

类型

描述了一种价值。

要设置文档样式，您需要使用不同类型的值：指定元素大小的长度、文本和形状的颜色等等。Typst 将它们分类为明确定义的类型，并告诉您它在哪里期望哪种类型的值。

除了数值的基本类型和[编程语言中已知的典型类型](#)之外，Typst 还提供了一种特殊的内容类型。这种类型的值可以包含您可以输入到文档中的任何内容：文本、标题和形状等元素以及样式信息。

例子

```
#let x = 10
#if type(x) == int [
  #x is an integer!
] else [
  #x is another value...
]

An image is of type
#type(image("glacier.jpg")).
```

```
10 is an integer!
An image is of type content.
```

的类型10是int.现在，int或 的类型是什么type?

```
#type(int) \
#type(type)
```

```
type
type
```

兼容性

在 Typst 0.7 及更低版本中，该type函数返回字符串而不是类型。与旧方式的兼容性将保留一段时间，以便为软件包作者提供升级时间，但它会在某个时候被删除。

- 检查如评估`int == "integer"` `true`
- 添加/连接类型和字符串将产生一个字符串
- `in`类型和字典上的运算符将计算字典 `true` 是否具有与类型名称匹配的字符串键

构造函数

确定值的类型。

类型 (任何) -> 类型

```
#type(12) \  
#type(14.7) \  
#type("hello") \  
#type(<glacier>) \  
#type([Hi]) \  
#type(x => x + 1) \  
#type(type)
```

integer
float
string
label
content
function
type

value 任何 必需的 位置性 ⓘ

其类型待确定的值。

< 系统
上一页

版本 >
下一页