wrap-content

Places to-wrap next to fixed, wrapping to-wrap as its height overflows fixed.

Basic Use:

```
Lorem ipsum dolor sit amet, consectetur
#let body = lorem(40)
                                                                  adipiscing elit, sed do eiusmod tempor
#wrap-content(rect(fill: teal), body)
                                                                  incididunt ut labore et dolore magnam
                                                     aliquam quaerat voluptatem. Ut enim aeque doleamus
                                                     animo, cum corpore dolemus, fieri tamen permagna
```

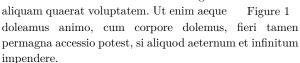
impendere.

Something More Fun:

```
#set par(justify: true)
// Helpers; not required
#let grad(map) = {
  gradient.linear(
    ..eval("color.map." + map)
}
#let make-fig(fill) = {
  set figure.caption(separator: "")
  fill = grad(fill)
  figure(
    rect(fill: fill, radius: 0.5em),
    caption: [],
  )
#let (fig1, fig2) = {
  ("viridis", "plasma").map(make-fig)
}
#wrap-content(fig1, body, align: right)
#wrap-content(fig2, [#body #body],
align: bottom)
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim aeque

accessio potest, si aliquod aeternum et infinitum



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



Figure 2

aliquam quaerat voluptatem. Ut enim aeque doleamus animo, cum corpore dolemus, fieri tamen permagna accessio potest, si aliquod aeternum et infinitum impendere.

Note that you can increase the distance between a figure's bottom and the wrapped text by boxing it with an inset:

```
\#let spaced = box(
  make-fig("rocket"),
  inset: (bottom: 0.3em)
#wrap-content(spaced, body)
```



Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim aeque Figure 3 doleamus animo, cum corpore dolemus, fieri

tamen permagna accessio potest, si aliquod aeternum et infinitum impendere.

Parameters

```
wrap-content(
fixed: content,
to-wrap: content,
 align: alignment,
styles,
size,
 ..grid-kwargs: any
```

fixed content

Content that will not be wrapped, (i.e., a figure).

to-wrap content

Content that will be wrapped, (i.e., text). Currently, logic works best with pure-text content, but hypothetically will work with any content.

align alignment

Alignment of fixed relative to to-wrap. top will align the top of fixed with the top of to-wrap, and bottom will align the bottom of fixed with the bottom of to-wrap. left and right alignments determine horizontal alignment of fixed relative to to-wrap. Alignments can be combined, i.e., bottom + right will align the bottom-right corner of fixed with the bottom-right corner of to-wrap.

```
#wrap-content(
  make-fig("turbo"),
  body,
  align: bottom + right
)
```

Lorem ipsum dolor $_{
m sit}$ amet. consectetur adipiscing do eiusmod tempor elit, sedincididunt 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. Figure 4

Default: top + left

..grid-kwargs any

Keyword arguments to pass to the underlying grid function. Of note:

- column-gutter controls horizontal margin between fixed and to-wrap. Or, you can surround the fixed content in a box with (inset: ...) for more fine-grained control.
- columns can be set to force sizing of fixed and to-wrap. For instance, columns: (50%, 50%) will force fixed and to-wrap to each take up half of the available space. If content isn't this big, the fill will be blank margin.

```
#let spaced = box(
   make-fig("mako"), inset: 0.5em
)
#wrap-content(spaced, body)
```





Figure 5

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.

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,



Figure 6

cum corpore dolemus, fieri tamen permagna accessio potest, si aliquod aeternum et infinitum impendere.

wrap-top-bottom

Wrap a body of text around two pieces of content. The logic only works if enough text exists to overflow both the top and bottom content. Use this instead of 2 separate wrap-content calls if you want to avoid a paragraph break between the top and bottom content.

Example:

```
#let fig1 = make-fig("inferno")
#let fig2 = make-fig("rainbow")
#wrap-top-bottom(fig1, fig2, lorem(60))
```



Figure 7

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 $in finitum\ impendere\ malum\ nobis\ opinemur.$ Quod idem licet transferre in voluptatem, ut postea variari voluptas distinguique possit,

augeri amplificarique non possit. At.



Parameters

```
wrap-top-bottom(
top-fixed: content,
bottom-fixed: content,
body: content,
top-kwargs: any,
bottom-kwargs: any
```

top-fixed content

Content that will not be wrapped, (i.e., a figure).

bottom-fixed content

Content that will not be wrapped, (i.e., a figure).

body content

Content that will be wrapped, (i.e., text)

top-kwargs any

Keyword arguments to pass to the underlying wrap-content function for the top content. x alignment is kept (left/right), but y alignment is overridden to top.

Default: (:)

bottom-kwargs any

Keyword arguments to pass to the underlying wrap-content function for the bottom content. x alignment is kept (left/right), but y alignment is overridden to bottom.

Default: (:)