

# subpar

v0.2.0

2024-11-18

MIT

Create sub figures easily.

tinger <ME@TINGER.DEV>

<https://github.com/tingerrr/subpar>

**SUBPAR** provides easy to use sub figures with sensible default numbering and an easy-to-use no-setup API.

## Table of contents

### I. Manifest

### II. Guide

II.1. Labeling .....	3
II.2. Grid Layout .....	3
II.3. Numbering .....	4
II.4. Supplements .....	4
II.5. Appearance .....	5

### III. Reference

III.1. Subpar .....	6
III.2. Default .....	10

### IV. Index

## Part I.

# Manifest

`SUBPAR` aims to be:

- simple to use
  - importing a function and using it should be all that is needed
  - setup required to make the package work should be avoided
- unsurprising
  - parameters should have sensible names and behave as one would expect
  - deviations from this must be documented and easily accesible to Typst novices
- interoperable
  - `SUBPAR` should be easy to use with other packages by default or provide sufficient configuration to allow this in other ways
- minimal
  - it should only provide features which are specifically used for sub figures

If you think its behavior is surprising, you believe you found a bug or think its defaults or parameters are not sufficient for your use case, please open an issue at [GitHub:tingerrr/subpar](https://github.com/tingerrr/subpar). Contributions are also welcome!

## Part II. Guide

### II.1. Labeling

Currently to refer to a super figure the label must be explicitly passed to super using `label: <...>`.

### II.2. Grid Layout

The default super function provides only the style rules to make sub figures correctly behave with respect to numbering. To arrange them in a specific layout, you can use any other Typst function, a common choice would be `grid`.

```
#subpar.super(  
  grid(  
    [#figure([a], caption: [An image]) <fig1a>],  
    [#figure([b], caption: [Another image]) <fig1b>],  
    figure([c], caption: [A third unlabeled image]),  
    columns: (1fr,) * 3,  
  ),  
  caption: [A figure composed of three sub figures.],  
  label: <fig1>,  
)
```

We can refer to `@fig1`, `@fig1a` and `@fig1b`.

---

a	b	c
(a) An image	(b) Another image	(c) A third unlabeled image

Figure 1: A figure composed of three sub figures.

We can refer to Figure 1, Figure 1a and Figure 1b.

Because this quickly gets cumbersome, `SUBPAR` provides a default grid layout wrapper called `grid`. It provides good defaults like `gutter: 1em` and hides options which are undesirable for sub figure layouts like `fill` and `stroke`. To label sub figures simply add a label after a figure like below.

```
#subpar.grid(
  figure([a], caption: [An image]), <fig2a>,
  figure([b], caption: [Another image]), <fig2b>,
  figure([c], caption: [A third unlabeled image]),
  columns: (1fr,) * 3,
  caption: [A figure composed of three sub figures.],
  label: <fig2>,
)
```

We can refer to @fig2, @fig2a and @fig2b.

---

a	b	c
(a) An image	(b) Another image	(c) A third unlabeled image

Figure 2: A figure composed of three sub figures.

We can refer to Figure 2, Figure 2a and Figure 2b.

## II.3. Numbering

subpar and grid take three different numberings:

**numbering** The numbering used for the sub figures when displayed or referenced.

**numbering-sub** The numbering used for the sub figures when displayed.

**numbering-sub-ref** The numbering used for the sub figures when referenced.

Similarly to a normal figure, these can be functions or string patterns. The numbering-sub and numbering-sub-ref patterns will receive both the super figure and sub figure number.

## II.4. Supplements

Currently, supplements for super figures propagate down to sub figures, this ensures that the supplement in a reference will not confuse a reader, but it will cause reference issues in multilingual documents (see [subpar#4](#)).

```
#subpar.grid(
  figure(``typst Hello Typst!``, caption: [Typst Code]), <sup-ex-code1>,
  figure(lorem(10), caption: [Lorem]),
  columns: (1fr, 1fr),
  caption: [A figure containing two super figures.],
  label: <sup-ex-super1>,
)
```

---

Hello Typst!	Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do.
(a) Typst Code	(b) Lorem

Figure 3: A figure containing two super figures.

When referring to the super figure we see “Figure 3”, when referring to the sub figure of a different kind, we still see the same supplement “Figure 3a”.

To turn this behavior off, set `propagate-supplement` to `false`, this will also resolve the issues from [subpar#4](#).

```
#subpar.grid(
  figure(``typst Hello Typst!``, caption: [Typst Code]), <sup-ex-code2>,
  figure(lorem(10), caption: [Lorem]),
  columns: (1fr, 1fr),
  propagate-supplement: false,
  caption: [A figure containing two super figures.],
  label: <sup-ex-super2>,
)
```

---

<p>Hello Typst!</p> <p>(a) Typst Code</p>	<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do.</p> <p>(b) Lorem</p>
---	--

Figure 4: A figure containing two super figures.

Now when referring the the super figure we see still see “Figure 4”, but when referring to the sub figure of a different kind, we the inferred supplement “Listing 4a”.

## II.5. Appearance

The `super` and `grid` functions come with a few arguments to control how super or sub figures are rendered. These work similar to `show rules`, i.e. they receive the element they apply to and display them.

**show-sub** Apply a show rule to all sub figures.

**show-sub-caption** Apply a show rule to all sub figures’ captions.

```
#subpar.grid(
  figure(lorem(2), caption: [An Image of ...]),
  figure(lorem(2), caption: [Another Image of ...]),
  numbering-sub: "1a",
  show-sub-caption: (num, it) => {
    it.supplement
    [ ]
    num
    [: ]
    it.body
  },
  columns: 2,
  caption: [Two Figures],
)
```

---

<p>Figure 5a: An Image of ...</p>	<p>Figure 5b: Another Image of ...</p>
-----------------------------------	--

Figure 5: Two Figures

Unfortunately, to change how a super figure is shown without changing how a sub figure is shown you must use a regular show rule and reconstruct the normal appearance in the sub figures using `show-sub`. Subpar provides a default implementation for this: `subpar.default.show-figure`, it can be passed directly to `show-sub`.

## Part III.

# Reference

### III.1. Subpar

The package entry point.

<code>#grid()</code>	<code>#super()</code>
----------------------	-----------------------

```
#super(  
  <kind>: "image",  
  <numbering>: "1",  
  <numbering-sub>: "(a)",  
  <numbering-sub-ref>: "1a",  
  <supplement>: auto,  
  <propagate-supplement>: true,  
  <caption>: none,  
  <placement>: none,  
  <scope>: "column",  
  <gap>: 0.65em,  
  <outlined>: true,  
  <outlined-sub>: false,  
  <label>: none,  
  <show-sub>: auto,  
  <show-sub-caption>: auto,  
  <body>
```

) → **content**

Creates a figure which may contain other figures, a *superfigure*. For the meaning of parameters take a look at the regular figure documentation.

See `#grid()` for a function which places its sub figures in a grid.

—Argument—

`<kind>: "image"` str | function

The image kind which should be used, this is mainly relevant for introspection and defaults to image. This cannot be automatically resolved like for normal figures and must be set.

—Argument—

`<numbering>: "1"` str | function

This is the numbering used for this super figure.

—Argument—

`<numbering-sub>: "(a)"` str | function

This is the numbering used for the sub figures.

### 3.1 Subpar

—Argument—

`<numbering-sub-ref>: "1a"` `str` | `function`

This is the numbering used for *references* to the sub figures. If this is a function, it receives both the super and sub figure numbering respectively.

—Argument—

`<supplement>: auto` `content` | `function` | `auto` | `none`

The supplement used for this super figure *and* the sub figures when referenced.

—Argument—

`<propagate-supplement>: true` `bool`

Whether the super figure's supplement should propagate down to its sub figures.

—Argument—

`<caption>: none` `content`

The caption of this super figure.

—Argument—

`<placement>: none` `alignment` | `auto` | `none`

The float placement of this super figure.

—Argument—

`<scope>: "column"` `str`

Relative to which containing scope the figure is placed. Set this to "parent" to create a full-width figure in a two-column document. Has no effect if placement is none. Can be set to "parent" or "column".

—Argument—

`<gap>: 0.65em` `length`

The gap between this super figure's caption and body.

—Argument—

`<outlined>: true` `bool`

Whether this super figure should appear in an outline of figures.

—Argument—

`<outlined-sub>: false` `bool`

Whether the sub figures should appear in an outline of figures.

—Argument—

`<label>: none` `label` | `none`

The label to attach to this super figure.

—Argument—

`<show-sub>: auto` `function` | `auto`

### 3.1 Subpar

A show rule override for sub figures. Receives the sub figure.

—Argument—

`<show-sub-caption>`: `auto`

`function` | `auto`

A show rule override for sub figure's captions. Receives the realized numbering and caption element.

```
#grid(  
  <columns>: auto,  
  <rows>: auto,  
  <gutter>: 1em,  
  <column-gutter>: auto,  
  <row-gutter>: auto,  
  <align>: bottom,  
  <inset>: "(:)",  
  <kind>: "image",  
  <numbering>: "1",  
  <numbering-sub>: "(a)",  
  <numbering-sub-ref>: "1a",  
  <supplement>: auto,  
  <propagate-supplement>: true,  
  <caption>: none,  
  <placement>: none,  
  <scope>: "column",  
  <gap>: 0.65em,  
  <outlined>: true,  
  <outlined-sub>: false,  
  <label>: none,  
  <show-sub>: auto,  
  <show-sub-caption>: auto,  
  ..<args>  
)
```

→ `content`

Provides a convenient wrapper around `#super()` which puts sub figures in a grid.

—Argument—

`<columns>`: `auto`

`auto` | `int` | `relative` | `fraction` | `array`

Corresponds to the grid's columns parameter.

—Argument—

`<rows>`: `auto`

`auto` | `int` | `relative` | `fraction` | `array`

Corresponds to the grid's rows parameter.

—Argument—

`<gutter>`: `1em`

`auto` | `int` | `relative` | `fraction` | `array`

Corresponds to the grid's gutter parameter.



### 3.1 Subpar

—Argument—

`<column-gutter>`: `auto` `auto` | `int` | `relative` | `fraction` | `array`

Corresponds to the grid's column-gutter parameter.

—Argument—

`<row-gutter>`: `auto` `auto` | `int` | `relative` | `fraction` | `array`

Corresponds to the grid's row-gutter parameter.

—Argument—

`<align>`: `bottom` `auto` | `array` | `alignment` | `function`

Corresponds to the grid's align parameter.

—Argument—

`<inset>`: `("(:)")` `relative` | `array` | `dictionary` | `function`

Corresponds to the grid's inset parameter.

—Argument—

`<kind>`: `"image"` `str` | `function`

Corresponds to the super figure's kind.

—Argument—

`<numbering>`: `"1"` `str` | `function`

Corresponds to the super figure's numbering.

—Argument—

`<numbering-sub>`: `"(a)"` `str` | `function`

Corresponds to the super figure's numbering-sub.

—Argument—

`<numbering-sub-ref>`: `"1a"` `str` | `function`

Corresponds to the super figure's numbering-sub-ref.

—Argument—

`<supplement>`: `auto` `content` | `function` | `auto` | `none`

Corresponds to the super figure's supplement.

—Argument—

`<propagate-supplement>`: `true` `bool`

Corresponds to the super figure's propagate-supplement.

—Argument—

`<caption>`: `none` `content`

Corresponds to the super figure's caption.

### 3.1 Subpar

—Argument—

`<placement>: none`

alignment | auto | none

Corressponds to the super figure's placement.

—Argument—

`<scope>: "column"`

str

Corressponds to the super figure's scope.

—Argument—

`<gap>: 0.65em`

length

Corressponds to the super figure's gap.

—Argument—

`<outlined>: true`

bool

Corressponds to the super figure's outlined.

—Argument—

`<outlined-sub>: false`

bool

Corressponds to the super figure's outlined-sub.

—Argument—

`<label>: none`

label | none

Corressponds to the super figure's label.

—Argument—

`<show-sub>: auto`

function

Corressponds to the super figure's show-sub.

—Argument—

`<show-sub-caption>: auto`

function

Corressponds to the super figure's show-sub-caption.

**#sub-figure-counter**

The counter used for sub figures.

## III.2. Default

Contains default implementations for show rules to easily reverse show rules in a scope.

**#show-figure()**

**#show-figure()[(self\_)] → content**

The default figure show rule. The active set rules will be used.

This function is contextual.

### 3.2 Default

Argument

`<self_>`

content

The figure to show using the default show rule.

## Part IV.

### Index

#### G

`#grid` ..... 8

#### S

`#show-figure` ..... 10

`#super` ..... 6