

# Hello, Astro With Typst

## 目录

前言 .....	1
数学 .....	1
图片 .....	1
样式 .....	2

## 前言

我有多久没发新的博客文章了？4 个月了。总之这段时间里，一月在忙着旅游，二月玩游戏玩疯了，三月在准备考研复试，结果初试分数不够美美下岸（悲）。总之这篇文章写在四月，目前正在准备考研调剂。

除了上面说的这些东西以外，我这些日子里还在忙着折腾新的博客主题——我想换一个花样：用 Typst 来写博客文章。Typst 是一个新兴的文档排版工具，在其 0.14 版本中有了比较够用的实验性 HTML 导出支持，虽然目前还不够完善，但我觉得已经足够用来写博客文章了。

## 数学

得益于 `typst.ts` 与 `astro-typst` 项目的存在，在 Astro 里使用 Typst 并不是一件难事。Typst 目前不支持数学公式的 HTML 导出——因为开发者认为数学公式应当被导出为 MathML 以符合语义、可访问性等要求。导出为 SVG 并不难，但是我不想这么做，这意味着我要自行处理基线问题，实在麻烦。理论上，你可以通过 `show: rule` 定义数学公式在 HTML 导出时的行为来自行实现 MathML 导出功能，也确实有人这么做了。`mathyml` 就是这么一个 Typst 库。这个库并不完善，只能对 Typst 原生的数学元素进行处理，但是对于博客写作已经基本够用了。

MathML 也有问题——各个浏览器的支持程度不一，默认显示效果不同。显示效果可以用网络字体 + CSS 来解决，`mathyml` 的解决方案是使用 GitHub 上的网络数学字体。我不太喜欢依赖 GitHub 提供服务，但是我姑且没找到一个好的方式来将数学字体整合到 Astro 6 自带的字体优化功能中。

## 图片

与此同时，用 Typst 来写博客还面临着另一个问题：图片。显然 Typst 并不默认支持使用在线图片。你当然可以通过 `#html.elem()` 来写一个 `<img>` 标签引入在线图片，但它不可能在 PDF 导出版本中使用。而我，本着「我都用 Typst 了」的想法，非常希望我的新主题中有 PDF 归档功能，所以在线图片这条路走不通了。那么本地图片呢？Typst 很好地支持了导入本地图片，只是有一些小小的问题：Typst 目前不支持多文件的 HTML 导出，于是所有的图片都是内联的 Base64 格式，让图像优化变成笑话。而且我有部分文章（旅游相关）基本全是图片，这显然是不可接受的。

万幸的是，`astro-typst` 项目通过一些技巧，支持在 Typst 导出时嵌入 JSX 代码，于是我可以使使用 Astro 自带的 `<Image />` 组件来分离图像并自动优化了……吗？又一个坑出现了：在 Typst 0.14 版本中，所有图像的 `path` 属性都是普通的字符串。与此同时，JSX 的导入路径处理的是 Typst 的入口文件，它不知道 Typst 文件的实际组织结构——这意味着只要有如下结构的 Typst 文件：

```
1 my-post/  
2   └─ index.typ  
3     └─ part/
```

```
4   └─ part1.typ
5   └─ image.png
```

在任何地方获取到的 `part1.typ` 中导入的 `image.png` 的导入路径都是 `./image.png`，而不是 `./part/image.png`，这就导致了 Astro 无法正确地处理图片的导入路径了。这个问题似乎不难解决：只需要在 Typst 文件中使用绝对路径来导入图片就好了。但是，Typst 包里的图像呢？在任何地方你都无法分辨一个路径是来自包的还是来自本地的，因为包里也可以使用绝对路径导入图像。这时候聪明的小伙伴们就想起来，我在提及 `path` 的类型时，特意添加了一个限定条件：Typst 0.14 版本。事实上，在 Typst 的主线版本中，文件路径 `path` 有了自己的独立类型，如果 Typst 0.15 版本没有解决多文件导出功能，我们也可以通过这个独立的路径类型来区分包内路径和本地路径，从而通过 JSX 来正确地处理图片了。

但 Typst 0.15 版本似乎一时半会儿不会发布，那现在该怎么办？答案是后处理。在 Astro 的构建过程中，我们可以通过 Astro 的构建钩子 来对构建产物进行后处理。我们可以在构建完成后扫描所有的 HTML 文件，找到其中的 `<img>` 标签，并将其 `src` 属性中的 Base64 数据提取出来，存放到构建产物中，并将 `src` 替换为正确的图片路径。这个过程虽然有点麻烦，但至少可以解决当前的问题了。

技术问题总体上是解决了，但（这都多少个「但」子）在 Git 仓库中存储大量本地图片并不优雅，会让你的 Git 仓库体积增大 114514 倍。优雅的方式自然是 Git LFS，可是我个人的开发工具链里的版本控制系统是 Jujutsu，其非常不幸地不支持 Git LFS，所以把博客文章迁移到 Typst 的工作就只能一拖再拖。好消息是现在似乎有个维护者愿意接受的 PR 能部分解决这个问题，所以迁移工作未来可期。

## 样式

如果只是改成用 Typst 写文章，倒也不必另搓个新主题。所以，我不只是想换个写作工具，还想换个主题。人的品味是会随着时间变化的，而我现在更喜欢极简风格的主题。我自己确实没啥设计能力，所以我选择抄一个我喜欢的极简风格的主题——Tufted CSS。脚注是一个不好的东西，尤其是在网页这种纵向长度不限制的地方。我想要更好的脚注，就像我对优秀教材的期望一般，Tufted CSS 化脚注为边注的设计就相当合我胃口。

总之，主题是搓出来了个大概，发布在了 Codeberg。迁移工作也在进行，建了个测试站点用来观察效果。发篇文章出来宣传一下我这个测试站点，顺便证明一下我人还没死，博客也没放弃。