Introduction

Welcome to the documentation for Otter Docs 0.1.0.

Writing documentation is a lame task. It is even more boring and frustrating when you have to setup toolchains and environments and debug for hours to make sure that they build correctly, only to find that the current tools cannot plot your diagrams, or the PDF generation is missing fonts and takes hours to build. So here's Otter Docs. A simple tool that has a single requirement: Typst.

Typst is good at generating PDFs. It also has experimental support for HTML files. Since #7964 Typst can export multiple files for a single project, which makes it ideal for stuff like documentation sites.

Otter Docs uses the bundle target. You can make a new project in Typst, set it to bundle export, and it would generate a site for you. It's only a typst template, and you don't need to worry about setting up the toolchain – Typst is the only tool required.

An Unfinished Project

Otter Docs is a decent choice for organizing long, comprehensive documentation. But just like Typst, Otter Docs is an unfinished project, and is (currently) not a serious tool. Specifically, it's missing these stuff:

- Search engine optimizations (SEO)
- HTML minification; useful for reducing the page size without losing content
- Proper footnote support

However, if you want pure Typst documentation, ease of use, and/or MathML formulae, you might want to give it a try. If you want stability and extremely easy syntax, then maybe you should consider mdBook. If you have any issues, please feel free to open a ticket on GitHub. If you would like to contribute, please open a pull request.

Installation

Because Otter Docs is a pure Typst framework, you do not need to install other tools. No Python, no shell scripts, only Typst. However, there are some caveats. Otter Docs is not currently available on the Typst Universe, and it requires a very specific version of Typst. You would, unfortunately, have to build the compiler yourself. If you are a Nix user, you can use the `flake.nix` file in the repository to setup the environment.

Once Typst 0.15 releases and Otter Docs makes its way to the Universe, the installation process should just be a simple one liner like `#import "@preview/otter-docs:0.1.0": *`.

Licensing

The source and the documentation are available under Apache License v2.0.

Tutorial

Otter Docs uses Typst. If you are not familiar with Typst, you can first take a look at Typst's official tutorial for info on how to write Typst.

Setting Up

If you've used a tool like Shiroa or mdBook, you might be familiar with `book.typ` or `summary.md`. Both files contain metadata and instructions on how to organize the book. In Otter Docs, all of that is concentrated to a single entrypoint – the `book` function:

```
// Always remember to import the package
// NOTICE: OTTER DOCS IS NOT PUBLISHED TO THE TYPST UNIVERSE YET
// THIS LINE WOULD NOT WORK
#import "@preview/otter-docs:0.1.0": *

#book(
  // The routing root. Useful when you are deploying to a folder
  // under your root (e.g. when deployed to GitHub Pages)
  root: "otter-docs",
  // Your document's contents
  tree: (
    // You can add arbitrary content. The content would be displayed
    // in the summary, but would not generate html pages.
    [= Introduction],
    // This would create otter-docs/index.html. The content of the
    // chapter would be from `doc/intro.typ`
    chapter("index", content: include "doc/intro.typ"),
    // This would create otter-docs/doc/tutorial.html. In this case,
    // the content of the chapter is not explicitly stated, so it
    // look into ./doc/tutorial.typ in the current workspace.
    chapter("doc/tutorial"),
    // you can add more chapters here.
  )
)
```

Building with the Command Line

You can then compile using the following arguments in your command line. Remember to replace `<input-file>` with your actual filename

```
$ typst compile --features bundle,html <input-file> --format bundle ./dist
```

This would create the documentation in the `./dist` folder. Additionally, you can change `compile` to `watch` and add the `--open` flag at the end to preview your document in real time. You can edit your .typ source files, and any edits would be reflected immediately.

Building with the Web App

(The typst.app web app currently does not support the bundle target and MathML exports).

Authoring

Each chapter should start with a `title`. The title of the page would also be displayed in the summary. Do not start your document with a level 1 heading that looks like this: `= Heading`. Instead, write this:

```
#title[My amazing document]

= Heading 1

Over seas from coast to coast

= Heading 2
```

```
To find a place I love the most
```

The rest of your document is just normal Typst content. You could apply show rules or use functions, just like writing a normal Typst document.

Contexual Output

You can write different content for different targets, for example, embedding a link to a page via `iframe` in the HTML target.

```
#context if target() == "html" {
  // target IS HTML. Write some HTML specific stuff
} else {
  // target IS NOT HTML. It is instead "paged", which is for PDF, PNG and SVG.
  // Write some PDF specific stuff
}
```

You are currently looking at the PDF target.

Demo

Typography

Tailwind Typography and Typst helps styling `inline code`, coloured inline code: `fn foo<T: Clone>(f: fn(T) -> ())`, **bold**, *italics*, ***bold italics***, highlighted text, underlined text, overlined text, “quoted text”, “quote’s ‘effect’ in quoted text”, SmallCaps, ~~strikethrough text~~, $_{\text{subscripts}}$, $^{\text{superscripts}}$, UPPERCASE TEXT, **And a** *comprehensive* ***EXAMPLE***

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim aeque doleamus animo, cum corpore dolemus, fieri tamen permagna accessio potest, si aliquod aeternum et infinitum impendere malum nobis opinemur. Quod idem licet transferre in voluptatem, ut postea variari voluptas distinguique possit, augeri amplificarique non possit. At etiam Athenis, ut e patre audiebam facete et urbane Stoicos irridente, statua est in quo a nobis philosophia defensa et collaudata est, cum id, quod maxime placeat, facere possimus, omnis voluptas assumenda est, omnis dolor repellendus. Temporibus autem quibusdam et.

Heading 1

Heading 2

Heading 3

Heading 4

Note that these headings are not outlined, so they would not appear on the sidebar.

Framed text:

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do.

Math

Inline: the root formula is $x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$ and the Chudnovsky algorithm is $\frac{1}{\pi} = \frac{\sqrt{10005}}{4270934400} \sum_{k=0}^{\infty} \frac{(-1)^{k(6k)!}(545140134k + 13591409)}{(3k)!(k!)^3(640320)^{3k}}$ with Lorem ipsum dolor sit amet, consectetur

adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim aeque doleamus animo, cum corpore dolemus, fieri.

$$\frac{1}{\pi} = \frac{\sqrt{10005}}{4270934400} \sum_{k=0}^{\infty} \frac{(-1)^{k(6k)!}(545140134k + 13591409)}{(3k)!(k!)^3(640320)^{3k}}$$

Figure 1: Click on the formula!

Lists

Unordered list:

- Foo
- Bar
- Baz
  - Fizz
  - Buzz

Numbered list:

1. Foo
2. Bar
3. Baz
   1. Fizz
   2. Buzz

Prime Number Table

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

CJK

[illegible] LED [illegible] [illegible] 240 [illegible]

CJK Vertical Layout

Unfortunately Paged target doesn't support vertical layout yet.

Code

The source code of this example:

```
#title[Demo]

= Typography

Tailwind Typography and Typst helps styling `inline code`, coloured inline code:
```rust fn foo<T: Clone>(f: fn(T) -> ())```, *bold*, _italics_, *_bold italics_*,
#highlight[highlighted text],
#underline[underlined text], #overline[overlined text], "quoted text", "quote's
'effect' in quoted text",
#smallcaps[SmallCaps], #strike[strikethrough text], #sub[subscripts],
#super[superscripts], #upper[uppercase text],
#highlight(underline(
  overline[*And* a #sub(
      smallcaps(strike["_comprehensive_"]),
    ) #super(upper[*_example_*])],
))

#lorem(100)

#for i in range(4) {
  heading(level: i + 1, [Heading #(i + 1)], outlined: false)
}

Note that these headings are not outlined, so they would not appear on the sidebar.

Framed text:

#html.frame[#lorem(10)]

= Math

Inline: the root formula is $x = (-b plus.minus sqrt(b^2 - 4 a c))/(2a)$ and the
Chudnovsky algorithm is
$1/pi = sqrt(10005)/4270934400 sum^oo_(k=0) ((-1)^k(6k)!(545140134k + 13591409))/
((3k)!(k!)^3(640320)^(3k))$ with
#lorem(30)

#link(
  "https://en.wikipedia.org/wiki/Chudnovsky_algorithm",
  figure(
    caption: [Click on the formula!],
    $
      1/pi = sqrt(10005)/4270934400 sum^oo_(k=0) ((-1)^k(6k)!(545140134k +
13591409))/((3k)!(k!)^3(640320)^(3k))
    $,
  ),
)

= Lists
```
```

```
Unordered list:

- Foo
- Bar
- Baz
  - Fizz
  - Buzz

Numbered list:

+ Foo
+ Bar
+ Baz
  + Fizz
  + Buzz

= Prime Number Table

#let is-prime(n) = {
  if n <= 1 {
    return false
  }
  if n == 2 {
    return true
  }
  if calc.rem(n, 2) == 0 {
    return false
  }

  let limit = int(calc.sqrt(n)) + 1
  for i in range(3, limit, step: 2) {
    if calc.rem(n, i) == 0 { return false }
  }
  return true
}

#table(
  columns: (2em,) * 10,
  rows: (2em,) * 10,
  ..range(100)
    .map(it => it + 1)
    .map(n => if is-prime(n) { table.cell(fill: yellow, highlight(fill: yellow)
[#n]) } else { [#n] })
)

= CJK

[illegible]
[illegible]
LED [illegible] [illegible] 240
[illegible]

= CJK Vertical Layout

#context if target() == "html" {
  [Please note that Typst itself doesn't yet support vertical layout. The following
```

```
is done by using the
    `writing-mode: vertical-rl` CSS property.]
  html.div(
    class: "[writing-mode:vertical-rl] max-h-[70vh] w-full overflow-x-auto prose-
p:mt-0 prose-p:mb-0 prose-p:ml-2",
  )[
    #set heading(outlined: false)

    #link(
      "https://zh.wikisource.org/zh/中華民國臨時約法",
    )[= 中華民國臨時約法]

    == 第一章 總綱

    第一條 中華民國由中華人民組織之。

    第二條 中華民國之主權屬於國民全體。

    第三條 中華民國領土，爲二十二行省、內外蒙古、西藏、青海。

    第四條 中華民國以參議院、臨時大總統、國務院、法院行使其統治權。

    == 第二章 人民

    第五條 中華民國人民一律平等，無種族、階級、宗教之區別。

    第六條 人民得享有左列各項之自由權：\
      一、人民之身體，非依法律不得逮捕、拘禁、審問、處罰。\
      二、人民之家宅，非依法律不得侵入或搜索。\
      三、人民有保有財產及營業之自由。\
      四、人民有言論、著作、刊行及集會結社之自由。\
      五、人民有書信秘密之自由。\
      六、人民有居住遷徙之自由。\
      七、人民有信教之自由。

    第七條 人民有請願於議會之權。

    第八條 人民有陳訴於行政官署之權。

    第九條 人民有訴訟於法院受其審判之權。

    第十條 人民對於官吏違法損害權利之行爲，有陳訴於平政院之權。

    第十一條 人民有應任官考試之權。

    第十二條 人民有選舉及被選舉之權。

    第十三條 人民依法律有納稅之義務。

    第十四條 人民依法律有服兵之義務。

    第十五條 本章所載人民之權利，有認爲增進公益、維持治安或非常緊急必要時，得依法律限制之。

    == 第三章 參議院

    第十六條 中華民國之立法權，以參議院行之。
```

第十七條 參議院以第十八條所定各地方選派之參議員組織之。

第十八條 參議員每行省、內蒙古、外蒙古、西藏各選派五人，青海選派一人。其選派方法，由各地方自定之。參議院會議時，每參議員有一表決權。

第十九條 參議院之職權如左：\
一、議決一切法律案。\
二、議決臨時政府之豫算、決算。\
三、議決全國之稅法、幣制及度量衡之準則。\
四、議決公債之募集及國庫有負擔之契約。\
五、承諾第三十四條、三十五條、四十條事件。\
六、答覆臨時政府諮詢事件。\
七、受理人民之請願。\
八、得以關於法律及其他事件之意見建議於政府。\
九、得提出質問書於國務員，並要求其出席答覆。\
十、得咨請臨時政府查辦官吏納賄違法事件。\
十一、參議院對於臨時大總統認爲有謀叛行爲時，得以總員五分四以上之出席，出席員四分三以上之可決，彈劾之。\
十二、參議院對於國務員認爲失職或違法時，得以總員四分三以上之出席，出席員三分二以上之可決，彈劾之。

第二十條 參議院得自行集會、開會、閉會。

第二十一條 參議院之會議，須公開之。但有國務員之要求，或出席參議員過半數之可決者，得秘密之。

第二十二條 參議院議決事件，咨由臨時大總統公布施行。

第二十三條 臨時大總統對於參議院議決事件，如否認時，得於咨達後十□內聲明理由，咨院覆議。但參議院對於覆議事件，如有到會參議員三分二以上仍執前議時，仍照第二十二條辦理。

第二十四條 參議院議長，由參議員用記名投票法互選之，以得票滿投票總數之半者爲當選。

第二十五條 參議院參議員於院內之言論及表決，對於院外不負責任。

第二十六條 參議院參議員除現行犯及關於內亂、外患之犯罪外，會期中非得本院許可，不得逮捕。

第二十七條 參議院法，由參議院自定之。

第二十八條 參議院以國會成立之□解散，其職權由國會行之。

（下略）

```
  ]
} else [
  Unfortunately Paged target doesn't support vertical layout yet.
]
```

= Code

The source code of this example:

```
#raw(block: true, lang: "typ", read("./demo.typ"))
```

Changelog

Otter Docs Changelog

This is the change log for Otter Docs, a pure Typst documentation framework.

UNRELEASED

- Basic documentation features
- Typst template