

Manual

- `init()`
- `theorem()`
- `theorem-style()`
- `proof()`
- `proof-style()`

init

Initialise the package.

Add `#show: trivial.init` to the start of the document.

Parameters

```
init(doc: content) -> content
```

doc `content`

The document.

theorem

Create and display a theorem.

One should define functions so that the first three arguments pre-applied, e.g.

```
#let numbered = trivial.theorem.with(  
  "numbered",  
  trivial.styles.theorem.default,  
  numbering: "1",  
)  
#let theorem = numbered.with([Theorem])  
#let lemma = numbered.with([Lemma])
```

Parameters

```
theorem(  
  kind: str,  
  style: function,  
  name: content,  
  numbering: none str function,  
  ..args: content  
) -> content
```

kind `str`

An identifier for the theorem.

This is used as the wrapper figure's [kind](#), so theorems with the same kind share the same counter.

style `function`

How to format the theorem.

See `theorem-style()` for more information.

name `content`

The name of the theorem.

numbering `none` or `str` or `function`

How to number the theorem. Accepts a [numbering pattern or function](#).

Default: `none`

..args `content`

The optional title and body.

Either one or two arguments are expected.

theorem-style

Type signature of the function expected in `theorem.style`.

Parameters

```
theorem-style(it: figure) -> content
```

it `figure`

The name of the environment is stored in the figure's [supplement](#) and the optional title in its [caption](#).

proof

Create and display a proof.

One should define a new function with the first three arguments pre-applied, e.g.

```
#let proof = trivial.proof.with(  
  [Proof],  
  trivial.styles.proof.default,  
  qed: $qed$,  
)
```

Parameters

```
proof(  
  title: content,  
  style: function,  
  qed: none content,  
  ..args: content  
) -> content
```

title `content`

The default title to use.

style `function`

How to format the proof.

See `proof-style()` for more information.

qed `none` or `content`

The QED symbol to use.

Default: `none`

..args `content`

The optional title and body.

Either one or two arguments are expected.

proof-style

Type signature of the function expected in `proof.style`.

Parameters

```
proof-style(  
  title: content,  
  body: content  
) -> content
```