`sheetstorm`

A template for assignment sheets.

v0.4.0 22 November 2025

https://github.com/rabuu/sheetstorm

Rasmus Buurman **Leander Herter**

Contents

1. Example

```
#import "@preview/sheetstorm:0.4.0": *

#show: assignment.with(
  course: smallcaps[Sheetstorm 101],
  title: "Assignment Example",
  authors: (
    (name: "John Doe", id: 123456),
    (name: "Erika Mustermann", id: 654321),
  ),
  info-box-enabled: true,
  score-box-enabled: true,
  // Here you can customize the layout of the page, the header, the widgets.
  // Look at the parameters of the `assignment` function.
)

#task(name: "Introduction")[
  This is #link("https://github.com/rabuu/sheetstorm")[`sheetstorm`],
  a template library that provides a sane default layout for assignment submissions
  with the option of customizability.

  Here you would write down your solutions for the first task:
  #lorem(30)
]

#task(name: "Subtasks", label: "task-subtasks", points: (1, 2))[
  + _What is the color of a banana?_ #subtask-label("banana", display: "a)")

    A banana is *yellow*.

  + _Solve the following equations for $x$._
    + $x^2 = 4 ==> x = plus.minus 2$
    + $x = "color of banana" ==> x = "yellow"$, see @banana.
]

#task(points: 11)[
  Another task but without a name.
  You can reference tasks from above where you set a label, for example @task-subtasks.

  Then you can do some cool math. You could, for example, try to proof that:
  $ forall n gt.eq 0: sum_(i=0)^n i = (n dot (n+1))/2 $

  #proof[
    It is easy to see that the statement is true for the number $0$:
    $sum_(i=0)^0 i = 0 = (0 dot 1)/2$.
    Let's assume that the statement is true for some $n$. It follows:
    $
      sum_(i=0)^(n+1) i & = sum_(i=0)^n i + (n+1)
                          = (n dot (n+1)) / 2 + (n + 1)
                          = (n^2 + n) / 2 + (2n + 2)/2 \
                        & = (n^2 + 3n + 2) / 2
                          = ((n+1) dot (n+2)) / 2
    $
  ]
]

#task(points: 1, bonus: true)[
  #theorem(name: "Very smart formula", label: "example-theorem")[
    Let $x$ be a natural number. Then:
    $ x + 1 + 1 = x + 2 $
  ]
  #proof[The proof of @example-theorem is trivial.]

  #lorem(50)
]
```

2. Template Setup

Function: assignment

Setup the document as an assignment sheet.

This is the template entrypoint and is expected to be used at the top of the document like so:

```
#show: assignment.with(
  title: "A cool title",
  page-numbering: "1",
  // ...
)
```

Here you can set many options to customize the template's behavior. Some of them replace settings that you would normally use Typst's built-in functions for. For a smooth experience, prefer to set the options here.

Parameters

```
assignment(
  title: content str none,
  title-style: function,
  title-size: length,
  course: content str none,
  authors: array dictionary str none,
  tutor: content str none,
  date: datetime none,
  date-format: auto str,
  margin-left: auto length,
  margin-right: auto length,
  margin-bottom: auto length,
  margin-above-header: auto length,
  margin-below-header: auto length,
  todo-show: bool,
  todo-box: function,
  paper: str,
  page-numbering: none str function,
  header-show-title-on-first-page: bool,
  header-extra-left: content str,
  header-extra-center: content str,
  header-extra-right: content str,
  header-tutor-prefix: content str,
  header-columns: array,
  header-align: array,
  header-column-gutter: length,
  header-row-gutter: length,
  header-padding-top: length,
  header-padding-bottom: length,
  disable-page-configuration: bool,
  initial-task-number: int,
  widget-order-reversed: false,
  widget-column-gap: length,
  widget-row-gap: length,
  widget-spacing-above: length,
  widget-spacing-below: length,
  score-box-enabled: bool,
  score-box-tasks: array auto,
  score-box-show-points: bool,
  score-box-bonus-counts-for-sum: bool,
  score-box-bonus-show-star: bool,
  score-box-inset: length,
  score-box-cell-width: length,
  info-box-enabled: bool,
  info-box-show-ids: bool,
  info-box-show-emails: bool,
  info-box-inset: length,
  info-box-gutter: length,
  hyperlink-style: function,
  doc: content
) -> content
```

title `content` or `str` or `none`

The document’s title.

Default: `none`

title-style `function`

A function that provides styling to the title.

Example

```
#show: assignment.with(
  title: [My *cool* title],
  title-style: t => t,
)
```

Default: `t => underline[*#t*]`

title-size `length`

The text size of the title.

Default: `1.6em`

course `content` or `str` or `none`

The course title, displayed in the header.

Default: `none`

authors `array` or `dictionary` or `str` or `none`

The name and information of the author/s of the document.

Can be just a single author or an array of authors. An author can be just the name as string or a dictionary with the fields:

- `name` (mandatory)
- `id` (optional)
- `email` (optional)

Example

```
#show assignment.with(
  authors: (
    (name: "John Doe", id: 123456, email: "john@doe.com"),
    (name: "Max Mustermann", id: 761524),
    "Bernd",
  ),
)
```

Default: `none`

tutor `content` or `str` or `none`

The name of the tutor, displayed in the header.

Default: `none`

date `datetime` or `none`

The document date, displayed in the header.

Default: `datetime.today()`

date-format `auto` or `str`

The date format that is used to display the document date.

When set to `auto`, a default value based on your document language is chosen.

Default: `auto`

margin-left `auto` or `length`

The left margin of the document. Prefer this to using `page.margin`.

Default: `1.7cm`

margin-right `auto` or `length`

The right margin of the document. Prefer this to using `page.margin`.

Default: `1.7cm`

margin-bottom `auto` or `length`

The bottom margin of the document. Prefer this to using `page.margin`.

Default: `1.7cm`

margin-above-header `auto` or `length`

The top margin of the document, taking the header into account. Prefer this to using `page.margin`.

Default: `0cm`

margin-below-header `auto` or `length`

The margin between the header and the rest of the document. Prefer this to using `page.margin`.

Default: `0cm`

todo-show `bool`

Whether to show a warning beside the title if there are any TODOs in the document.

Default: `true`

todo-box `function`

The layout for the TODO box that may be displayed in the title.

Set this using the provided `todo-box` function.

Example

```
#show: assignment.with(
  todo-box: todo-box.with(stroke: none),
)
```

Default: `todo-box`

paper `str`

The document's paper size. Prefer this to using `page.paper`.

Default: `"a4"`

page-numbering `none` or `str` or `function`

The page numbering. Prefer this to using `page.numbering`.

Default: `"1 / 1"`

header-show-title-on-first-page `bool`

Whether to show the title in the header on the first page as well.

Default: `false`

header-extra-left `content` or `str`

Extra content that is displayed in the left header column.

Default: `none`

header-extra-center `content` or `str`

Extra content that is displayed in the center header column.

Default: `none`

header-extra-right `content` or `str`

Extra content that is displayed in the right header column.

Default: `none`

header-tutor-prefix `content` or `str`

The word that prefixes the tutor name in the header.

Default: `context i18n.translate("Tutor")`

header-columns `array`

The columns settings of the header.

This can break the entire header if you play with it carelessly.

Default: `(1fr, 1.25fr, 1fr)`

header-align `array`

The alignment of the header columns.

This can break the entire header if you play with it carelessly.

Default: `(left, center, right)`

header-column-gutter `length`

The space between the header columns.

Default: `0.5em`

header-row-gutter `length`

The space between the header and the header line.

Default: `0.5em`

header-padding-top `length`

The top padding of the header.

Default: `0.8cm`

header-padding-bottom `length`

The bottom padding of the header.

Default: `1cm`

disable-page-configuration `bool`

Whether to disable basic page settings.

Don’t use this option in a normal document. It is included to use in sepecific settings like example rendering.

Default: `false`

initial-task-number `int`

The initial counter value that is used for task numbering.

Default: `1`

widget-order-reversed `false`

Whether to reverse the order in which the widgets are displayed.

By default, the info box is display first and the score box second.

Default: `false`

widget-column-gap `length`

The horizontal gap between the widgets.

Default: `4em`

widget-row-gap `length`

The vertical gap between the widgets.

Default: `1em`

widget-spacing-above `length`

The padding above the widgets.

Default: `0em`

widget-spacing-below `length`

The padding below the widgets.

Default: `1em`

score-box-enabled `bool`

Whether to display the score box.

Default: `false`

score-box-tasks `array` or `auto`

Set the score box task list manually.

If `auto` is set, figure out the tasks from the context.

Default: `auto`

score-box-show-points `bool`

Whether to show the points per task.

Default: `true`

score-box-bonus-counts-for-sum `bool`

Whether the points of bonus tasks count into the sum.

Default: `false`

score-box-bonus-show-star `bool`

Whether to mark bonus tasks in the score box with a star.

Default: `true`

score-box-inset `length`

The `inset` value of the score box.

Default: `0.7em`

score-box-cell-width `length`

The `cell-width` value of the score box.

Default: `4.5em`

info-box-enabled `bool`

Whether to display the info box.

Default: `false`

info-box-show-ids `bool`

Whether to show the authors `id` values the info box.

Default: `true`

info-box-show-emails `bool`

Whether to show the authors `email` values the info box.

Default: `true`

info-box-inset `length`

The `inset` value of the info box.

Default: `0.7em`

info-box-gutter `length`

The `gutter` value of the info box.

Default: `1em`

hyperlink-style `function`

A function that is used to style hyperlinks created by `link`.

This does *not* have an effect on in-document links but only links that are specified using a string.

Default: `underline`

doc `content`

The document’s body.

3. Tasks

Function: task

Create a task section.

Use this function as primary way to structure your document.

Example

```
#task(name: "Pythagorean theorem", points: 1)[
  _What is the Pythagorean theorem?_

  $ a^2 + b^2 = c^2 $
]
```

Parameters

```
task(
  name: content str none,
  task-prefix: content str,
  label: str none,
  supplement: auto content str function none,
  counter: auto int function,
  counter-show: bool,
  todo-show: bool,
  todo-box: function,
  subtask-numbering: function str none,
  points: int array,
  points-show: bool,
  points-prefix: content str,
  bonus: bool,
  bonus-show-star: bool,
  hidden: bool,
  space-above: auto length,
  space-below: auto length,
  content: content
) -> content
```

name `content` or `str` or `none`

The name of the task.

Default: `none`

task-prefix `content` or `str`

The prefix that is displayed before the task count.

Default: `context i18n.translate("Task")`

label `str` or `none`

A label that you can reference or query.

Default: `none`

supplement `auto` or `content` or `str` or `function` or `none`

The supplement of the task which is used in outlines and references.

When set to `auto`, it uses the same value as `task-prefix`.

Default: `auto`

counter `auto` or `int` or `function`

Set the value of the task counter manually.

If set to `auto`, the counter is not reset. Regardless of this value, the counter is stepped at the end of the task.

Default: `auto`

counter-show `bool`

Whether to display the value of the task counter in the task's title.

Default: `true`

todo-show `bool`

Whether to show a warning beside the title if there are any TODOs in the task.

Default: `true`

todo-box `function`

The layout for the TODO box that may be displayed in the title.

Set this using the provided `todo-box` function.

Example

```
#task(todo-box: todo-box.with(stroke: none))[
  #todo[Some TODO message.]
]
```

Default: `todo-box`

subtask-numbering `function` or `str` or `none`

Set the numbering for subtasks.

If you give a numbering, it just sets the enum numbering accordingly. If you set it to `none`, it disables the custom numbering completely.

The library provides a handy `custom-enum-numbering` function that is expected to be used when setting the numbering to a non-trivial value. The required `enum.full` option is set automatically.

Default: `custom-enum-numbering("a)", "1.", "i.")`

points `int` or `array`

How many points the task can give.

If there are subtasks, you can specify an array of numbers.

Default: `none`

points-show `bool`

Whether to show the points on the right side of the page above the task.

Default: `true`

points-prefix `content` or `str`

The word that is displayed before the points.

Default: `context i18n.translate("Points")`

bonus `bool`

Whether the task is a bonus task.

Default: `false`

bonus-show-star `bool`

Whether bonus tasks are marked with a star in the title.

Default: `true`

hidden `bool`

Whether the task is hidden in the score box.

Default: `false`

space-above `auto` or `length`

Padding above the task.

Default: `auto`

space-below `auto` or `length`

Padding below the task.

Default: `2em`

content `content`

The body of the task.

4. The "TODO" System

Function: todo-box

A box for displaying TODOs.

It is used in different places to layout warning boxes. Usually, you don't use it directly but may pass it to other functions that use it, so you can customize its appearance.

Example

```
#let todo-box = todo-box.with(stroke: none)
#let todo = todo.with(todo-box: todo-box)
#let task = task.with(todo-box: todo-box)
```

Parameters

```
todo-box(
  todo-prefix: content,
  todo-style: function,
  text-style: function,
  stroke: stroke,
  inset: relative dictionary,
  outset: relative dictionary,
  radius: relative dictionary,
  ..content: arguments
) -> content
```

todo-prefix `content`

The text that is displayed at the beginning of the box.

Default: `[TODO]`

todo-style `function`

The styling of the `todo-prefix`.

Default: `t => text(red)[*#t*]`

text-style `function`

The styling of the box's contents.

Default: `t => t`

stroke `stroke`

The stroke of the box.

Default: `red`

inset `relative` or `dictionary`

The `inset` value of the box.

Default: `(x: 0.2em)`

outset `relative` or `dictionary`

The `outset` value of the box.

Default: `(y: 0.3em)`

radius `relative` or `dictionary`

The `radius` value of the box.

Default: `0.3em`

..content `arguments`

Arbitrary content that is passed into the box.

You can only pass *positional* arguments.

Function: todo

Mark a part of your document as unfinished.

This displays an inline warning box and signals to the context.

Parameters

```
todo(
  todo-box: function,
  ..comments: arguments
) -> content
```

todo-box `function`

The layout of the box.

Set this using the provided `todo-box` function.

Example

```
#todo(todo-box: todo-box.with(stroke: none))[Custom box.]
```

Default: `todo-box`

..comments `arguments`

Arbitrary content that is passed into the box.

You can only pass *positional* arguments.

5. Theorem and Proof Environments

Function: theorem

Theorem environment.

Example

```
#theorem(name: "Pythagoras")[$a^2 + b^2 = c^2$]
```

Parameters

```
theorem(
  kind: content str,
  numbering: auto none content str,
  name: content str none,
  label: str none,
  supplement: auto content str function none,
  reference-prefer-name: bool,
  emphasized: bool,
  fill: none color gradient tiling,
  stroke: stroke,
  inset: length,
  outset: length,
  radius: length,
  above: length,
  below: length,
  width: auto relative fraction,
  content: content
) -> content
```

kind `content` or `str`

The theorem kind which is displayed as prefix.

Default: `context i18n.translate("Theorem")`

numbering `auto` or `none` or `content` or `str`

The numbering of the theorem.

- `auto`: Use an automatic counter.
- `none`: Don't number the theorem at all.
- Everything else is displayed as is.

Default: `auto`

name `content` or `str` or `none`

The name of the theorem.

Default: `none`

label `str` or `none`

A label that you can reference or query.

Default: `none`

supplement `auto` or `content` or `str` or `function` or `none`

The supplement of the theorem which is used in outlines and references.

When set to `auto`, it uses the same value as `kind`.

Default: `auto`

reference-prefer-name `bool`

Whether to reference the theorem by name if possible.

By default, theorems are referenced by number if possible.

Default: `false`

emphasized `bool`

Whether the contents of the theorem is emphasized.

Default: `true`

fill `none` or `color` or `gradient` or `tiling`

The `fill` value of the theorem box.

Default: `none`

stroke `stroke`

The `stroke` value of the theorem box.

Default: `none`

inset `length`

The `inset` value of the theorem box.

Default: `0em`

outset `length`

The `outset` value of the theorem box.

Default: `0.5em`

radius `length`

The `radius` value of the theorem box.

Default: `0em`

above `length`

The `above` value of the theorem box.

Default: `1.3em`

below `length`

The `below` value of the theorem box.

Default: `1.3em`

width `auto` or `relative` or `fraction`

The `width` value of the theorem box.

Default: `100%`

content `content`

The body of the theorem.

Function: corollary

Corollary environment, based on the `theorem` environment.

Parameters

```
corollary(
  kind: content str,
  numbering: auto none content str,
  name: content str none,
  label: str none,
  supplement: auto content str function none,
  reference-prefer-name: bool,
  emphasized: bool,
  fill: none color gradient tiling,
  stroke: stroke,
  inset: length,
  outset: length,
  radius: length,
  above: length,
  below: length,
  width: auto relative fraction,
  content: content
) -> content
```

kind `content` or `str`

The theorem kind which is displayed as prefix.

Default: `context i18n.translate("Corollary")`

numbering `auto` or `none` or `content` or `str`

The numbering of the theorem.

- `auto`: Use an automatic counter.
- `none`: Don’t number the theorem at all.
- Everything else is displayed as is.

Default: `auto`

name `content` or `str` or `none`

The name of the theorem.

Default: `none`

label `str` or `none`

A label that you can reference or query.

Default: `none`

supplement `auto` or `content` or `str` or `function` or `none`

The supplement of the theorem which is used in outlines and references.

When set to `auto`, it uses the same value as `kind`.

Default: `auto`

reference-prefer-name `bool`

Whether to reference the theorem by name if possible.

By default, theorems are referenced by number if possible.

Default: `false`

emphasized `bool`

Whether the contents of the theorem is emphasized.

Default: `true`

fill `none` or `color` or `gradient` or `tiling`

The `fill` value of the theorem box.

Default: `none`

stroke `stroke`

The `stroke` value of the theorem box.

Default: `none`

inset `length`

The `inset` value of the theorem box.

Default: `0em`

outset `length`

The `outset` value of the theorem box.

Default: `0.5em`

radius `length`

The `radius` value of the theorem box.

Default: `0em`

above `length`

The `above` value of the theorem box.

Default: `1.3em`

below `length`

The `below` value of the theorem box.

Default: `1.3em`

width `auto` or `relative` or `fraction`

The `width` value of the theorem box.

Default: `100%`

content `content`

The body of the theorem.

Function: lemma

Lemma environment, based on the `theorem` environment.

Parameters

```
lemma(
  kind: content str,
  numbering: auto none content str,
  name: content str none,
  label: str none,
  supplement: auto content str function none,
  reference-prefer-name: bool,
  emphasized: bool,
  fill: none color gradient tiling,
  stroke: stroke,
  inset: length,
  outset: length,
  radius: length,
  above: length,
  below: length,
  width: auto relative fraction,
  content: content
) -> content
```

kind `content` or `str`

The theorem kind which is displayed as prefix.

Default: `context i18n.translate("Lemma")`

numbering `auto` or `none` or `content` or `str`

The numbering of the theorem.

- `auto`: Use an automatic counter.
- `none`: Don't number the theorem at all.
- Everything else is displayed as is.

Default: `auto`

name `content` or `str` or `none`

The name of the theorem.

Default: `none`

label `str` or `none`

A label that you can reference or query.

Default: `none`

supplement `auto` or `content` or `str` or `function` or `none`

The supplement of the theorem which is used in outlines and references.

When set to `auto`, it uses the same value as `kind`.

Default: `auto`

reference-prefer-name `bool`

Whether to reference the theorem by name if possible.

By default, theorems are referenced by number if possible.

Default: `false`

emphasized `bool`

Whether the contents of the theorem is emphasized.

Default: `true`

fill `none` or `color` or `gradient` or `tiling`

The `fill` value of the theorem box.

Default: `none`

stroke `stroke`

The `stroke` value of the theorem box.

Default: `none`

inset `length`

The `inset` value of the theorem box.

Default: `0em`

outset `length`

The `outset` value of the theorem box.

Default: `0.5em`

radius `length`

The `radius` value of the theorem box.

Default: `0em`

above `length`

The `above` value of the theorem box.

Default: `1.3em`

below `length`

The `below` value of the theorem box.

Default: `1.3em`

width `auto` or `relative` or `fraction`

The `width` value of the theorem box.

Default: `100%`

content `content`

The body of the theorem.

Function: proof

Proof environment.

Example

```
#proof[Let $a=1$, then is $a+1 = 2$ immediate.]
```

Parameters

```
proof(
  qed: content,
  prefix: content str,
  prefix-style: function,
  fill: none color gradient tiling,
  stroke: stroke,
  inset: length,
  outset: length,
  radius: length,
  above: length,
  below: length,
  width: auto relative fraction,
  content: content
) -> content
```

qed `content`

The symbol that marks the end of the proof.

Default: `$square$`

prefix `content` or `str`

The prefix that displayed at the beginning of the proof.

Default: `context i18n.translate("Proof")`

prefix-style `function`

The styling of the `prefix`.

Default: `p => [_#p._]`

fill `none` or `color` or `gradient` or `tiling`

The `fill` value of the proof box.

Default: `none`

stroke `stroke`

The `stroke` value of the proof box.

Default: `none`

inset `length`

The `inset` value of the proof box.

Default: `0em`

outset `length`

The `outset` value of the proof box.

Default: `0.5em`

radius `length`

The `radius` value of the proof box.

Default: `0em`

above `length`

The `above` value of the proof box.

Default: `1em`

below `length`

The `below` value of the proof box.

Default: `1em`

width `auto` or `relative` or `fraction`

The `width` value of the proof box.

Default: `100%`

content `content`

The body of the proof.

6. Widgets

Function: score-box

The “score box” widget.

This function creates an empty table for each task where the scores can be filled in. By default, it reads the number of tasks from the `task` counter, but you can set the task values manually.

Usually you just want to use the score box that is automatically available in the `assignment` layout. It uses this implementation internally but all configuration is done in the `assignment` function.

Parameters

```
score-box(
  tasks: array auto,
  show-points: bool,
  bonus-show-star: bool,
  bonus-counts-for-sum: bool,
  fill-space: bool,
  cell-width: length,
  inset: length
) -> content
```

tasks `array` or `auto`

Set the score box task list manually.

If `auto` is set, figure out the tasks from the context.

Default: `auto`

show-points `bool`

Whether to show the points per task.

Default: `true`

bonus-show-star `bool`

Whether the points of bonus tasks count into the sum.

Default: `true`

bonus-counts-for-sum `bool`

Whether to mark bonus tasks in the score box with a star.

Default: `false`

fill-space `bool`

Whether to fill all available horizontal space.

Default: `false`

cell-width `length`

The `inset` value of the score box.

Default: `4.5em`

inset `length`

The `cell-width` value of the score box.

Default: `0.7em`

Function: info-box

The “info box” widget.

This function creates a box with information about the authors of the document. You need to provide the names of the authors and optionally student IDs and/or email addresses.

Usually you just want to use the info box that is automatically available in the `assignment` layout. It uses this implementation internally but all configuration is done in the `assignment` function.

Parameters

```
info-box(
  names: array,
  student-ids: array none,
  emails: array none,
  inset: length,
  gutter: length
) -> content
```

names `array`

The authors names.

student-ids `array` or `none`

The IDs of the authors.

Default: `none`

emails `array` or `none`

The emails of the authors.

Default: `none`

inset `length`

The `inset` value of the info box.

Default: `0.7em`

gutter `length`

The `gutter` value of the info box.

Default: `1em`

7. Utilities

Function: appendix

Setup an appendix layout.

Use this function to turn the rest of the document into an appendix.

Example

```
#show: appendix

= Blablabla
Now everything is displayed as appendix.
```

Parameters

```
appendix(
  title: content str none,
  title-size: length,
  supplement: content str function none,
  numbering: str none,
  body: content
) -> content
```

title `content` or `str` or `none`

The title of the whole appendix.

Default: `context i18n.translate("Appendix")`

title-size `length`

The text size of the appendix title.

Default: `1.6em`

supplement `content` or `str` or `function` or `none`

The supplement of the appendix sections.

Default: `context i18n.translate("Appendix")`

numbering `str` or `none`

The numbering pattern for the appendix sections.

Default: `"A.1."`

body `content`

The appendix body itself.

Function: subtask-label

Attach a label to a subtask.

Example

```
#task[
  + This is the first subtask. #subtask-label("subtask-a", display: "a)")
  + This is the second subtask, see @subtask-a.
]
```

The correct display value does not get inherited from the enum context. You must set it manually.

Parameters

```
subtask-label(
  lbl: str,
  display: content str auto
) -> content
```

lbl `str`

The label identifier.

display `content` or `str` or `auto`

How the subtask is displayed in references.

If set to `auto`, it is set to the same value as `lbl`.

Default: `auto`

Function: custom-enum-numbering

A helper to set a custom enum numbering based on the list depth

You must set `enum(full: true)` for it to work.

Example

```
#set enum(
  full: true,
  numbering: custom-enum-numbering("a)", "1.")
)
```

The numbering for depth levels that are not specified is decided as follows:

- If `cycle` is true then the given patterns get applied cyclically.
- Otherwise, the `default` argument is applied for all unspecified levels.
- If `default` is set to `auto`, the last specified pattern is chosen.

Parameters

```
custom-enum-numbering(
  cycle: bool,
  default: auto str,
  ..patterns: arguments
) -> numbering
```

cycle `bool`

Whether to cycle through the patterns for unspecified depth levels.

Default: `false`

default `auto` or `str`

The default argument for unspecified depth levels.

If set to `auto`, the last specified pattern is chosen.

Default: `auto`

..patterns `arguments`

The string patterns for the numbering in order of their according depth level.