matryoshka

Nested compilers

v0.1.0 2024-07-02 MIT

Run typst from within typst. Useful for documentation authors.

Adrian Freund <ADRIAN@FREUND.IO>

https://github.com/freundTech/typst-matryoshka

MATRYOSHKA, named after the famous nesting dolls, is a typst compiler as a typst plugin. It allows you to compile typst documents form within typst. This is especially useful for documentation authors, who might want to display example code and resulting document in their documentation. MATRYOSHKA renders typst code to svg and then embeds that svg into your original document.

Table of contents

Part I.
About

MATRYOSHKA, named after the famous nesting dolls, is a Typst package that bundles a full Typst compiler as a Typst plugin. This allows you to render Typst documents from within your Typst documents. Why would you want to do this? If you are a documentation author you might want to show Typst source code and the resulting document side-by-side in your documentation. Without MATRYOSHKA you would have to save your example code into separate files, compile it manually and then finally load both the source code and the generated image from your main document. MATRYOSHKA simplified this process. Just write the example code directly into your document and use a `#show` rule to show both the code and the resulting document.

````
#show <example>: it => {
  set grid.cell(inset: 1em, align: horizon)
  grid(
    columns: 2,
    gutter: 1em,
    grid.cell(stroke: 1pt, it),
    grid.cell(fill: silver, matryoshka.compile(it.text))
  )
}
```typ
#align(center, text(17pt)[
  *A fluid dynamic model
  for glacier flow*
])
#grid(columns: (1fr, 1fr), align: center)[
  Therese Tungsten \
  #link("mailto:tung@artos.edu")
][
  Dr. John Doe \
  #link("mailto:doe@artos.edu")
]
#pagebreak()
#lorem(100)
```<example>
````

```
#align(center, text(17pt)[
  *A fluid dynamic model
  for glacier flow*
])
#grid(columns: (1fr, 1fr), align:
center)[
  Therese Tungsten \
  #link("mailto:tung@artos.edu")
][
  Dr. John Doe \
  #link("mailto:doe@artos.edu")
]
#pagebreak()
#lorem(100)
```

Part II.
Usage

II.1. Using matryoshka

MATRYOSHKA is imported using

```
#import "@preview/matryoshka:0.1.0"
```

You can then use the `#compile()` and `#compile-pages()` commands to render Typst code.

While `#compile()` returns `content` directly, `#compile-pages()` returns an `array`, which can be used when more control over how the pages are displayed is needed.

```
#matryoshka.compile("= Hello World")
```

Because pages are `image` elements they are affected by `image` set and show rules.

```
#set image(width: 3cm)
#matryoshka.compile("= Hello World")
```

Note that in contrast to a normal Typst compiler, MATRYOSHKA automatically uses a page height of `auto`. You can change this using a set rule in the code you want to compile.

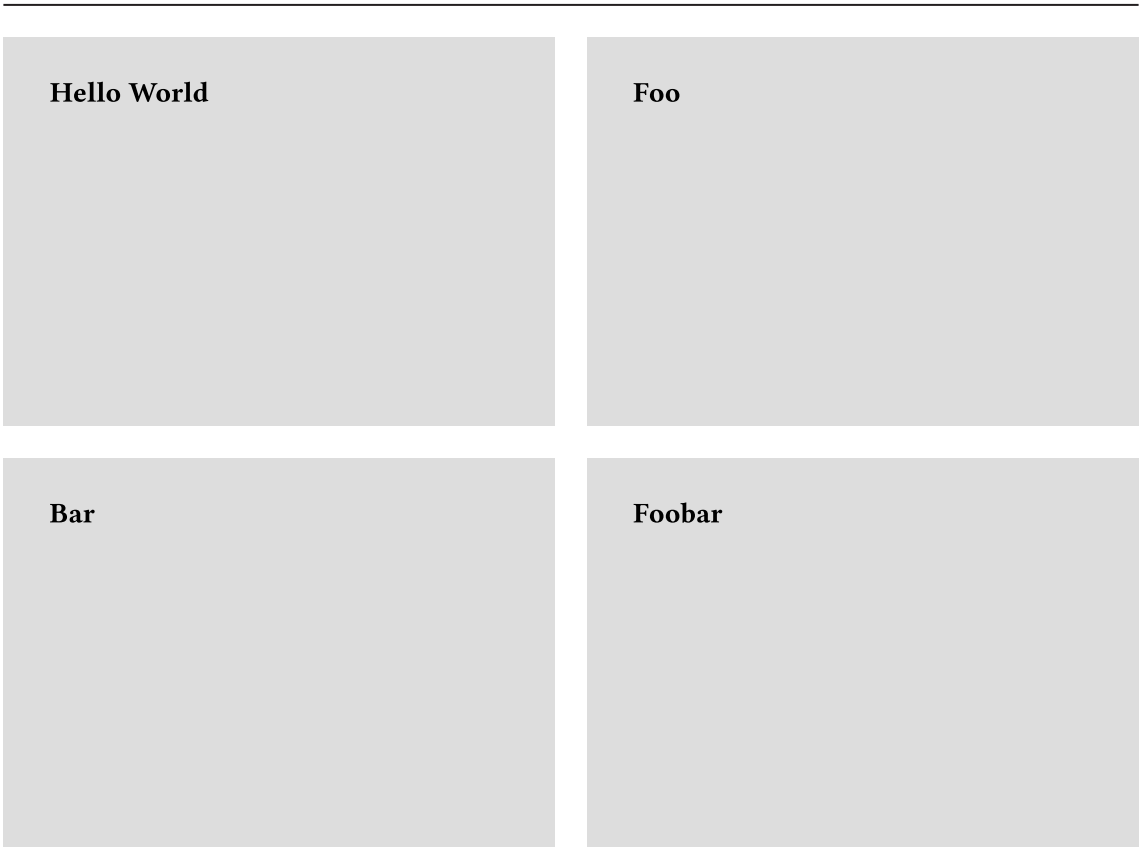
II.2. Available Commands

```
#compile()                #compile-pages()
```

`#compile-pages``(`⟨**`source`**⟩`, ⟨`**`filesystem`**⟩`: `"(:)"`, ⟨`**`dont-fail`**⟩`: `false`) ⟶ content`

Takes typst source and returns the compiled document as an array of pages.

```
grid(
  columns: 2,
  gutter: 1em,
  ..matryoshka.compile-pages("
    #set page(paper: \"a7\", flipped: true, fill: silver)
    = Hello World
    #pagebreak()
    = Foo
    #pagebreak()
    = Bar
    #pagebreak()
    = Foobar
  "),
)
```



Argument

⟨source⟩ `str`

The typst source to compile

Argument

⟨filesystem⟩: "(:)" `dictionary`

A dictionary from file-paths to file contents as strings or bytes. These files will be available to the source code. Doesn't yet support nested paths.

Argument

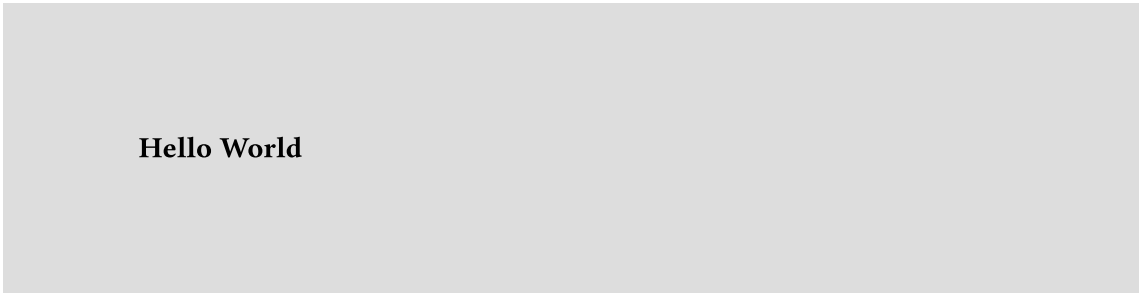
⟨dont-fail⟩: false `bool`

When set to true and compilation fails this function will return the error message instead of panicking.

`#compile(⟨source⟩, ⟨filesystem⟩: "(:)", ⟨dont-fail⟩: false)` → `content`

Takes typst source and returns the compiled documents This function returns opaque content. If you need more control use compile-pages instead.

```
matryoshka.compile("
  #set page(fill: silver)
  = Hello World
")
```



Argument

⟨source⟩ `str`

The typst source to compile

Argument

⟨filesystem⟩: "(:)" `dictionary`

A dictionary from file-paths to file contents as strings or bytes. These files will be available to the source code. Doesn't yet support nested paths.

Argument

⟨dont-fail⟩: false `bool`

When set to true and compilation fails this function will return the error message instead of panicking.

Part III.
Index

C