Meander User guide

Abstract

Meander implements a content layout algorithm to provide text threading (when text from one box spills into a different box if it overflows), uneven columns, and image wraparound.

Feature requests

For as long as the feature doesn't exist natively in Typst (see issue: github:typst/typst #5181), feel free to submit test cases of layouts you would like to see supported by opening a new issue.

Versions

- dev
- 0.2.0 (latest)
- 0.1.0

adipiscing elit, sed do eiusmod tempor.

Lorem ipsum dolor sit anet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magnam aliquam quaerat voluplatem. Ut enim aeque doleamus animo, cum corpore dolemus, fieri tamen permagna accessio potest, si aliquod aeternum et infinitum impendere maulum nobio spinemur. Quod idem licet transferre in voluplatem, ut postea variari voluplata distinguique poosit, augeir amplificarique non possit. At eliam Athenis, ut e patre audiebam facete et urbane Stoicos irridente, statua est in quo a nobis pilnosophia defensa et collaudata est, cum id, quod maxime placeat, facere possimus, omnis voluptas assumenda est, omnis dolor repel-lendus. Temporibus autem quibasdam et aut officias debits aut rerum necessitatibus saepe eveniet, ut et voluptates repudiandae sint et molestiae non recusandae. Itaque earum rerum defuturum, quas natura non depravada desiderat. Et quem ad me acederis, salito: chaere; Inquam, Tile! Hiores, turna omnis chorusque: 'chaere, Tile!' hine hostis mi Albucius, hine inimicus. Sed iure Mucius.

Lipsum

Lorem ipsum dolor sit amet, consectetur adipiscing eliit, sed do eiusmod tempor incididunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim acque doloamus animo, cum corpore dolemus, feri tamen permagna accessio potest, si alquod acternum et infinitum impendere malum nobis opinamur. Quod idem licet transferre in voluptatem, ut postea variari voluptas distinguique possit, augeri amplificarique non possit. At etiam Athenis, ut e pater audiebam faceté et urbane Stoicos irridente, statua est in quo a nobis philosophia defensa et collaudata est, cum id, quod maxime placest, facere possimus, omnis voluptas assumenda est, omnis dolor repellendus. Temporbrus autem quabusdam et aut officis debitis aut rerum mecessitatibus suspe eveniet, ut et voluptates repudiandae sint et molestiae non recusandae. Ilaque earum rerum defatturum, quas natura non depravata desiderat. El quem ad me accedis, saluto: 'chaere; inquam, Titel' lictores, turma omnis chorusque: 'chaere, Titel' hinc hostis mi Albucius, hin inimieus. Sed iure Mucius. Ego autem mirari satis non queo unde hoe sit tam insolens domesticarum rerum fastidium. Non est omnino hic docendi locus; sed ita prorsus existimo, neque cum Torquatum, qui hoe primus cognomen invenert, aut torquem illum hosti detravisse; ut aliquam ex o o est consecutus? - Laudem et caritatem, quae sunt vitae sine metu degendae praesida firmissima ex o ost consecutus? - Laudem et caritatem, quae sunt vitae sine metu degendae praesida firmissima ex o ost consecutus? - Laudem et caritatem, quae sunt vitae sine metu degendae praesida firmissima ex o ost consecutus? - Laudem et caritatem, quae sunt vitae sine metu degendae praesida firmissima ex o ost consecutus? - Laudem et caritatem, quae sunt vitae sine metu degendae praesida firmissima ex o ost consecutus? - Laudem et caritatem, quae sunt vitae sine metu degendae praesida firmissima ex o ost consecutus? - Laudem et caritatem, quae suntilati nu, geometriaquice praesidae, con muniquam patavises, si a Polyaeno, familiari suo, geomet

Contents

I	Quick start	2
	Showcase	
	Understanding the algorithm	
	Advanced techniques	
	Modularity (WIP)	
	Style-sensitive layout	
	Module details	

I Quick start

The main function provided is #meander.reflow, which takes as input a sequence of "containers", "obstacles", and "flowing content", created respectively by the functions #container, #placed, and #content. Obstacles are placed on the page with a fixed layout. After excluding the zones occupied by obstacles, the containers are segmented into boxes then filled by the flowing content.

I.a A simple example

Below is a single page whose layout is fully determined by Meander. Currently multi-page setups are not supported, but this is definitely a desired feature.

Loren ipsum dolor sit amet, consecteur adipiscing ellit, sed do ciuamod tempor incididant ut labore et dolore magam aliquum quaerai volupitatem. Ut enim acque dolorama amino, cum corpore dolerus, fori tamen permaga accessio potes, si aliquod aeternum et infinitum inpendere malum nobis sopinemus. Quod idem licet transferre in volupitatem, ut postes variari volupitas distinguique possis, augeri amplificarique non possis. A.I. Lorem ipsum dolor sit amet, consectetur adipiscing elli, sed do eiusmod tempor incidiatum ti tabore et dolore magama allquam quaerat volupitatem. Ut enim acque dolorama animo, cum corpore dolemus, fieri tamen permagna accessio potest, si aliquod aeternum et infinitum impendere malum nobis opinemus. Quod idem licet transferre in volupitatem, ut postes variari volupitad distinguique possis, augeri amplificarique non possis. At etiam Athenis, ut e patre audiebama facete et urban civa toricos irridente, statua est in quo a nobis philosophia defensa et collaudata est, cum id, quod maxime placeal, facere possimus, omnis volupitas assumenda est, omnis dolor repellentus. Emporptibus autem gubusdam et aut officis debitis aut rerum necessitatibus saepe evenite, ut et volupitates repudiandae sint et molestiae non recusandae. Itaque earum rerum defuturum, quas natura non deparvata desideral. Et quem ad me accedis, salutor 'chaere', inquam, 'Tile' lictores, turma omnis chorusque' chaere, Tile' hinc hostis mi Albucius, hinc inimicus. Sed iure Mucius. Ego autem mirari saits non queo unde hose sit tam insolens domesticarum rerum fastidium. Non est omnino hic docendi locus; sed its prorsus existimo, neque eum Torquatum, qui hoc primus cognomen invenent, aut torquem illum hosti detrasisse, ut aliquam ex co est consecutus' – Laudem et caritatem, quae sunt vitae sine netu degendae praesida firmissima. – Filiam morte multavit.

— si sine causa, nollemo, familiari suo, generitera dispere desida mirissima. – Filiam morte multavit.

— si sine causa, nollemo, familiari suo, generitera dispere malusse, utatum enim es

Meander is expected to respect the majority of styling options, including headings, paragraph justification, font size, etc. Notable exceptions are detailed in Section VI. If you find a discrepancy make sure to file it as a bug report if it is not already part of the known limitations.

Note: paragraph breaks may behave incorrectly. You can insert vertical spaces if needed.

I.b Multiple obstacles

#meander.reflow can handle as many obstacles as you provide (at the cost of potentially performance issues if there are too many, but experiments have shown that up to ~100 obstacles is no problem).

```
#meander.reflow({
  import meander: *
  // As many obstacles as you want
  placed(top + left, my-img-1)
  placed(top + right, my-img-2)
  placed(horizon + right, my-img-3)
  placed(bottom + left, my-img-4)
  placed(bottom + left, dx: 32%,
         my-img-5)
  // The container wraps around all
  container()
  content[
    #set par(justify: true)
    #lorem(600)
  1
})
```

Lorem ipsum dolor sit amet, consecteur adjuliaris de de ciusmod tempor inci-didunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim aeque doleamus animo, cum corpore dolemus, fieri tamen permagna accessio potest, si aliquod aetermum et infinitum impendere malum nobis opinemur. Quod idem licet transferre in volupatem, ut postes avariar voluptas distinguisue possit, augeri amplificarique non possit. At etiam Athenis, ut e patre audiebum facet et urbane Stoicos irridente, statua est in quo a nobis philosophia distinguisue possit, augeri amplificarique non possit. At etiam Athenis, ut e patre audiebum facet e utrabane Stoicos irridente, statua est in quo a nobis philosophia dam et aut officiis debitis aut rerum necessitatibus saepe evenie, ut et volupatem facet e utrabane Stoicos irridente, statua est in quo a nobis philosophia dam et aut officiis debitis aut rerum necessitatibus saepe evenie, ut et volupatem repudianta sint et molestiae non recusandae. Itaque earum rerum defiturum, quas natura non depravata desiderat. Et quem ad me accedis, saluto 'chaere,' inquam. 'The' lictores, turtum onnis chorosgue: 'chaere. The' hinc hostis mi Albucius, hine timinicus. Sed ture Mucius. Ego autem mirari satis non queo unde hoc sit tam insolens domesticamum rerum fastitum. Non est omnino hic docendi locus, sed ita prorsus existimo, neque eum Ornquatam, quitor primus cognomen invenent, aut torquem illum hosti de-traxisse, ut alquum et cor eo est consecutus? Laudem et caritatem, quae sunt vitae sine metu degendae praeratis firmissisma — Fillium morte multavit. — Si sine causa, nollem me ab eo delectuar, quod ista Platonia, Aristoteli, Theophrasti orationis oranementa neglezerit. Nam illud quidem physici, credere aliquid esse minimum, quod profecto nunquam puturissisma. Aribitum morte multavit. — Si sine causa, nollem me ab eo delectuari, quod ista Platonia, Aristoteli, Theophrasti orationis oranementa neglezerit sun designationis oranementa neglezerit, aud oranementa neglezerit sun designationis oranementa neglezer

I.c Columns

In order to simulate a multi-column layout, you can provide several container invocations. They will be filled in the order provided.

```
#meander.reflow({
   import meander: *
   placed(bottom + right, my-img-1)
   placed(center + horizon, my-img-2)
   placed(top + right, my-img-3)

// With two containers we can
// emulate two columns.
   container(width: 55%)
   container(align: right, width: 40%)
   content[#lorem(600)]
})
```

Loren ipsum dolor att amet, consectetur aulpracing elit, sed do etumost tempor incidionat at labora et dolore magnam aliquam quaestar voluptatem. (It enim seque doloamus antino, accesso polect, si aliquod atterm est inflationation imperadere malum mobio opinemus. Quod dieral incidentational control designation of the polect transferre in voluptatem, to potest a valiaria voluptas distinguispe posat, augeri ampilitation imperadere malum mobio opinemus. Quod dieral incidentation of the polect transferre in voluptatem, postes avairative voluptas distinguispe posat, augeri ampilitation est composition de la respectation de la respectatio

II Showcase

A selection of nontrivial examples of what is feasible.

seminar.sty
is a LaTeX style
for typesetting slides
or transparencies, and accompanying notes. Here are
some of its special features: It is
compatible with AmS-LaTeX, and you
can use PostScript and AmS fonts. Slides can
be landscape and portrait. There is support for
color and frames. The magnification can be changed
easily. Overlays can be produced from a single slide environment. Accompanying notes, such as the
text of a presentation, can be put outside the
slide environments. The slides, notes or both
together can then be typeset in a variety of formats.

Lorem ipsum dolor sit amet, consectetur adipiscing ellit, sed do eiusmod tempor incididunt ut labore et dolore magnam aliquam quaerat voluptatem. O'e erim acque doleamus antino, cum corpore dolemus, fieri tamen permagna accessio potest, si aliquod acternum et infintum impendere maluro nobio pionemu. Quoi deime lice transferre in voluptatem, qui potest variari voltpate stituguique possi, augeri amplificarique non possit. At etiam Athenis, ut e patre audiebam facete et urbane Stoicos irridente, status est in quo a nobis philosophia defense et colludate act, cum di, qued mantier placent, facere possimus, omnivo voltpates assumende et, ormis dolo repellendus. Temporibos autem quibusdam et aut officiis debits aut rerum nocessitatibus sespe eveniet, ut et voluptates repudandae inte et molestica propriosa controlica debits aut rerum nocessitatibus sespe eveniet, ut et voluptates repudandae inte et molestica propriosa controlica debits autoritate in no depevate desidente. Et quem an fene accedia, saluto charec, inquam, Tille "Bucteves, tuma omnis chorusque" charec. Title line hosts un Albecius, hite inministra. Sed um Nethonis. Sego interna internation of the controlica debits autoritate debit autoritate debits autoritatione la debit autoritate debits autoritatione la debits autoritatione la deb

examples/5181-a/main.typ
Inspired by github:typst/typst #5181 (a)

examples/5181-b/main.typ
Inspired by github:typst/typst #5181 (b)

III Understanding the algorithm

The same page setup as the previous example will internally be separated into

- obstacles my-img-1, my-img-2, and my-img-3.
- containers #(x: 0%, y: 0%, width: 55%, height: 100%) and #(x: 60%, y: 0%, width: 40%, height: 100%)
- flowing content #lorem(600).

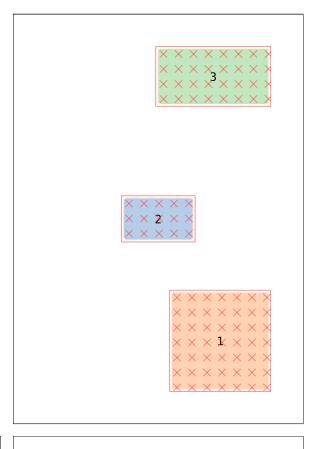
Initially obstacles are placed on the page (\rightarrow) . If they have a boundary parameter, it recomputes the exclusion zone.

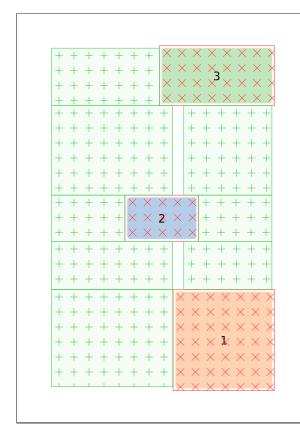
Then the containers are placed on the page and segmented into rectangles to avoid the exclusion zones (\downarrow) .

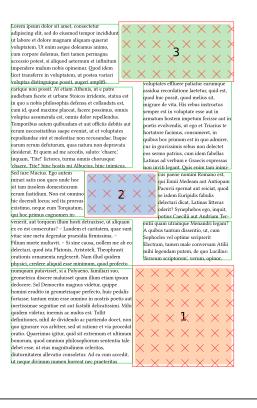
Finally the flowing content is threaded through those boxes (\(\)), which may be resized vertically a bit compared to the initial segmentation.

The debug views on this page are accessible via #meander.regions and

#meander.reflow.with(debug: true)





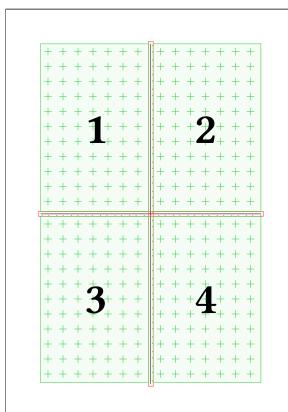


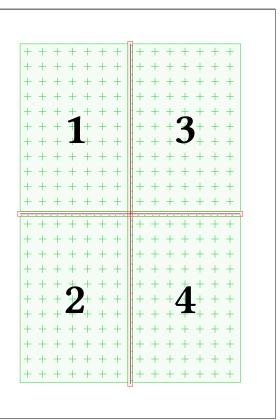
The order in which the boxes are filled is in the priority of

- · container order
- $top \rightarrow bottom$
- left \rightarrow right

which has implications for how your text will be laid out. Indeed compare the following situations that result in the same boxes but in different orders:

```
#meander.regions({
                                            #meander.regions({
  import meander: *
                                              import meander: *
  placed(center + horizon,
                                              placed(center + horizon,
    line(end: (100%, 0%)))
                                                line(end: (100%, 0%)))
  placed(center + horizon,
                                              placed(center + horizon,
    line(end: (0%, 100%)))
                                                line(end: (0%, 100%)))
  container(width: 100%)
                                              container(width: 50%)
                                              container(align: right, width: 50%)
})
                                            })
```





And even in the example above, the box 1 will be filled before the first line of 2 is used. In short, Meander does not "guess" columns. If you want columns rather than a top-bottom and left-right layout, you need to specify them.

IV Advanced techniques

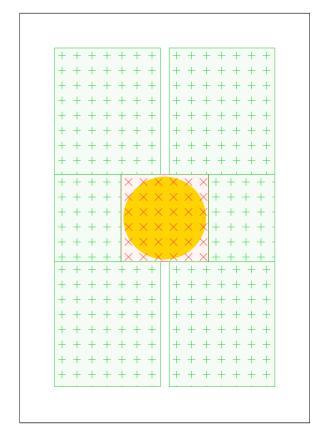
Although Meander started as only a text threading engine, the ability to place text in boxes of unequal width has direct applications in more advanced paragraph shapes. This has been a desired feature since at least issue #5181.

Even though this is somewhat outside of the original feature roadmap, Meander makes an effort for this application to be more user-friendly, by providing functions to redraw the boundaries of an obstacle. Here we walk through these steps.

Here is our starting point: a simple double-column page with a cutout in the middle for an image.

```
#meander.reflow({
   import meander: *
   placed(center + horizon)[#circle(radius: 3cm, fill: yellow)]
   container(width: 48%)
   container(align: right, width: 48%)

content[
    #set par(justify: true)
    #lorem(600)
]
})
```



Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magnan aliquam quaerat voluptatem. Ut enim aeque doleamus arinno, cum corpore dolemus, fieri tamen permagna accessio potest, si aliquo aterimu nei infinitum impendere malum nobis opinemur. Quod idem licet transferre in outputatem, ut postea variari voluptate distinguique possit, augeri amplificarique non possit. At etiam Athenis, ut e patre audiebum facete et urbane Siotoos irridente, statua est in quo a nobis philosophia defensa et collaudata est, cum id, quod maxime placeat, facere possimus, comitis voluptates assumende est, ormits dolor repellendus. Temporibus autem quibusdam et aut officiis debitis aut rerum necestutibus saepe eveniet, ut et voluptates repudiandae sint et molestate non recusandae Itaque earum rerum defurum, quas natura non deprevavata desiderat. Et quem ad me accedis, sultot chaere, in quam, 'Titef' lictores, turma comisi chorvasque: 'haere, 'Titef' hinc hostis mi Albucius, hinc iminicus. Sed iure Mucius. Ego autem mitari satis non que ou ude hos est tam insolens domesticarum rerum fastidium. Non est omnition bic docendi locus; sed ita prorsus existimo, neque eum Torqua- tum, qui hoe primus cognomen invenerit, aut torquem illum hosti detraxises, ut aliquam ex eo est consecutus?' – Laudem et cariatem, quae sunt viae sine metu degendae praesidia firmissima. – Filium morte multavit. – Si sine causas, nollem ma bes o delectari, quod sta placinia, viatoria, quod profecto numquam putavisset, si a Polyaeno, familiar suo, goometrica discere maluisset quam illum etiam ispum dedocere. Sol Democrito magnus videtur, quipe homini erudito in geometriaque perfecto, huic pedalis fortassismi. Mihi quidem videtur, inemis ac mulus est. Toliit definitiones, nihil de dividendo ac partiendo docet, non quo ignorare mitaria delectaria et au discontinum monitoria contratue est aut fastidii delicatissim. Mihi quidem videtur, inemis ac mulus est. Toliit definitiones, nihil de dividendo ac partiendo docet, non

Meander sees all obstacles as rectangular, so the circle leaves a big ugly square hole in our page.

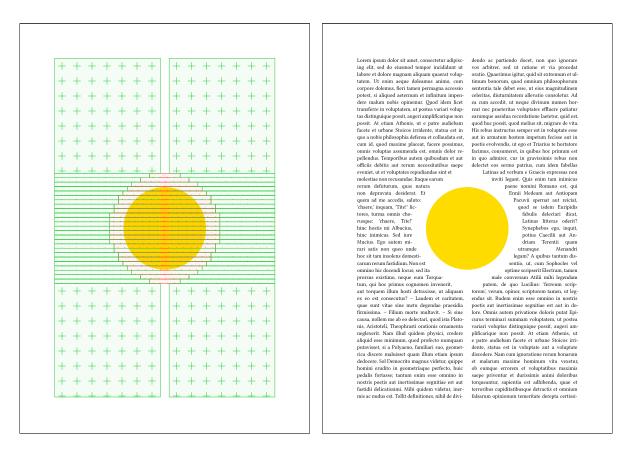
Fear not! We can redraw the boundaries. #meander.placed accepts as parameter boundary a sequence of box transformers to change the way the object affects the layout. These transformations are normalized to the interval [0,1] for convenience. The default boundary value is #contour.margin(5pt).

#meander.contour.grid is one such redrawing function, from $[0,1] \times [0,1]$ to bool, returning for each normalized coordinate (x, y) whether it belongs to the obstacle.

So instead of placing directly the circle, we write:

```
#meander.reflow({
  import meander: *
  placed(
    center + horizon,
    boundary:
      // Override the default margin
      contour.margin(1cm) +
      // Then redraw the shape as a grid
      contour.grid(
        // 25 vertical and horizontal subdivisions (choose whatever looks good)
        div: 25,
        // Equation for a circle of center (0.5, 0.5) and radius 0.5
        (x, y) \Rightarrow calc.pow(2 * x - 1, 2) + calc.pow(2 * y - 1, 2) \Leftarrow 1
      ),
    // Underlying object
    circle(radius: 3cm, fill: yellow),
  )
  // ...
})
```

This results in the new subdivisions of containers below.



This enables in theory drawing arbitrary paragraph shapes. If your shape is not convenient to express through a grid function, here are the other options available:

• $vert(div: _, fun):$ subdivide vertically in div sections, then fun(x) = (top, bottom) produces an obstacle between top and bottom.

- height(div: _, flush: _, fun): subdivide vertically in div sections, then fun(x) = (anchor, height) produces an obstacle of height height, with the interpretation of anchor depending on the value of flush:
 - if flush = top then anchor will be the top of the obstacle;
 - if flush = bottom then anchor will be the bottom of the obstacle;
 - if flush = horizon then anchor will be the center of the obstacle.
- horiz: a horizontal version of vert.
- width: a horizontal version of height.

Reminder: all of these functions operate on values normalized to [0, 1]. See some examples below.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magram aliquam quaerat voluptatem. It enim acque dolarnus animo, cum corpore dolernus, fieri tamen permagna accessio potest, si aliquoud aeterum en infinitum imprudere madum nobis opiniemus. Quod idem licet transferre in voluptatem, ut postea variari voluptas distinguique possit, augeri amplificarique non positi. At etiam Althenis, ut e patre audiebam facet et urbame Stotos irridente, status est in quo a nobis philosophia detensa et collaudata est, cum ali, quod manime placest, facere possituse, omnius voluptas assumenda est, cumais dolor repellendus. Temporibus autem quibusdam et auto officias debits aut terum necesstalubas seape evente, at ev toutpates regulanduse faire et molestate non recusandae, fisaque carum rerum defuturum, qua natura non depravata desiderat. Et quem ad me accedis, saluto: chaere, fiquam. Tirle Hucturum, qua matura non depravata desiderat. Et quem ad me accedis, saluto: chaere, fiquam. Tirle Huctures, turma omus chorosague: chaere, Tirle hine bostis mi Albucius, hine inimicus. Sed ture Mucius. Ego autem mirara satis non queo unde hoc sit tam insolens domesticarum rerum fastidium. Non est orumino in docendi locus; sed lu prorusse existino, neque eum Torquatum, qui hoc primus cognomen invenerit, aut torquem libum hosti detrasisse, via aliquam ex co est consecuturis — Laudem et cartalentu, quae sunt viae sine metu degendae praesida firmitsisma.

— Filium morte muliavit. — Si sine causa, nollem me ab eo detectari, quod sista Piatonis, Aristoteli, Theophrasti orationis orammenta negleserit. Num illud quidem physic, redere aliquide ses minimum, quod profecto nunquam putavisset, si a Polyaeno, familiari suo, geometrica discere maluisset quam illum etiam ipsum deolorer. Sol Democrito magnus videru, gruppe homini erudio in geometraque perfecto, huic pedalis fortases; tatutum enim esse monito in nostris poetis aut inertissimae seguitiae est aut fastidi deliciassismi. Mihi quidem viderui, i

```
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ul labore et dolore magnam alequam quaerat voluptatem. Ut enim aeque doleamus animo, cum corpore dolemus, fieri tamen permagna accessio potest, si aliquod aeternum et infinitum impendere malum nobis opinemur. Quod idem licet transferre in voluptatum, ut posteav variari voluptas distingiuge possit, augeri amplificarique non possit. At etiam Athenis, ut e patre audicbam facete et urbane Stoicos irridente, statua est in quo a nobis philosophia defensa et collaudata est, cum id, quod maximis placeaf, facere possimus, omnis voluptas assumenda est, omnis dolor repellendus. Temporibus autem quibudam et aut officia debits aut remn necessitatibus assec e ceita, et a volviptates repudiandas aint et molestiae non recussandae. Haque earum rerum defuturum, quas natura non depravata desiderat. Et quem ad me accedis, saluto: 'chaere; inquam. Title' lictore, turna omnis chorsuque c'haere. Ell'er hinc hostis mi Albucius, hinc inimicus. Sed iure Mucius. Ego autem mirari satis non quec unde hoe sit tam insolens domesticamu rerum fastidium. Non est omnion in bio doedli ous; sed ita prorsus existimo, neque eum Torquatum, qui hoe primus ognomen invenerit, aut torquem illum obsti defraxisse, ut aliquam ex co est consecutus? — Laudem et cariatem, que sunt vitas im metu degendae presidia firmissima.

— Filium morte multavit. — Si sine causa, nollem me ab eo delecturi, qud ista Platonis, Aristoteli, Theophratis orianis orianenta negleeerii. Nam illud quidiem plysici, credere aliquide ses minimum, quod profecto numquam putavisset, si a Polyaeno, familiari suo, geometrica discere maluiset quam ultum etiam ipsum dedocere. Sol Demecrito mongsus videtur, quiepe homini erudiqui esse minimum, quo demonifi Romanos est, qui separa attra rietait, quod est demonitari reruiti quodus tam reruiti diest, altinas lipetium ceita di stati dieletaissima. Miki quidem videturi, mermia ae mudus est. Tolli definitiones, nihil de dividendo ac partiemdo docet, non quo igno
```

```
#meander.reflow({
  import meander: *
  placed(right + bottom,
    boundary:
      // The right aligned edge makes
      // this easy to specify using
      // `horiz`
      contour.horiz(
        div: 20,
        // (left, right)
        y => (1 - y, 1),
      // Add a post-segmentation margin
      contour.margin(5mm)
  )[...]
  // ...
})
```

```
#meander.reflow({
  import meander: *
  placed(center + bottom,
    boundary:
      // This time the vertical symetry
      // makes `width` a good match.
      contour.width(
        div: 20,
        flush: center,
        // Centered in 0.5, of width y
        y => (0.5, y),
      ) +
      contour.margin(5mm)
  )[...]
 // ...
})
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eissmod tempor incididunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim aeque dolorams animo, cum corpore dolerams, ferit tamen permagna accessio potest, si aliquo da deternum et infinitum impendere malum nobis opinemur. Quod idem lieet transferre in voluptatem, ut postea variari voluptas distinguispe possit, augeri amplificacique non possit. At eliam Athenis, ut e pater audieham facete et urbane Stocios irridente, status est in quo a nobis philosophia defensa et collaudata est, cum id, quod maxime placest, facere possimus, omnis voluptas assumenta est, omnis dolor repellendas. Temporabus autem quibudam et aut officiis debitis aut rerum necessitatibus saepe eveniet, ut et voluptates repudiandes sint et molestiae non recusandae. Baque earum rerum defuturum, quas natura non depravata desiderat. Et queen and seccelis, saluto: 'chaere; finquam, 'The' lictores, turma omnis chorsouse; chaere. The' hine hostis mi Albucius, hine inimicus. Sed inre Mucius. Ego autem mirari satis non queo unde hoc sit tam insoleno domesticarum rerum fattutium. Non est comino in écoerdo leuxes sed in prorsus existimo, neque em Torquaturu, qui hoc primas cognomen invenerit, aut torquem illum hosti de traxisses, ut aliquam ex co est consecturis. "Laudem et cartafiem, quae sunt vitae sine metu degendae praesidia firmissima. "Filium morte multavit." —Si sine cusus, nollem me ab eo dedectari, quoi tari Palzonia, Aristoticil, Theophrasti orationis oramaenta neglecerit. Nami illud quidem physici, credere aliquid esse minimum, quol perfoeto numquam putu-vises, si a Folyaena, familiari suo, geometreis catecere maluises ecumino in nostris poetis aut inertissimae segnitiae est aut fastidi delicatissimi. Mihi quidem videturi, temris ae nodus est. Tolist dehinitores, mili de dividendo are partiendo docet, non quo ignorare vos arbitrer, sed ut ration et via procedar toria. Quaeritras, diuturnitation ali devidendo are partiendo docet, non quo ignorare vos arbitrer, sed ut r

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magnam aliquum quaerat voluptatem. Ut enim aeque doleamus antimo, cum corpore dolemus, fiert tamen permagna accessio potest, si aliquod ateretumu et infinitum impendere malum nobis opinemus. Quod idem finet transferre in voluptatem, ut postea variari voluptate distinguique possit, auger amplificarique non possit. At etiam Aliens, sit e patre audiebam facete et rabame Stoicos irridente, statua est in que a nobis philosophia defensa et coliaudatu est, cum it quod maxime placeat, facere possimus, omnis voluptas assumenda est, omnis dolor repellendas. Temporitum auteru quibasdam et aut officiti debitas at rerum nescatatibus saepe eveniet, ut et voluptates repudiandae nin et molestia non recusandae. Itaque earum rerum definiturum, quas natura non depravita desidera. Est quem and ma accedia, saluto: cheare: Inquam, Tiler! lictores, turns omnis choraque: chaere. Tiler! hine hostis in Albacitis, hine inimicus. Sed ture Mueius. Ego autem mirari stati non queo unde hoc at tam insolems domesticarum rerum fastidum. Nos et omnis hic docendi locus sed in proreau estition, neque eum Torquatum, qui hoe primas cognomen invenent; aut toquem li liun hosti detratiscs; un aliquam ex oe est consceutus? — Laudem et caristatem, que sami vituae sime metu degendae praesida firmissima. Filium morte mulavit.— 5 sime causas, nollem me de no declerari, quo di internativa di deligem physici, ercedere aliqual esse minimum, quel profecto munuquam putavises, si a ricolyano, familiari suo, geometric adiscere maluisset quam illum etiam ipsum dedocere. Sol Democrito naguna videru, quippe homini erudito in geometriaque perfecto, huic pedalis fortasse; tantum enim esse omnino in nostris poetis aut inenties inane eguita delectari ortico. Queritas similari quidem viderum, inernis ac mulau est. Tolli definitiones, nihi de dividendo ac partiendo docet, non quo ignorare voa arbitre, sed ut ratione et via procedar ortico. Queritas similari una mom

```
#meander.reflow({
  import meander: *
  placed(left + horizon,
    boundary:
       contour.height(
       div: 20,
       flush: horizon,
       x => (0.5, 1 - x),
      ) +
       contour.margin(5mm)
)[...]
// ...
})
```

```
#meander.reflow({
   import meander: *
   placed(left + horizon,
      boundary:
      contour.horiz(
          div: 25,
          y => if y <= 0.5 {
            (0, 2 * (0.5 - y))
          } else {
            (0, 2 * (y - 0.5))
          },
          ) +
      contour.margin(5mm)
)[...]
//...
})</pre>
```

The contouring functions available should already cover a reasonable range of use-cases, but if you have other ideas you could always try to submit one as a new issue.

There are of course limits to this technique, and in particular increasing the number of obstacles will in turn increase the number of boxes that the layout is segmented into. This means

- performance issues if you get too wild (though notice that having 20+ obstacles in the previous examples went completely fine, and I have test cases with up to ~100)
- text may not fit in the boxes, and the vertical stretching of boxes still needs improvements.

In the meantime it is highly discouraged to use a subdivision that results in obstacles much smaller than the font height.

V Modularity (WIP)

Because meander is cleanly split into three algorithms (content segmentation, page segmentation, text threading), there are plans to provide

- configuration options for each of those steps
- the ability to replace entirely an algorithm by either a variant, or a user-provided alternative that follows the same signature.

VI Style-sensitive layout

Meander respects most styling options through a dedicated content segmentation algorithm. Bold, italic, underlined, stroked, highlighted, colored, etc. text is preserved through threading, and easily so because those styling options do not affect layout much.

There are however styling parameters that have a consequence on layout, and some of them require special handling. Some of these restrictions may be relaxed or entirely lifted by future updates.

VI.a Paragraph justification

In order to properly justify text across boxes, Meander needs to have contextual access to #par.justify, which is only updated via a #set rule.

```
As such do not use #par(justify: true)[...].
```

Instead prefer #[#set par(justify: true); ...], or put the #set rule outside of the invocation of #meander.reflow altogether.

Wrong

Correct

```
#meander.reflow({
    // ...
    content[
        #set par(justify: true)
        #lorem(600)
    ]
})
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim aeque doleamus animo, cum corpore dolemus, ferit tamen permagna accessio potest, si aliquad aetrum et infinitum impender malum nobis opinemur. Quod idem licet transferre in voluptatem, ut postea variari voluptas distinguique possit, augeri amplificarique non possit. At etiam Athenis, ut e pate andiebam facete et urbane Stoicos irridente, statua est in quo a nobis philosophia defensa et collaudata est, cum id, quod maxime placeat, facere possimus, omnis voluptas assumenda est, omnis dolor repellendus. Temporibus autre etiam quibusdam et aut officiis debitis aut rerum necessitatibus saepe eveniet, ut et voluptates repudiandase sint et molestiae non recuandate. Itaque earum rerum defutrum, quas natura non depravata desiderat. Et quem ad me accedis, saluto: 'chaeree', inquam, 'Itte' lictores, turma omnis chorusque: 'chaere.' Tited' hinc hostis mi Albucius, hine inimitus. Sed iur Mucius. Ego autem mirrai satis non que oune due hos sit tam insolens domesticarum rerum fastidium. Non est omnino hic docendi locus; sed its proesus esistimo, neque eum Torquatum, qui hoc primas cognomen invenerit, aut roquem illum hosti detraxisse, ut aliquam ex eo est consecutus? - Laudem et caritatem, quae sunt vitae sine metu degendate praesida firmissima. - Filium morte multavit. - Si sine causa, nollem en a be o delectari, quod ista Platonis, Aristoteli, Theophrasti orationis ornamenta neglescrit. Nami illud quidem physici, credere aliquid esse minimum, quod profecto unuquam putatisset, si a Polyanon, familiari suo, geometrica discere maluiset quam illum etiam ipsum dedocere. Sol Democrito magnus videtur, quippe homini erudito in geometriquae perfecto, hou pedalis forbase; tantum enim esse omnino in nostris poetis aut inertissimae segnitiae est auf fastidii delicatione, and no quo ginorare vos ennino in nostris poetis aut inertissimae segnitiae est aut fastidii delicatione hectur, qui

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ul labore et dolore magnam aliquum quaerat volupataem. Ut enim aeque dolorums animo, cum corpore dolerums, fieri tamen permagna accessio potest, si aliquod aeterumi en ifinditum impendere malnum nobis opinemur. Quodi idem licet transferre in volupataem, ut postea variari voluptas distinguique possit, augeri amplificarique non possit. At etiam Athenis, ut e postea variari voluptas distinguique possit, augeri amplificarique non possit. At etiam Athenis, ut e patre audebam facete et urbane Solicios riridemie, status est in que a nobis publicosphia defense et colluadata est, cumi di, quel mastimar placeat, facere possimus, omnis voluptas assumenade est, omnis dolor repelleradus. Temporibas auteru quiussdam et aut officiis debits aut rerum necessitatubus saepe eveniet, ut et voluptatae reprudandae sint et nodestiae non recusandae. Itaque earum rerum defuturum, quas natura non depravata desiderat. It quem ad ma excedis, salutor chaere, finquen, Tirle* lictores, turma omnis chorusque: chaere. Tirle* linic hostis mi Albacius, hine inimicus. Sed ure Mucius. Ego autem mirari satis non queo umde hoe sit am insolens domesticarum rerum fastidium. Non est ominion his docendi locus; sed la prorsus existino, neque cum Torquatum, qui hoe primus cognomen invenerit, aut torquem illum hosti derraxises, ut aliquum ex co est conocentura? – Luadum et curiatem, quae aunt vitae sim embu degendae praesida firmissima – Filium morte multavit. – Si sine causa, nollem me ab co delectari, quoi dietraxise, cut aliquum ex co est conocentura? – Luadum en legicerari. Nam luid quidem physic, credere aliquid esse minimum, quod profecto numquam putavisset, si a Polyaeno, familiari suo, geometrica discree maluiser quam illum etiam ipsum dedocere. Sol Democrito mangus videtur, quippe homini erudito in geometriaque perfecto, huic pedalis fortasse; tantum enim esse omninio in notris poetis aut incretissimae seguitiae est ati afatidi delicatissimi. Midh quiddem vid

VI.b Font size

The font size indirectly affects layout because it determines the spacing between lines. When a linebreak occurs between containers, Meander needs to manually insert the appropriate spacing there. Since the spacing is affected by font size, make sure to update the font size outside of the #meander.reflow invocation if you want the correct line spacing.

As such, it is currently discouraged to do large changes of font size in highly segmented regions from within the invocation. A future update will provide a way to do this in a more well-behaved manner.

Wrong

```
#meander.reflow({
    // ...
    content[
        #set text(size: 30pt)
        #lorem(600)
    ]
})
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim aeque doleamus animo, cum corpore dolemus, fieri tamen permagna accessio potest, si aliquod aeternum et infinitum impendere malum nobis opinemur. Quod idem licet transferre in voluptatem, ut postea variari voluptas distinguique possit, augeri amplificarique non possit. At etiam Athenis, ut e patre audiebam facete et urbane Stoicos irridente, statua est in quo a nobis philosophia defensa et collaudata est, cum id, quod maxime placeat, facere possimus, omnis voluptas assumenda est, omnis dolor repellendus. Temporibus autem quibusdam et aut officiis debitis aut rerum necessitatibus saepe eveniet, ut et voluptates repudiandae sint et molestiae non recusandae. Itaque earum rerum defuturum, quas natura non depravata desiderat. Et

Correct

```
#set text(size: 30pt)
#meander.reflow({
    // ...
    content[
        #lorem(600)
    ]
})
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim aeque doleamus animo, cum corpore dolemus, fieri tamen permagna accessio potest, si aliquod aeternum et infinitum impendere malum nobis opinemur. Quod idem licet transferre in voluptatem, ut postea variari voluptas distinguique possit, augeri amplificarique non possit. At etiam Athenis, ut e patre audiebam facete et urbane Stoicos irridente, statua est in quo a nobis philosophia defensa et collaudata est, cum id, quod maxime placeat, facere possimus,

VI.c Hyphenation and language

The language is not yet configurable. This feature will come soon.

Hyphenation can only be fetched contextually, and highly influences how text is split between boxes. Thus hyphenation can currently only be enabled or disabled outside of the #meander.reflow invocation. A future update will provide a means to change it more locally.

Wrong

#meander.reflow({ // ... content[#set text(hyphenate: true) #lorem(600)] })

Correct

```
#set text(hyphenate: true)
#meander.reflow({
    // ...
    content[
        #lorem(600)
    ]
})
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim aeque doleamus animo, cum corpore dolemus, international control de la co

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ul labor et dolore magnam aliquam quaerat voluptatem. Ut enin aeque doleamus animo, cum corpore dolemus, fieri tamen permagna acessio potest, si aliquod aeterumu et infinitum impendere malum nobis opinemus. Quoi diene het transferre in voluptatem. Ut obeste variari voluptas distinguique posses, augeri amplificarique non possit. At etiam Athenis, ut e pare audiebam facete et urbane Stoicos irridente, attuate att in quo anobis philosophia defensas et colludate act, cum id, quod maxime placent. Gecer possimus, omnia voluptas assumenda est, omnia folor repellendus. Temporibus autem quibusdam et aut Gificia debits ant erum necessitatibus saepe evenite, ut et voluptatas repudiandos eint et molestiae non recusandae. Bique earum rerum defuturum, quas naturu non depravata desiderat. Et quem ad me accedis, saltoo: chaere; inquam. Tite? lictores, turma omnis chorusque: chaere, Tite? linc boats mi Albocitsa, lini: nimincus. Sed inze Medicis. Ego autem anirari suits nor quee unde hoc sit tam insolens domesticarum rerum fastidum. Non est omnism bic docendi locus: sed its provisus estistimo, neque extendim place and proquation, qui portura desideratur rerum fastidum. Non est omnism bic docendi locus: sed its provisus estistimo, neque extendim resume fastigui este sum insolens domesticarum rerum fastidum. Non est omnism bic docendi locus: sed its provisus distributiva. Pos inse cassas, nollem me sibe ende degendae praesida firmissima. "Fili imm morter mulvist". Si sinc cassas, nollem me sibe ende degendae praesida firmissima. "Fili mum morter mulvist". Si sinc cassas, nollem me sibe ende degendae praesida firmissima. "Fili mum morter mulvist". Si sinc cassas, nollem me sibe ende degendae praesida firmissima. "Fili mum morter mulvist". Si sinc cassas, nollem me sibe ende degendae praesida firmissima. "Fili mum morte mulvist". Si sinc cassas, nollem me sibe ende degendae praesida firmissima. "Fili mum morter uniquativa." Si sinc cassas, nol

VII Module details

VII.a Geometry (geometry.typ)

Generalist functions for 1D and 2D geometry.

- clamp()
- between()
- intersects()
- resolve()
- align()

clamp

Bound a value between min and max. No constraints on types as long as they support inequality testing.

Parameters

```
clamp(
  val: any,
 min: any none,
 max: any none
) -> any
val
       any
Base value.
min
        any or none
Lower bound.
Default: none
max
        any or none
Upper bound.
```

between

Testing a <= b <= c, helps only computing b once.

Parameters

```
between(
  a: length,
  b: length,
  C: length
) -> bool
```

Default: none

```
a length
Lower bound.
```

```
b length
Tested value.
```

```
c length
Upper bound. Asserted to be >= c.
```

intersects

Tests if two intervals intersect.

Parameters

```
intersects(
  i1: (length, length),
  i2: (length, length),
  tolerance: length
)
```

```
i1 (length, length)
```

First interval as a tuple of (low, high) in absolute lengths.

```
i2 (length, length)
Second interval.
```

```
tolerance length

Set to nonzero to ignore small intersections.

Default: Opt
```

resolve

Converts relative and contextual lengths to absolute. The return value will contain each of the arguments once converted, with arguments that contain 'x' or start with 'w' being interpreted as horizontal, and arguments that contain 'y' or start with 'h' being interpreted as vertical.

```
#context resolve(
   (width: 100pt, height: 200pt),
   x: 10%, y: 50% + 1pt,
   width: 50%, height: 5pt,
)
(x: 10pt, y: 101pt, width: 50pt, height: 5pt)
```

Parameters

```
resolve(
    size: (width: length, height: length),
    ..args: dictionary
) -> dictionary

size (width: length, height: length)
Size of the container as given by the layout function.
```

align

Compute the position of the upper left corner, taking into account the alignment and displacement.

Parameters

```
align(
  alignment: alignment,
  dx: relative,
  dy: relative,
  width: relative,
  height: relative
) -> (x: relative, y: relative)

alignment alignment
Absolute alignment.

dx relative

Horizontal displacement.
```

```
dx relative

Horizontal displacement.

Default: Opt
```

```
dy relative
Vertical displacement.
Default: 0pt
```

```
width relative
Object width.
Default: 0pt
```

```
height relative

Object height.

Default: Opt
```

VII.b Tiling (tiling.typ)

Page splitting algorithm.

- placed()
- container()
- content()
- separate()
- pat-forbidden()
- pat-allowed()
- forbidden-rectangles()
- tolerable-rectangles()
- regions()

placed

Core function to create an obstacle.

Parameters

```
placed(
   align: alignment,
   dx: relative,
   dy: relative,
   boundary: (..function,),
   content: content
) -> obstacle
```

```
align alignment
```

Reference position on the page (or in the parent container).

```
dx relative
```

Horizontal displacement.

Default: 0% + 0pt

```
dy relative
```

Vertical displacement.

Default: 0% + 0pt

```
boundary (..function,)

An array of functions to transform the bounding box of the content. By default, a 5pt margin. See contour.typ.

Default: (auto,)

content content
```

container

Inner content.

Core function to create a container.

Parameters

```
container(
  align: alignment,
  dx: relative,
  dy: relative,
  width: relative,
  height: relative
) -> container

align alignment
Location on the page.
Default: top + left
```

```
dx relative
Horizontal displacement.
Default: 0% + 0pt
```

```
dy relative
Vertical displacement.
Default: 0% + 0pt
```

```
width relative

Width of the container.

Default: 100%
```

```
height relative

Height of the container.

Default: 100%
```

content

Core function to add flowing content.

Parameters

```
content(data: content) -> flowing

data    content

Inner content.
```

separate

Splits the input sequence into obstacles, containers, and flowing content.

An "obstacle" is data produced by the placed function. It can contain arbitrary content, and defines a zone where flowing content cannot be placed.

A "container" is produced by the function container. It defines a region where (once the obstacles are subtracted) is allowed to contain flowing content.

Lastly flowing content is produced by the function content. It will be threaded through every available container in order.

```
#separate({
    // This is an obstacle
    placed(top + left, box(width: 50pt, height: 50pt))
    // This is a container
    container(height: 50%)
    // This is flowing content
    content[#lorem(50)]
})

Parameters
    separate(seq: content) -> (containers: (..box,), obstacles: (..box,), flow: (..content,))
```

pat-forbidden

Pattern with red crosses to display forbidden zones.

Parameters

```
pat-forbidden(sz: length) -> pattern
```

```
sz length
Size of the tiling.
```

pat-allowed

Pattern with green pluses to display allowed zones.

Parameters

```
pat-allowed(sz: length) -> pattern

sz length
Size of the tiling.
```

forbidden-rectangles

From a set of obstacles (see separate: an obstacle is any placed content) construct the blocks (x: length, y: length, width: length, height: length) that surround the obstacles.

The return value is as follows:

- rects, a list of blocks (x: length, y: length, width: length, height: length)
- display, show this to include the placed content in the final output
- debug, show this to include helper boxes to visualize the layout

Parameters

```
forbidden-rectangles(
  obstacles: (..box,),
  size: (width: length, height: length)
) -> (rects: (..box,), display: content, debug: content)

obstacles (..box,)
Array of all the obstacles that are placed on this document.
```

```
size (width: length, height: length)
Dimensions of the parent container, as provided by layout.
Default: none
```

tolerable-rectangles

Partition the complement of avoid into containers as a series of rectangles.

The algorithm is roughly as follows:

```
for container in containers {
  horizontal-cuts = sorted(top and bottom of zone for zone in avoid)
  for (top, bottom) in horizontal-cuts.windows(2) {
    vertical-cuts = sorted(
      left and right of zone for zone in avoid
      if zone intersects (top, bottom)
    )
    new zone (top, bottom, left, right)
  }
}
```

The main difficulty is in bookkeeping and handling edge cases (weird intersections, margins of error, containers that overflow the page, etc.) There are no heuristics to exclude zones that are too small, and no worries about zones that intersect vertically. That would be the threading algorithm's job.

Blocks are given an additional field bounds that dictate the upper limit of how much this block is allowed to stretch vertically, set to the dimensions of the container that produced this block.

Parameters

```
tolerable-rectangles(
  containers: (..box,),
  avoid: (..box,),
  size: (width: length, height: length)
) -> (rects: (..box,), debug: content)

containers (..box,)
Array of the containers in which content can be placed.
```

```
avoid (..box,)
Array of all the obstacles that are placed on this document. Will be subtracted from containers.
Default: ()
```

```
size (width: length, height: length)
Dimensions of the parent container, as provided by layout.
Default: none
```

regions

Debug version of the toplevel reflow, that only displays the partitioned layout.

Parameters

```
regions(
  ct: content,
  display: bool
) -> content
```

```
ct content
```

Content to be segmented and have its layout displayed.

display bool

Whether to show the placed objects.

Default: true

VII.c Contouring (contour.typ)

Image boundary transformers.

- margin()
- frac-rect()
- horiz()
- vert()
- width()
- height()
- grid()
- ascii-art()

Variables

• phantom

margin

Contouring function that pads the inner image.

Parameters

```
margin(size: length) -> function

size length
Padding.
```

frac-rect

Helper function to turn a fractional box into an absolute one.

Parameters

```
frac-rect(
  frac: (x: fraction, y: fraction, width: fraction, height: fraction),
  abs: (x: length, y: length, width: length, height: length),
    ..style
) -> (x: length, y: length, width: length, height: length)

frac (x: fraction, y: fraction, width: fraction, height: fraction)
Child dimensions as fractions.
```

```
abs (x: length, y: length, width: length, height: length)
Parent dimensions as absolute lengths.
```

```
..style
Currently ignored.
```

horiz

Horizontal segmentation as (left, right)

Parameters

```
horiz(
  div: int,
  fun: function(fraction) => (fraction, fraction)
) -> function
```

```
div int
```

Number of subdivisions.

Default: 5

```
fun function(fraction) => (fraction, fraction)
```

For each location, returns the left and right bounds.

vert

Vertical segmentation as (top, bottom)

Parameters

```
vert(
  div: int,
  fun: function(fraction) => (fraction, fraction)
) -> function
```

```
div int
```

Number of subdivisions.

Default: 5

```
fun
function(fraction) => (fraction, fraction)
```

For each location, returns the top and bottom bounds.

width

Horizontal segmentation as (anchor, width).

```
Parameters
```

```
width(
    div: int,
    flush: alignment,
    fun: function(fraction) => (fraction, fraction)
) -> function

div int
Number of subdivisions.
Default: 5
```

```
flush alignment
```

Relative horizontal alignment of the anchor.

Default: center

```
fun function(fraction) => (fraction, fraction)
```

For each location, returns the position of the anchor and the width.

height

Vertical segmentation as (anchor, height).

Parameters

```
height(
  div: int,
  flush: alignment.,
  fun: function(fraction) => (fraction, fraction)
) -> function
```

```
div int
```

Number of subdivisions.

Default: 5

```
flush alignment.
```

Relative vertical alignment of the anchor.

Default: horizon

```
fun
function(fraction) => (fraction, fraction)
```

For each location, returns the position of the anchor and the height.

grid

Cuts the image into a rectangular grid then checks for each cell if it should be included. The resulting cells are automatically grouped horizontally.

Parameters

```
grid(
    div: int (x: int, y: int),
    fun: function(fraction, fraction) => bool
) -> function

div    int or (x: int, y: int)

Number of subdivisions.

Default: 5
```

```
fun function(fraction, fraction) => bool
```

Returns for each cell whether it satisfies the 2D equations of the image's boundary.

ascii-art

Allows drawing the shape of the image as ascii art.

Blocks

- #: full
- : empty

Half blocks

- [: left
-]: right
- ^: top
- _: bottom

Quarter blocks

- `: top left
- ': top right
- ,: bottom left
- .: bottom right

Anti-quarter blocks

- J: top left
- L: top right
- 7: bottom left

• F: bottom right

Diagonals

- /: positive
- \: negative

Parameters

```
ascii-art(ascii: code)
```

```
ascii code
```

Draw the shape of the image in ascii art.

phantom function

Drops all boundaries. Using boundary: phantom will let other content flow over this object.

VII.d Bisection (bisect.typ)

Content splitting algorithm.

- fits-inside()
- default-rebuild()
- take-it-or-leave-it()
- has-text()
- has-child()
- has-children()
- is-list-item()
- is-enum-item()
- has-body()
- dispatch()
- fill-box()

fits-inside

Tests if content fits inside a box.

WARNING: horizontal fit is not very strictly checked A single word may be said to fit in a box that is less wide than the word. This is an inherent limitation of measure(box(...)) and I will try to develop workarounds for future versions.

The closure of this function constitutes the basis of the entire content splitting algorithm: iteratively add content until it no longer fits-inside, with what "iteratively add content" means being defined by the content structure. Essentially all remaining functions in this file are about defining content that can be split and the correct way to invoke fits-inside on them.

```
#let dims = (width: 100%, height: 50%)
#box(width: 7cm, height: 3cm)[#layout(size
=> context {
  let words = [#lorem(12)]
  [#fits-inside(dims, words, size: size)]
  linebreak()
  box(..dims, stroke: 0.1pt, words)
```

})]

true

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor.

```
#let dims = (width: 100%, height: 50%)
#box(width: 7cm, height: 3cm)[#layout(size
=> context {
  let words = [#lorem(15)]
  [#fits-inside(dims, words, size: size)]
  linebreak()
  box(..dims, stroke: 0.1pt, words)
})]
```

false

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore.

Parameters

```
fits-inside(
  dims: (width: relative, height: relative),
  ct: content,
  size: (width: length, height: length)
) -> bool
```

```
dims (width: relative, height: relative)
```

Maximum container dimensions. Relative lengths are allowed.

```
ct content
```

Content to fit in.

```
size (width: length, height: length)
```

Dimensions of the parent container to resolve relative sizes. These must be absolute sizes.

Default: none

default-rebuild

Destructure and rebuild content, separating the outer content builder from the rest to allow substituting the inner contents. In practice what we will usually do is recursively split the inner contents and rebuild the left and right halves separately.

Inspired by wrap-it's implementation (see: _rewrap in github:ntjess/wrap-it)

```
#let content = box(stroke: red)[Initial]
#let (inner, rebuild) = default-rebuild(
    content, "body",
```

```
Content: #content \
Inner: #inner \
Rebuild: #rebuild("foo")
```

```
Content: Initial
Inner: Initial
Rebuild: foo
```

```
#let content = [*_Initial_*]
#let (inner, rebuild) = default-rebuild(
   content, "body",
)

Content: #content \
Inner: #inner \
Rebuild: #rebuild("foo")
```

```
Content: Initial
Inner: Initial
Rebuild: foo
```

```
#let content = [a:b]
#let (inner, rebuild) = default-rebuild(
   content, "children",
)

Content: #content \
Inner: #inner \
Rebuild: #rebuild(([x], [y]))
```

```
Content: a:b
Inner: ([a], [:], [b])
Rebuild: xy
```

Parameters

```
default-rebuild(
  ct: content,
  inner-field: string
) -> (dictionnary, function)
```

```
inner-field string
What "inner" field to fetch (e.g. "body", "text", "children", etc.)
```

take-it-or-leave-it

"Split" opaque content.

Parameters

```
take-it-or-leave-it(
  ct: content,
  fits-inside: function
) -> (content?, content?)
```

```
ct content
```

This content cannot be split. If it fits take it, otherwise keep it for later.

fits-inside function

Closure to determine if the content fits (see fits-inside above).

has-text

Split content with a "text" main field. Strategy: split by " " and take all words that fit. Then if hyphenation is enabled, split by syllables and take all syllables that fit. End the block with a linebreak that has the justification of the paragraph.

Parameters

```
has-text(
  ct: content,
  split-dispatch: function,
  fits-inside: function,
  cfg: dictionary
)
```

ct content

Content to split.

split-dispatch function

Recursively passed around (see split-dispatch below).

fits-inside function

Closure to determine if the content fits (see fits-inside above).

```
cfg dictionary
```

Extra configuration options.

has-child

Split content with a "child" main field. Strategy: recursively split the child.

Parameters

```
has-child(
  ct: content,
  split-dispatch: function,
  fits-inside: function,
  cfg: dictionary
)
```

```
ct content
```

Content to split.

```
split-dispatch function
```

Recursively passed around (see split-dispatch below).

```
fits-inside function
```

Closure to determine if the content fits (see fits-inside above).

```
cfg dictionary
```

Extra configuration options.

has-children

Split content with a "children" main field. Strategy: take all children that fit.

Parameters

```
has-children(
  ct: content,
  split-dispatch: function,
  fits-inside: function,
  cfg: dictionary
)
```

ct content

Content to split.

```
split-dispatch function
```

Recursively passed around (see split-dispatch below).

```
fits-inside function
```

Closure to determine if the content fits (see fits-inside above).

```
cfg dictionary
```

Extra configuration options.

is-list-item

Split a list.item. Strategy: recursively split the body, and do some magic to simulate a bullet point indent.

Parameters

```
is-list-item(
  ct: content,
  split-dispatch: function,
  fits-inside: function,
  cfg: dictionary
)
```

```
ct content
```

Content to split.

```
split-dispatch function
```

Recursively passed around (see split-dispatch below).

```
fits-inside function
```

Closure to determine if the content fits (see fits-inside above).

```
cfg dictionary
```

Extra configuration options.

is-enum-item

Split an enum.item. Strategy: recursively split the body, and do some magic to simulate a numbering indent.

Parameters

```
is-enum-item(
  ct: content,
  split-dispatch: function,
  fits-inside: function,
  cfg: dictionary
)
```

```
ct content
```

Content to split.

```
split-dispatch function
```

Recursively passed around (see split-dispatch below).

fits-inside function

Closure to determine if the content fits (see fits-inside above).

```
cfg dictionary
```

Extra configuration options.

has-body

Split content with a "body" main field. There is a special strategy for list.item and enum.item which are handled separately. Elements strong, emph, underline, stroke, overline, highlight are splittable, the rest are treated as non-splittable.

Parameters

```
has-body(
  ct: content,
  split-dispatch: function,
  fits-inside: function,
  cfg: dictionary
)
```

ct content

Content to split.

split-dispatch function

Recursively passed around (see split-dispatch below).

fits-inside function

Closure to determine if the content fits (see fits-inside above).

```
cfg dictionary
```

Extra configuration options.

dispatch

Based on the fields on the content, call the appropriate splitting function. This function is involved in a mutual recursion loop, which is why all other splitting functions take this one as a parameter.

Parameters

```
dispatch(
  ct: content,
  fits-inside: function,
  cfg: dictionary
)
```

ct content

Content to split.

fits-inside function

Closure to determine if the content fits (see fits-inside above).

```
cfg dictionary
```

Extra configuration options.

fill-box

Initialize default configuration options and take as much content as fits in a box of given size. Returns a tuple of the content that fits and the content that overflows separated.

Parameters

```
fill-box(
  dims: (width: length, height: length),
  ct: content,
  size: (width: length, height: length),
  cfg: dictionary
) -> (content, content)

dims (width: length, height: length)
Container size.
```

```
ct content
Content to split.
```

```
size (width: length, height: length)
Parent container size.
Default: none
```

```
cfg dictionary
Configuration options.

• list-markers: (..content,), default value ([•], [•], [-], [•], [-]). If you change the markers of list, put the new value in the parameters so that lists are correctly split.

• enum-numbering: (..str,), default value ("1.", "1.", "1.", "1.", "1.", "1."). If you change the numbering style of enum, put the new style in the parameters so that enums are
```

Default: (:)

correctly split.

VII.e Threading (threading.typ)

Filling and stretches boxes iteratively.

- smart-fill-boxes()
- reflow()

smart-fill-boxes

Thread text through a list of boxes in order, allowing the boxes to stretch vertically to accommodate for uneven tiling.

Parameters

```
smart-fill-boxes(
  body: content,
  avoid: (..block,),
  boxes: (..block,),
  extend: length,
  size: (width: length, height: length)
) -> (..content,)
```

```
body content
```

Flowing text.

```
avoid (..block,)
Obstacles to avoid. A list of (x: length, y: length, width: length, height: length).
Default: ()
```

```
boxes (..block,)
Boxes to fill. A list of (x: length, y: length, width: length, height: length, bound: block).
bound is the upper limit of how much to stretch the container, i.e. also (x: length, y: length, width: length, height: length).
Default: ()
```

extend length

How much the baseline can extend downwards (within the limits of bounds).

Default: 1em

```
size (width: length, height: length)
Dimensions of the container as given by layout.
Default: none
```

reflow

Segment the input content according to the tiling algorithm, then thread the flowing text through it.

Parameters

```
reflow(
   ct: content,
   debug: bool
) -> content
```

ct content

See module tiling for how to format this content.

debug bool

Whether to show the boundaries of boxes.

Default: false