

内容

一段文档内容。

这种类型是 Typst 的核心。您编写的所有标记和您调用的大多数[函数](#)都会生成内容值。您可以通过将标记括在方括号中来创建内容值。这也是将内容传递给函数的方式。

例子

```
Type of *Hello!* is  
#type([*Hello!*])
```

```
Type of Hello! is content
```

内容可以用+运算符相加、[连接在一起](#)并与整数相乘。无论何时需要内容，您都可以传递[字符串](#)或[none](#)。

表示

内容由带有字段的元素组成。当使用其[元素函数](#)构造元素时，您提供这些字段作为参数，并且当您有内容值时，您可以使用字段访问[语法访问](#)其字段。

某些字段是必需的：在构造元素时必须提供这些字段，因此，它们始终可以通过对该类型的内容进行字段访问来使用。必填字段在文档中进行了标记。

大多数字段都是可选的：与必填字段一样，它们可以传递给元素函数来为单个元素配置它们。但是，也可以使用[设置规则](#)来配置它们，以将它们应用于范围内的所有元素。可选字段仅在显式传递给元素函数时才可用于字段访问语法，而不是在它们来自设定规则时可用。

每个元素都有默认外观。但是，您也可以使用[显示规则](#)完全自定义其外观。显示规则通过元素传递。它可以访问元素的字段并从中生成任意内容。

在 Web 应用程序中，您可以将鼠标悬停在内容变量上，以准确查看内容由哪些元素组成以及它们具有哪些字段。或者，您可以检查函数的输出[repr](#)。

定义

func

内容的元素函数。此函数可用于创建此内容中包含的元素。它可用于设置和显示元素的规则。可以与全局函数进行比较，以检查是否具有特定类型的元素。

自己。[函数](#)（）-> [功能](#)

has

内容是否有指定字段。

```
自己。有 (斯特) -> 布尔值
```

field 斯特 必需的 位置性 ⓘ

要寻找的字段。

at

访问内容上的指定字段。如果该字段不存在，则返回默认值；如果未指定默认值，则失败并出现错误。

```
自己。在 (
  斯特,
  默认: 任何,
) -> 任何
```

field 斯特 必需的 位置性 ⓘ

要访问的字段。

default 任何

如果字段不存在则返回默认值。

fields

返回此内容的字段。

```
自己。字段 () -> 字典
```

```
#rect(
  width: 10cm,
  height: 10cm,
```

```
(width: 283.46pt, height: 283.46pt)
```

```
).fields()
```

location

内容的位置。这仅适用于由[查询](#)返回的内容或由[显示规则](#)提供的内容，对于其他内容也适用`none`。生成的位置可与[计数器](#)、[状态](#)和[查询](#)一起使用。

自己。地点（）-> 没有任何 地点

< 计算
上一页

约会时间 >
下一页