

斯特

Unicode 代码点序列。

您可以使用 [for 循环](#) 迭代字符串的字素簇。字素簇基本上是字符，但将属于在一起的事物放在一起，例如一起形成标志表情符号的多个代码点。字符串可以用运算符相加+、[连接在一起](#)并与整数相乘。

Typst 提供了用于字符串操作的实用方法。其中许多方法（例如 `split`、`trim` 和 `replace`）对模式进行操作：模式可以是字符串或[正则表达式](#)。这使得这些方法非常通用。

所有长度和索引均以 UTF-8 字节表示。索引从零开始，负索引环绕到字符串末尾。

您可以使用该类型的构造函数将值转换为字符串。

例子

```
#"hello world!" \
#"\"hello\n world\"!" \
#"1 2 3".split() \
#"1,2;3".split(regex("[,;]")) \
#(regex("\\d+") in "ten euros") \
#(regex("\\d+") in "10 euros")
```

```
hello world!
"hello
 world!"
("1", "2", "3")
("1", "2", "3")
false
true
```

转义序列

就像在标记中一样，您可以转义字符串中的一些符号：

- `\\`反斜杠
- `\`获取报价
- `\n`对于换行符
- `\r`用于回车
- `\t`对于一个选项卡
- `\u{1f600}`对于十六进制 Unicode 转义序列

构造函数

将值转换为字符串。

- 整数以 10 为基数进行格式化。这可以使用可选 `base` 参数覆盖。

- 浮点数以 10 为基数格式化，从不采用指数表示法。
- 从标签中提取名称。
- 字节被解码为 UTF-8。

如果您希望在 Unicode 代码点之间进行转换，请参阅[to-unicode](#)和[from-unicode](#)函数。

```
str (
  整数  漂浮  斯特  字节  标签  版本  类型 ,
  根据: 整数 ,
) -> 斯特
```

```
#str(10) \
#str(4000, base: 16) \
#str(2.7) \
#str(1e8) \
#str(<intro>)
```

```
10
fa0
2.7
100000000
intro
```

value 整数 或者 漂浮 或者 斯特 或者 字节 或者 标签 或者 版本 或者 类型 必需的 位置性 ?

应转换为字符串的值。

base 整数

显示 2 到 36 之间整数的底数（基数）。

默认: 10

定义 ?

len

以 UTF-8 编码字节表示的字符串长度。

自己。伦 () -> 整数

first

提取字符串的第一个字素簇。如果字符串为空，则会失败并出现错误。

自己。第一的 () -> 斯特

last

提取字符串的最后一个字素簇。如果字符串为空，则会失败并出现错误。

自己。最后的 () -> 斯特

at

提取指定索引后的第一个字素簇。如果索引超出范围，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 (
 整数 ,
 默认: 任何 ,
) -> 任何

index 整数 必需的 位置性 ?

字节索引。如果为负数，则从后面开始索引。

default 任何

索引越界时返回的默认值。

slice

提取字符串的子字符串。如果开始或结束索引超出范围，则会失败并出现错误。

自己。切片 (
 整数 ,
 没有任何 整数 ,
 数数: 整数 ,
) -> 斯特