

## 把文档格式化处理

到目前为止，你已经写了一份包含一些文字、几个方程式和图片的报告。然而，它看起来仍然非常朴素。你的助教还不知道你正在使用一个新的排版系统，而你希望你的报告能够与其他学生的提交作业相匹配。在本章中，我们将学习如何使用 Typst 的样式系统来格式化你的报告。

### 设定规则

正如我们在上一章中所看到的，Typst 有一些函数用于插入内容（例如 `image` 函数），还有一些函数用于操作接收到的内容参数（例如 `align` 函数）。当你想要，比如说，使报告两端对齐时，你的第一反应可能是寻找一个能够实现这一功能的函数，并用它包裹整个文档。

```
#par(justify: true)[  
  = Background  
  In the case of glaciers, fluid  
  dynamics principles can be used  
  to understand how the movement  
  and behaviour of the ice is  
  influenced by factors such as  
  temperature, pressure, and the  
  presence of other fluids (such as  
  water).  
]
```

### Background

In the case of glaciers, fluid dynamics principles can be used to understand how the movement and behaviour of the ice is influenced by factors such as temperature, pressure, and the presence of other fluids (such as water).

等等，函数的所有参数不是应该都在圆括号内指定吗？为什么在圆括号之后还有一组方括号包含内容？答案是，由于在 Typst 中向函数传递内容是如此常见的操作，因此有一种特殊的语法：不必将内容放在参数列表内，你可以直接在常规参数后的方括号中写入内容，这样可以节省标点符号。

如上所示，这种方法确实有效。`par` 函数使其中的所有段落两端对齐。然而，用无数函数包裹文档并选择性地在原地应用样式很快就会变得繁琐。

幸运的是，Typst 有一个更优雅的解决方案。通过 `设置规则`，你可以将样式属性应用于所有出现的内容上。要写一个设置规则，你需要输入 `set` 关键字，后跟你想要设置属性的函数名，然后在圆括号中列出参数。

```
#set par(justify: true)  
  
= Background  
In the case of glaciers, fluid  
dynamics principles can be used  
to understand how the movement  
and behaviour of the ice is  
influenced by factors such as  
temperature, pressure, and the  
presence of other fluids (such as  
water).
```

### Background

In the case of glaciers, fluid dynamics principles can be used to understand how the movement and behaviour of the ice is influenced by factors such as temperature, pressure, and the presence of other fluids (such as water).

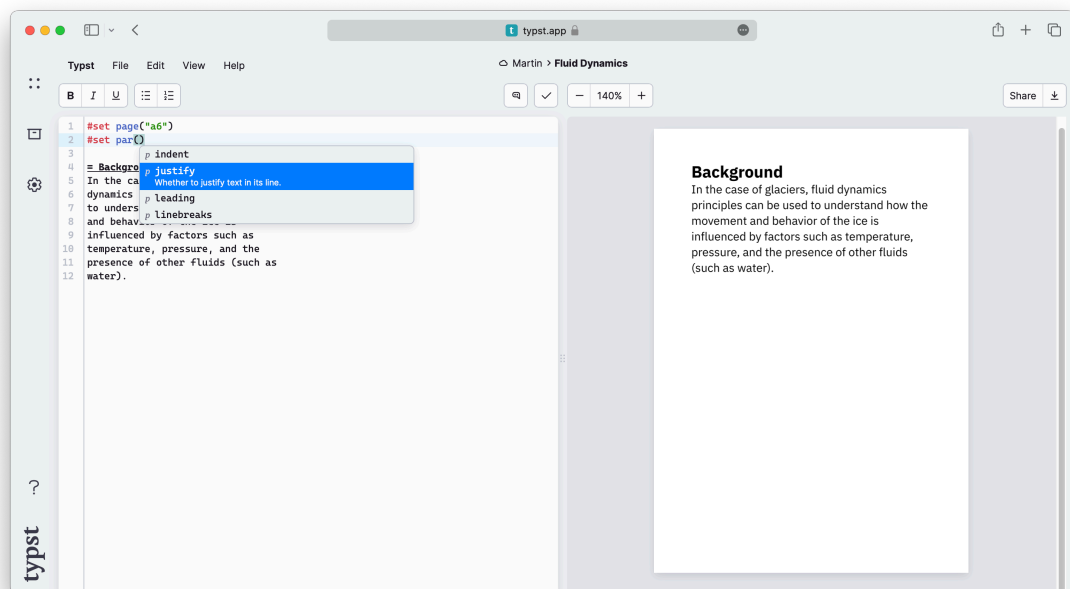
#### 信息 INFO

想要用更专业的术语了解这里发生了什么吗？

设置规则可以被理解为对某个函数的部分参数设定默认值， 这些默认值将应用于该函数的所有后续使用中。

## 自动完成面板

如果您按照说明操作并尝试了应用中的一些功能，可能您已经注意到，每当输入一个 `#` 字符时，都会弹出一个面板，显示可用的函数，以及在参数列表中可用的参数。这就是自动完成面板。在撰写文档时，它可以非常有用：您可以通过按 `Return` 键应用其建议，或者使用箭头键导航到所需的完成项。可以通过按 `Escape` 键关闭面板，通过输入 `#` 或按下 `(Ctrl) + (Space)` 再次打开。请使用自动完成面板发现函数的正确参数。大多数建议都附有一小段描述，说明它们的作用。



## 页面设定

返回设定规则的部分：编写规则时，您可以根据要设置样式的元素类型选择函数。以下是一些常用的在设置规则中使用的函数的列表：

- `text` 用于设置文本的字体族、大小、颜色和其他属性
- `page` 用于设置页面大小、边距、标题、启用列和页脚
- `par` 用于对齐段落、设置行距等
- `heading` 用于设置标题的外观并启用编号
- `document` 用于设置 PDF 输出中包含的元数据，例如标题和作者

并非所有函数参数都可以设置。通常，只有告诉函数如何执行操作的参数才可以设置，而不是告诉它用什么来执行的参数。函数参考页面会指示哪些参数是可以设置的。

让我们为我们的文档添加一些更多的样式。我们想要更大的边距和衬线字体。为了演示目的，我们还将设置另一个页面大小。

```

#set text(
  font: "New Computer Modern",
  size: 10pt
)
#set page(
  paper: "a6",
  margin: (x: 1.8cm, y: 1.5cm),
)
#set par(
  justify: true,
  leading: 0.52em,
)

= Introduction
In this report, we will explore the
various factors that influence fluid
dynamics in glaciers and how they
contribute to the formation and
behaviour of these natural structures.

...

#align(center + bottom)[
  #image("glacier.jpg", width: 70%)

  *Glaciers form an important
  part of the earth's climate
  system.*
]

```

## Introduction

In this report, we will explore the various factors that influence fluid dynamics in glaciers and how they contribute to the formation and behaviour of these natural structures.

Glacier displacement is influenced by a number of factors, including

1. The climate
2. The topography
3. The geology

This report will present a physical model of glacier displacement and dynamics, and will explore the influence of these factors on the movement of large bodies of ice.



Glaciers form an important part of the earth's climate system.

这里有一些需要注意的事项。

首先是[设置页面](#)的规则。它接收两个参数：页面大小和页面边距。页面大小是一个字符串。Typst 接受许多[标准页面大小](#)，但您也可以指定自定义页面大小。边距以[字典](#)的形式指定。字典是键值对的集合。在这种情况下，键是 `x` 和 `y`，值分别是水平和垂直边距。我们也可以通过传递一个具有键 `left`、`right`、`top` 和 `bottom` 的字典来指定每边的单独边距。

接下来是[设置文本](#)的规则。在这里，我们将字体大小设置为 `10pt`，字体族设置为 `"New Computer Modern"`。Typst 应用程序附带的许多可供您的文档使用的字体。当您处于文本函数的参数列表中时，可以在自动完成面板中发现可用的字体。

我们还设置了行间距（也称为前导）：它以[长度值](#)（`length`）的形式指定，我们使用 `em` 单位来相对于字体大小指定前导：`1em` 等于当前字体大小（默认为 `11pt`）。

最后，我们通过向我们的中心对齐添加垂直对齐来底部对齐我们的图像。垂直和水平对齐可以使用 + 运算符组合以获得 2D 对齐。

## 一点便捷

为了更清晰地组织我们的文档，我们现在想对标题进行编号。我们可以通过设置 `heading` 函数的 `numbering` 参数来实现这一点。

```
#set heading(numbering: "1.")
```

```
= Introduction
```

```
#lorem(10)
```

```
== Background
```

```
#lorem(12)
```

```
== Methods
```

```
#lorem(15)
```

## 1. Introduction

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

### 1.1. Background

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

### 1.2. Methods

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore.

我们将字符串 `"1."` 指定为 `numbering` 参数。这告诉 Typst 用阿拉伯数字对标题进行编号，并在每个级别的数字之间放置一个点。我们还可以使用字母、罗马数字和符号来对我们的标题进行编号。

```
#set heading(numbering: "1.a")
```

```
= Introduction
```

```
#lorem(10)
```

```
== Background
```

```
#lorem(12)
```

```
== Methods
```

```
#lorem(15)
```

## 1 Introduction

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

### 1.a Background

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

### 1.b Methods

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore.

这个例子还使用了 `lorem` 函数来生成一些占位符文本。这个函数接受一个数字作为参数，并生成相同数量的 *Lorem Ipsum* 文本。

### 信息 INFO

您是否想知道为什么标题和文本设置规则适用于所有文本和标题，即使它们没有使用相应的函数生成？

Typst 在您编写 `= Conclusion` 时内部调用标题函数。实际上，函数调用 `#heading[Conclusion]` 等同于上面的标题标记。其他标记元素也是如此，它们只是对应函数调用的语法糖。



## 显示规则 (Show rules)

您对这个成果已经相当满意了。但有最后一件事需要解决：您所撰写的报告是为一个更大的项目服务的，该项目的名称在文章中出现时，都应该伴随相应的 Logo。

您可以有以下选择。可以使用搜索和替换在每个 logo 实例之前添加 `#image("logo.svg")` 调用来实现。不过这样处理很繁琐。相反，您可以定义一个[自定义函数](#)，该函数始终生成带有其 logo 的内容。但是，有一个更简单的方法：

使用显示规则，您可以重新定义 Typst 显示某些元素的方式。您指定 Typst 应该如何显示哪些元素，以及它们应该看起来如何。显示规则可以应用于文本实例、函数，以及整个文档。

```
#show "ArtosFlow": name => box[
  #box(image(
    "logo.svg",
    height: 0.7em,
  ))
  #name
]
```

This report is embedded in the  ArtosFlow project.  ArtosFlow is a project of the Artos Institute.

This report is embedded in the  
ArtosFlow project. ArtosFlow is a  
project of the Artos Institute.

这个例子中包含大量新语法：我们首先键入 `show` 关键字，随后跟上希望以不同方式显示的文本字符串及冒号。接着，我们定义一个函数，该函数将要展示的内容作为参数。在此例中，我们将该参数命名为 `name`。现在，我们可以在函数体中使用 `name` 变量来输出 ArtosFlow 名称。通过我们的显示规则，将在名称前添加 logo 图片，并将结果置于一个框(box)中，以防止 logo 与名称之间出现换行。图片同样被置于框内，确保它不会单独占据一个段落。

对首个 `box` 函数及 `image` 函数的调用并未要求前置 `#` 号，原因在于它们并非直接嵌入到标记语言中。当 Typst 期望接收代码而非标记时，访问函数、关键字及变量并不需要使用领先的 `#` 号。这一规则在参数列表、函数定义以及[代码块](#)中均可见。

## 回顾

你现在已掌握在 Typst 文档中应用基础排版的方法。你学会了如何设定字体、对齐段落、调整页面尺寸，以及为标题添加编号等规范操作。此外，你还掌握了使用基础显示规则，以改变文档中文字呈现方式的技巧。

你已经提交了报告，你的上司对它非常满意，以至于他们希望将其改编成一篇会议论文！在接下来的部分里，我们将学习如何运用更高级的显示规则和功能，将你的文档精心打磨成一篇专业的学术论文。