制作模板

在本教程的前三章中，你已经学会了如何在 Typst 中编写文档、应用基本样式和深入定制其外观以遵循出版商的样式指南。因为你在前一章中写的论文取得了巨大的成功，你被要求为同一场会议撰写续作。这一次，你想将你在前一章创建的风格转换为可重复使用的模板。在本章中，你将学习如何创建一个仅需一个 show 规则即可使用的模板。让我们开始吧！

玩具模板（Toy Template）

在 Typst 中，模板是可以包裹整个文档的函数。为了了解如何做到这一点，让我们首先回顾一下如何编写自己的函数。它们可以执行你想要的任何操作，所以为什么不稍微变得有点疯狂呢？

```
#let amazed(term) = box[✨ #term ✨]

You are #amazed[beautiful]!
```

You are ✨ beautiful ✨!

这个函数接受一个参数 term，并返回一个带有星号包围的 term 的内容块。我们还将整个内容放入一个框中，以防止我们所惊叹的 term 被换行符与星号分离。

许多附带的 Typst 函数都有可选的命名参数。我们的函数也可以拥有它们。让我们向我们的函数添加一个参数，使我们能够选择文本的颜色。如果未提供参数，则需要提供默认颜色。

```
#let amazed(term, color: blue) = {
  text(color, box[✨ #term ✨])
}

You are #amazed[beautiful]!
I am #amazed(color: purple)[amazed]!
```

You are ✨ beautiful ✨! I am ✨ amazed ✨!

现在，我们使用 *"everything"* show 规则将自定义函数应用于整个文档，并以此来创建模板。让我们使用我们的 amazed 函数来做这件事。

```
#show: amazed
I choose to focus on the good
in my life and let go of any
negative thoughts or beliefs.
In fact, I am amazing!
```

✨ I choose to focus on the good in my life and let go of any negative thoughts or beliefs. In fact, I am amazing! ✨

我们的整个文档现在将被传递给 amazed 函数，就像我们将其包装在其周围一样。这在使用此特定函数时可能不太有用，但是当结合 set 规则和命名参数使用时，它可以非常强大。

嵌入 set 和 show 规则

为了将一些 set 和 show 规则应用到我们的模板中，我们可以在函数中的内容块中使用 set 和 show，然后将文档插入到该内容块中。

```typ
#let template(doc) = [
  #set text(font: "Inria Serif")
  #show "something cool": [Typst]
  #doc
]

#show: template
I am learning something cool today.
It's going great so far!
```

I am learning Typst today. It's going great so far!

就像我们在上一章中已经发现的一样，`set` 规则将应用于它们的内容块中的所有内容。由于*"everything"* `show` 规则将整个文档传递给 `template` 函数，因此在模板中的文本设定规则和字符串显示规则将应用于整个文档。利用这个知识，我们可以创建一个模板，以复制我们在上一章中撰写的论文的正文样式。

```typ
#let conf(title, doc) = {
  set page(
    paper: "us-letter",
    header: align(
      right + horizon,
      title
    ),
    ...
  )
  set par(justify: true)
  set text(
    font: "Linux Libertine",
    size: 11pt,
  )

  // Heading show rules.
  ...

  columns(2, doc)
}

#show: doc => conf(
  [Paper title],
  doc,
)

= Introduction
#lorem(90)
```

...


Paper title

Introduction

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.

Motivation. 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 aut officiis debitis aut rerum necessitatibus saepe eveniet, ut et voluptates repudiandae sint et molestiae non recusandae. Itaque earum rerum defuturum, quas natura non depravata desiderat. Et quem ad me accedis, saluto: 'chaere,' inquam, 'Tite!' lictores, turma omnis chorusque: 'chaere.

Problem Statement. 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.

Related Work

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 aut officiis debitis aut rerum necessitatibus saepe eveniet, ut et voluptates repudiandae sint et molestiae non recusandae. Itaque earum rerum defuturum, quas natura non depravata desiderat. Et quem ad me accedis, saluto: 'chaere,' inquam, 'Tite!' lictores, turma omnis chorusque: 'chaere, Tite!' hinc hostis mi Albucius, hinc inimicus. Sed iure Mucius. Ego autem mirari satis non queo unde hoc sit tam insolens domesticarum rerum fastidium. Non est omnino hic docendi locus; sed ita prorsus existimo, neque eum Torquatum, qui hoc primus cognomen invenerit, aut torquem illum hosti detraxisse, ut aliquam ex eo est consecutus? – Laudem et caritatem, quae sunt vitae.

我们从上一章复制了大部分代码。唯一的两个区别是：我们把所有内容都封装在了 `conf` 函数中，并且由于 `doc` 参数已经包含了文档的内容，我们直接在它上面调用了 `columns` 函数。此外，我们使用了大括号包围的代码块，而不是内容块。这样，我们就无需在所有的 `set` 规则和函数调用前添加 `#` 符号。作为折衷，我们不能再直接在其中编写标记了。

还请注意标题是如何来的：我们之前是将标题放在一个变量里的。现在，它是作为模板函数的第一个参数传入的。因此，在我们调用模板的 `show` 规则时，我们必须在那里指定标题。

带有命名参数的模板

上一章节的论文包括了标题和作者列表。现在，让我们将这些元素加入到我们的模板中。除了标题之外，我们希望模板还能够接受包含作者隶属信息和论文摘要的作者列表作为输入。为了保持代码的可读性，我们将这些信息作为命名参数加入。最终，我们期望它能够这样工作：

```
#show: doc => conf(
  title: [Towards Improved Modelling],
  authors: (
    (
      name: "Theresa Tungsten",
      affiliation: "Artos Institute",
      email: "tung@artos.edu",
    ),
    (
      name: "Eugene Deklan",
      affiliation: "Honduras State",
      email: "e.deklan@hstate.hn",
    ),
  ),
  abstract: lorem(80),
  doc,
)

...
```

让我们构建这个新的模板函数。首先，我们为标题参数添加了一个默认值。这样，我们就可以在不指定标题的情况下调用模板。我们还添加了名为 `authors` 和 `abstract` 的命名参数，并设置了空的默认值。接下来，我们复制了上一章中生成标题、摘要和作者列表的代码到模板中，用参数替换了固定的详细信息。

新的 `authors` 参数期望是一个包含字典的数组，字典的键包括 `name`（姓名）、`affiliation`（隶属机构）和 `email`（电子邮件）。因为我们可能有任意数量的作者，我们动态地确定是否需要一个、两个或三个列来显示作者列表。首先，我们使用 `authors` 数组上的`.len()`方法来确定作者的数量。然后，我们将列的数量设置为这个计数与三的最小值，这样我们就永远不会创建超过三列。如果有超过三位作者，就会插入新的一行。为此，我们还向 `grid` 函数添加了一个 `row-gutter` 参数。否则，行之间的距离会太近。要提取字典中关于作者的详细信息，我们使用字段访问语法（*field access syntax*）。

我们仍然需要为每个作者提供 `grid` 的一个参数：这里就是数组的 `map` 方法发挥作用的地方。它接受一个函数作为参数，该函数将被调用，用于数组的每个项目。我们传递给它一个格式化每个作者详细信息的函数，并返回一个包含内容值的新数组。现在我们得到了一个值的数组，我们希望将其作为多个参数用于 `grid`。我们可以通过使用展开运算符（*spread operator*）来做到这一点。它接受一个数组，并将其中的每个项目作为单独的参数应用到函数中。

生成的模板函数看起来像这样：

```
#let conf(
  title: none,
  authors: (),
  abstract: [],
  doc,
) = {
  // Set and show rules from before.
  ...
```

```
  set align(center)
  text(17pt, title)

  let count = authors.len()
  let ncols = calc.min(count, 3)
  grid(
    columns: (1fr,) * ncols,
    row-gutter: 24pt,
    ..authors.map(author => [
      #author.name \
      #author.affiliation \
      #link("mailto:" + author.email)
    ]),
  )

  par(justify: false)[
    *Abstract* \
    #abstract
  ]

  set align(left)
  columns(2, doc)
}
```

独立文件

大多数时候，模板是在一个不同的文件中定义的，然后导入到文档中。这样做可以保持你主文档的整洁，并且可以轻松地重用模板。通过点击文件面板上的加号按钮创建一个新的文本文件，并将其命名为 `conf.typ`。将 `conf` 函数的定义移动到这个新文件中。现在，你可以通过在 `show` 规则前添加一个 `import` 语句，从你的主文件中访问它。在 `import` 关键字和冒号之间指定文件的路径，然后命名你想要导入的函数。

```
#import "conf.typ": conf
#show: doc => conf(
  title: [
    Towards Improved Modelling
  ],
  authors: (
    (
      name: "Theresa Tungsten",
      affiliation: "Artos Institute",
      email: "tung@artos.edu",
    ),
    (
      name: "Eugene Deklan",
      affiliation: "Honduras State",
      email: "e.deklan@hstate.hn",
    ),
  ),
  abstract: lorem(80),
  doc,
)

= Introduction
#lorem(90)

== Motivation
#lorem(140)

== Problem Statement
#lorem(50)

= Related Work
#lorem(200)
```

Towards Improved Modelling

Towards Improved Modelling

Theresa Tungsten
Artos Institute
tung@artos.edu

Eugene Deklan
Honduras State
e.deklan@hstate.hn

Abstract

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.

Introduction

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.

Motivation. 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 aut officiis debitis aut rerum necessitatibus saepe eveniet, ut et voluptates repudiandae sint et molestiae non recusandae. Itaque earum rerum defuturum, quas natura non depravata desiderat. Et quem ad me accedis, saluto: 'chaere,' inquam, 'Tite!' lictores, turma omnis chorusque: 'chaere.

Problem Statement. 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.

Related Work

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 aut officiis debitis aut rerum necessitatibus saepe eveniet, ut et voluptates repudiandae sint et molestiae non recusan-

1

我们现在已将会议论文转换为该会议的可重复使用的模板！为什么不将它分享到 Typst Discord 服务器上，这样其他人也可以用呢？

回顾 Review

恭喜，您已经完成了 Typst 的教程！在本节中，您学会了如何定义自己的函数以及如何创建和应用定义可重复使用文档样式的模板。您已经走得很远，学到了很多。现在，您可以使用 Typst 来编写自己的文档并与他人分享。

我们仍然是一个非常年轻的项目，正在寻求反馈。如果您有任何问题、建议或发现了错误，请通过 Typst Discord 服务器、我们的联系表单或社交媒体告知我们。

那么，您还在等什么呢？注册并写一些东西吧！