Typst □□□□

□ PDF 中，我们可以轻松改变文字的颜□，□□□或字体。

typ

```
□ PDF 中，我们可以轻松改变文字的 #text(fill: blue)[颜色]，#text(size: 20pt)[大小] 或 #text(font: "Liu Jian Mao Cao")[字体]。
```

段落对齐

文字默认是左对齐的。

但我想临时居中！

而不影响其他文字。

该段落及之后的段落都是左对齐

该段落及之后的段落都是右对齐

该段落及之后的段落都是居中对齐

我也居中了

typ

```
文字默认是左对齐的。

#align(center)[但我想临时居中！]

而不影响其他文字。

#set align(left)
该段落及之后的段落都是左对齐

#set align(right)
该段落及之后的段落都是右对齐

#set align(center)
该段落及之后的段落都是居中对齐

我也居中了
```

分栏

这是双栏布局的第一部分。Typst 会自动平衡两栏的高度，使得排版更加美观。

我们使用 `lorem` 函数来生成一些占位符文本：Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do.

这是第二栏的内容。你可以□这里放置图片、公式或□其他任何内容。即便内容较少，它们也会并排显示。

typ

```
#columns(2, gutter: 1em)[
  这是双栏布局的第一部分。Typst 会自动平衡两栏的高度，使得排版更加美观。

  我们使用 `lorem` 函数来生成一些占位符文本：
  #lorem(10)

  #colbreak() // 强制换栏

  这是第二栏的内容。你可以□这里放置图片、公式或者其他任何内容。即便内容较少，它们也会并排显示。
]
```

表格

推荐使用橘子大佬写的 `tablem` 包根据 markdown 表格格式生成表格：

Name	Location	Height	Score
John	Second St.	180 cm	5
Wally	Third Av.	160 cm	10

typ

```
#import "@preview/tablem:0.3.0": *

#align(center)[
  #three-line-table[
    | *Name* | *Location* | *Height* | *Score* |
    | :----: | :--------: | :------: | :-----: |
    | John   | Second St. | 180 cm   | 5       |
    | Wally  | Third Av.  | 160 cm   | 10      |
  ]
]
```

支持单元格合并：

Soldier	Hero		Soldier
Guard	Horizontal		Guard
	Soldier	Soldier	
Soldier	Gate		Soldier

typ

```
#align(center)[
  #tablem[
    | Soldier | Hero       | <        | Soldier |
    | Guard   | Horizontal | <        | Guard   |
    | ^       | Soldier    | Soldier  | ^       |
    | Soldier | Gate       | <        | Soldier |
  ]
]
```