

造型

Typst 包括一个灵活的样式系统，可自动将您选择的样式应用于您的文档。使用设置的规则，您可以配置元素的基本属性。这样，您就可以创建最常见的样式。但是，可能没有用于您希望执行的所有操作的内置属性。出于这个原因，Typst 进一步支持可以完全重新定义元素外观的显示规则。

设置规则

使用设置的规则，您可以自定义元素的外观。它们被写成对元素函数的函数调用，前面有关键字（或在标记中）。只能向设置规则提供该函数的可选参数。请参阅每个函数的文档，了解哪些参数是可选的。在下面的示例中，我们使用两个设置规则来更改字体系列和标题编号。

```
#set heading(numbering: "I.")
#set text(
  font: "New Computer Modern"
)

= Introduction
With set rules, you can style
your document.
```

I. Introduction

With set rules, you can style your document.

顶级集规则在文件末尾之前一直有效。当嵌套在块内时，它仅在该块结束之前有效。因此，使用块，您可以将规则的效果限制为文档的特定部分。下面，我们使用内容块将列表样式范围限定为一个特定列表。

```
This list is affected: #[
  #set list(marker: [--])
  - Dash
]

This one is not:
- Bullet
```

This list is affected:

- Dash

This one is not:

- Bullet

有时，您需要有条件地应用设置规则。为此，您可以使用 `set-if` 规则。

```
#let task(body, critical: false) = {
  set text(red) if critical
  [- #body]
}

#task(critical: true)[Food today?]
#task(critical: false)[Work deadline]
```

- Food today?
- Work deadline

显示规则

使用显示规则，您可以深度自定义元素类型的外观。show 规则的最基本形式是 `show-set` 规则。这样的规则被写成关键字，后跟一个选择器、一个冒号，然后是一个集合规则。选择器的最基本形式是元素函数。这样一来，设置规则仅适用于所选元素。在下面的示例中，标题变为深蓝色，而所有其他文本保持黑色。

```
#show heading: set text(navy)
```

```
= This is navy-blue  
But this stays black.
```

This is navy-blue

But this stays black.

使用显示集规则，您可以混合和匹配来自不同函数的属性，以实现许多不同的效果。但是它们仍然将您限制在 Typst 中预定义的内容。为了获得最大的灵活性，您可以改为编写一个显示规则来定义如何从头开始设置元素的格式。要编写这样的 show 规则，请将冒号后面的 set 规则替换为任意函数。此函数接收有问题的元素，并可以返回任意内容。传递给函数的元素上的可用字段再次与相应元素函数的参数匹配。下面，我们定义一个显示规则，用于格式化幻想百科全书的标题。

```
#set heading(numbering: "(I)")  
#show heading: it => [  
  #set align(center)  
  #set text(font: "Inria Serif")  
  \~ #emph(it.body)  
  #counter(heading).display(  
    it.numbering  
  ) \~  
]
```

Dragon

With a base health of 15, the dragon is the most powerful creature.

Manticore

While less powerful than the dragon, the manticore gets extra style points.

~ ***Dragon (I)*** ~

With a base health of 15, the dragon is the most powerful creature.

~ ***Manticore (II)*** ~

While less powerful than the dragon, the manticore gets extra style points.

与设置规则一样，显示规则在当前块或文件结束之前一直有效。

除了函数之外，show 规则的右侧还可以采用应直接替换元素的文本字符串或内容块。除了函数之外，show 规则的左侧还可以采用许多其他选择器，这些选择器定义将转换应用于哪些内容：

- **一切：**
在显示规则之后转换一切。这对于将更复杂的布局应用于整个文档而不将所有内容包装在一个巨大的函数调用中非常有用。`show: rest => ..`
- **文本：**
设置文本样式、转换或替换文本。`show "Text": ..`
- **正则表达式：**
使用正则表达式选择和转换文本，以获得更大的灵活性。有关详细信息，请参阅[正则表达式类型的文档](#)。`show regex("\w+"): ..`
- **带字段的函数：**
仅转换具有指定字段的元素。例如，您可能只想更改 1 级标题的样式。
`show heading.where(level: 1): ..`
- **标签：**
选择并转换具有指定标签的元素。有关更多详细信息，请参阅[标签类型的文档](#)。
`show <intro>: ..`

```
#show "Project": smallcaps  
#show "badly": "great"
```

We started Project in 2019
and are still working on it.
Project is progressing badly.

We started PROJECT in 2019 and are still
working on it. PROJECT is progressing great.