

# subpar

v0.2.0

2025-02-07

MIT

Create sub figures easily.

TINGER

✉ [tinger@tinger.dev](mailto:tinger@tinger.dev)

**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 .....	7
III.2 Default .....	11

### 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]) <figla>],
    [#figure([b], caption: [Another image]) <figlb>],
    figure([c], caption: [A third unlabeled image]),
    columns: (lfr,) * 3,
  ),
  caption: [A figure composed of three sub figures.],
  label: <figl>,
)
```

We can refer to `@figl`, `@figla` and `@figlb`.

---

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 `@figl`, `@figla`, and `@figlb`.

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 , and .

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

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>,
)
```

---

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

Figure 4: A figure containing two super figures.

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

## 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],
)
```

---

Lorem ipsum.                      Lorem ipsum.  
 Figure 5a: An Image of ...    Figure 5b: Another Image of ...  
 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.

#grid #super

```
#grid(
  {kind}: image,
  {numbering}: "1",
  {numbering-sub}: "(a)",
  {numbering-sub-ref}: "1a",
  {supplement}: auto,
  {propagate-supplement}: true,
  {outlined-sub}: false,
  {label}: none,
  {show-sub}: auto,
  {show-sub-caption}: auto,
  {figure-overrides}: figure-overrides,
  {grid-overrides}: grid-overrides,
  {grid-styles}: auto,
  ..{args}
) → content
```

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

Argument

{kind}: image string | function

The image kind which should be used, this is mainly relevant for introspection and defaults to image.

Argument

{numbering}: "1" string | function

This is the numbering used for this super figure.

Signature: (integer) → content

Argument

{numbering-sub}: "(a)" string | function

This is the numbering used for the sub figures.

Signature: (integer) → content

Argument

`(numbering-sub-ref): "1a" string | 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.

Signature: (integer, integer) → content

Argument

`(supplement): auto content | auto`

The super figure's supplement.

Argument

`(propagate-supplement): true boolean`

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

Argument

`(outlined-sub): false boolean`

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`

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

Signature: (content) → content

Argument

`(show-sub-caption): auto function | auto`

A show rule override for sub figure's captions. Receives the realized numbering and caption element. The numbering can be used directly without any further formatting.

Signature: (content, content) → content

Argument

`(figure-overrides): figure-overrides dictionary`

The names of named arguments to pass through to the figure directly.



Argument

`{grid-overrides}: grid-overrides`

dictionary

The names of named arguments to pass through to the grid directly.

Argument

`{grid-styles}: auto`

function | auto | none

A template function which applies grid set rules. By default this applies a gutter of 1m. These will be overridden by explicitly passing grid arguments, but will take precedence over the style chain, disabling them allows using the style chain.

Signature: (`content`) → `content`

Argument

`..(args)`

any

Named arguments to pass to figure and grid verbatim, these are selected using `#grid.figure-overrides` and `#grid.grid-overrides` respectively.

```
#super(
  {kind}: image,
  {numbering}: "1",
  {numbering-sub}: "(a)",
  {numbering-sub-ref}: "1a",
  {supplement}: auto,
  {propagate-supplement}: true,
  {outlined-sub}: false,
  {label}: none,
  {show-sub}: auto,
  {show-sub-caption}: auto,
  {overrides}: figure-overrides,
  ..(args),
  {body}
) → content
```

Creates a figure which may contain other figures, a *superfigure*.

This function makes no assumptions about the layout of its sub figures, it simply applies the necessary show and set rules such that all figures within its body get the appropriate numbering.

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

Argument

`{kind}: image`

string | function

The image kind which should be used, this is mainly relevant for introspection and defaults to image.

Must be one of:

- image
- table
- raw

Argument

(numbering): "1"

string | function

This is the numbering used for this super figure.

Signature: (integer) → content

Argument

(numbering-sub): "(a)"

string | function

This is the numbering used for the sub figures.

Signature: (integer) → content

Argument

(numbering-sub-ref): "1a"

string | 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.

Signature: (integer, integer) → content

Argument

(supplement): auto

content | auto

The super figure's supplement.

Argument

(propagate-supplement): true

boolean

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

Argument

(outlined-sub): false

boolean

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

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

Signature: (content) → content

Argument

`(show-sub-caption): auto`

function | auto

A show rule override for sub figure's captions. Receives the realized numbering and caption element. The numbering can be used directly without any further formatting.

Signature: (content, content) → content

Argument

`(overrides): figure-overrides`

dictionary

The names of named arguments to pass through to the figure directly.

Argument

`..(args)`

any

Named arguments to pass to figure verbatim, these are selected using `#super.overrides`.

Argument

`(body)`

The figure body, this may contain other figures which will be numbered appropriately.

**#figure-overrides**

dictionary

The default overrides to use for figures, these are used to pass arguments through to the elements directly.

**#grid-overrides**

dictionary

The default overrides to use for figures, these are used to pass arguments through to the elements directly.

**#sub-figure-counter**

counter

The counter used for sub figures. This is automatically counted within and reset after for each super figure.

## III.2 Default

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

#show-figure

↵ context

#show-figure[it] → content

The default figure show rule. This can be used to display a figure the same way as typst does by default.

Argument

{it}

content

The figure to show using the default show rule.

## Part IV

# Index

## F

`#figure-overrides` ..... 11

## G

`#grid` ..... 7, 9

`#grid-overrides` ..... 11

## S

`#show-figure` ..... 12

`#sub-figure-counter` ... 11

`#super` ..... 7, 9, 11