, 139, 142, 145, 148, 150, 153, 156, 161 163, 167, 169, 172, 175, 177, 180, 182, 184, 188, 190 193, 195, 199, 215, 238

font 5, 12, 14, 17, 20–21, 27, 31, 34, 41, 46, 52, 57, 62, 73 80, 87, 93, 105, 111, 116, 121, 127, 133, 138, 144, 150 155, 161, 163, 169, 174, 179, 184, 189, 195, 199–202 204, 213, 218, 225–226
footnote 7, 9, 12, 19, 21, 40, 199, 202, 204–205, 208, 215 227, 234
form 6, 10, 12–13, 19, 21, 27–28, 33, 35, 40–41, 46–47 52–53, 57–58, 62–63, 70–72, 74, 80–81, 87–88, 93–94 104, 106, 108, 110, 112, 116–117, 121–122, 127–128 132–133, 138–139, 144–145, 149–151, 155–156, 158 162, 164, 168–169, 174–175, 179–180, 183–184 189–190, 194, 196, 202, 204, 206–207, 212, 219, 225
frame 8, 10–12, 15–19, 21, 24–27, 30–33, 36, 38–40 43–46, 49–52, 54–57, 59–62, 68–72, 75–80, 83–87 89–93, 100–104, 107–110, 113–116, 118–121 123–127, 129–132, 135–138, 140–144, 146–149 152–155, 159–162, 165–168, 171–174, 176–179 181–183, 186–194, 197, 200–201, 215, 221, 230, 236

G

glossary ii, 151, 153, 156, 225, 235, 241
gradient 82, 84, 210–211, 228, 239

H

heading 1–2, 5, 13–16, 18–21, 28, 35, 41, 47, 53, 58, 63 74, 81, 88, 94, 105–106, 112, 117, 122, 128, 133, 137 139, 142–143, 145, 148, 150–151, 156–158, 163–164 169, 175, 180, 184, 190, 196–204, 207–208, 214, 217 224, 230, 235

I

image 8, 11, 15, 18, 22, 24, 26, 30, 32, 36, 39, 43, 45, 49 51, 54, 56, 59, 61, 68, 71, 74–81, 83, 86, 89, 92, 100 103, 107, 109, 113, 115, 118, 120, 123, 126, 129, 131 135, 137, 140, 143, 146, 148, 152, 154, 159, 161, 165 167, 171, 173, 176, 178, 181, 183, 186, 188, 191, 193 204–205, 216–218, 220, 228–229, 235, 239
include 12, 19, 27, 33, 40, 46, 52, 57, 62, 72, 80, 87, 93 104, 110, 116, 121, 127, 132, 138, 144, 149, 155, 162 168, 174, 179–180, 183, 189, 194, 236
index ii, 128, 139, 144–145, 147–148, 150, 171, 199 221–222, 225, 235, 242

K

keepTogether 35, 39, 161

L

list 1, 9, 12, 14, 18–19, 22, 26–28, 30, 33, 35, 40, 42 46–47, 52–53, 57–58, 62–63, 69, 72, 74, 77, 80, 82 87–88, 90, 93–94, 101, 104, 106, 110, 112, 116–117 121–122, 124, 127–128, 132, 134, 136–140, 142 144–145, 149, 151, 155–157, 162, 164, 168, 170 174–175, 179–180, 183, 185, 189–190, 194, 196–197 199–201, 212, 220, 224–225, 235, 237
listing 10, 17, 25, 28, 30–31, 34, 38, 44, 50, 55, 60, 70, 78 85, 91, 102, 109, 114, 119, 125, 130, 136, 142, 148 153, 161, 167, 172, 177, 182, 188, 193, 217

M

margins 1, 213–215, 227–228

N

numbering 14, 18–19, 30, 33, 71, 79, 90, 92, 101, 104 112, 115, 126, 203–204, 213–214, 216–217, 223, 227 229, 233

P

padding 5, 13, 16, 20–21, 28, 35, 37, 39, 41–42, 47, 53 58, 63, 69, 74, 81, 88, 94, 106, 112, 117, 122, 128, 133 139, 145, 151, 156, 164, 169, 175, 180, 184, 190 196–197, 200, 203, 205, 209, 220
page break 1, 35, 70
paragraph ii, 1, 7, 9–10, 13–14, 16, 18–19, 21, 28, 35, 41 47, 53, 58, 63, 74, 81, 88, 94, 106, 112, 117, 122, 128 133, 139, 145, 151, 156, 163–164, 169, 175, 180, 184 190, 196–202, 207–208, 220, 227, 235
PDF/A 213, 224, 228, 230
PDF/UA 12, 19, 27, 33, 40, 46, 52, 57, 62, 72, 74, 80, 87 93, 100, 104, 110, 116, 121, 127, 132, 138, 144, 149 155, 162, 168, 174, 179, 183, 189, 194, 212–213 219–220, 224, 229–230

R

repeat 8, 12, 15, 19, 24, 27, 30, 33, 36, 40, 43, 46, 49, 52 54, 57, 59, 62, 64, 68, 70, 72, 76, 80, 83, 87, 89, 93, 100 104, 107, 110, 113, 116, 118, 121, 123, 127, 129, 132 135, 138, 141, 144, 146, 149, 152, 155, 160, 162, 166 168, 170–172, 174, 176, 179, 181, 183, 186, 189, 191 194, 236
rule 37, 58, 61, 70, 83, 198–200, 202, 221, 235
runs 7, 9, 12, 14, 19, 21, 26, 30, 69, 77, 90, 101, 103, 124 145, 147, 156, 199, 202, 208, 220, 229–230

S

sections 1, 139, 144, 150, 170, 223, 228, 230–231

shape 1, 10, 12, 16–17, 19, 21, 25, 27, 31, 33, 35, 38, 40 44, 46, 50, 52, 55, 57, 60, 62, 69, 72, 77, 80, 82–85 87–88, 90–91, 93, 101–102, 104, 108, 110, 114, 116 119, 121, 124, 127, 130, 132, 136, 138, 141, 144, 147 149, 153, 155, 160, 162, 166, 168, 172, 174, 177, 179 182–183, 187, 189, 192, 194, 205, 209, 220, 228, 235 239
signature 48, 164–165, 204, 212, 237
spacer 53, 235
story 185–187, 190, 236
style 1–2, 5, 13–14, 19–21, 26, 37, 39, 88, 137, 142 156–157, 197, 201–205, 216–217, 220, 224–225, 227 230–231, 239
svg 12, 18–19, 26–27, 33, 39–40, 45–46, 51–52, 56–57 61–62, 64, 71–72, 79–80, 83–84, 86–88, 91–94, 104 110, 112, 114–117, 119–122, 126–127, 132, 138 143–144, 149, 154–155, 157–158, 162, 168, 173–174 178–179, 183, 189, 194, 204–205, 235

T

table ii, 1–2, 9–10, 14, 16–17, 19, 21–22, 25, 31, 34, 38 44, 50, 55, 60, 64, 68–74, 78, 85, 91, 102, 108–109 114, 119, 125, 130, 134, 136, 140, 142, 146, 148, 153 161, 167, 172, 177, 182, 188, 193, 197, 199–201, 208 217, 219–220, 222, 224, 228, 235
tagged ii, 1–2, 10, 12, 14, 16, 18–19, 21, 24, 27, 31, 33 37, 40, 43, 46, 49, 52, 54, 57, 59, 62, 69–70, 72, 75, 77 80, 83–84, 87, 89–90, 93, 100–101, 104, 108, 110, 113 116, 118, 121, 123–124, 127, 129, 132, 135, 138, 141 144, 147, 149, 152, 155, 158, 160, 162, 166, 168, 171 174, 176, 179, 181, 183, 185, 187, 189, 192, 194 204–205, 213, 222–223, 226, 228–230, 232
thread 186–187, 230
toc 19, 40, 134, 140, 223, 225, 235, 239

W

watermark 215, 217–218