

漂浮

浮点数。

实数的有限精度表示。Typst 使用 64 位来存储浮点数。无论何时需要浮点数，您也可以传递[整数](#)。您可以使用该类型的构造函数将值转换为浮点数。

例子

```
#3.14 \
#1e4 \
#(10 / 4)
```

```
3.14
10000
2.5
```

构造函数


将值转换为浮点数。

- 布尔值转换为 0.0 or 1.0。
- 整数将转换为最接近的 64 位浮点数。
- 比率除以 100%。
- 字符串以 10 为基数解析为最接近的 64 位浮点数。支持指数表示法。

漂浮 ([布尔值](#) [或者](#) [整数](#) [漂浮](#) [比率](#) [斯特](#)) -> [漂浮](#)

```
#float(false) \
#float(true) \
#float(4) \
#float(40%) \
#float("2.7") \
#float("1e5")
```

```
0
1
4
0.4
2.7
100000
```

value [布尔值](#) [或者](#) [整数](#) [或者](#) [漂浮](#) [或者](#) [比率](#) [或者](#) [斯特](#) [必需的](#) [位置性](#) 

应转换为浮点数的值。

定义

is-nan

检查浮点数是否不是数字。

在 IEEE 754 中，多个位模式表示 NaN。true 如果浮点数是这些位模式中的任何一个，则该函数返回。

自己。是南() -> 布尔值

```
#float.is-nan(0) \
#float.is-nan(1) \
#float.is-nan(calc.nan)
```

```
false
false
true
```

is-infinite

检查浮点数是否无限。

对于浮点数来说，有正无穷大和负无穷大。true 如果浮点数为正无穷大或负无穷大，则此函数返回。

自己。是无穷大() -> 布尔值

```
#float.is-infinite(0) \
#float.is-infinite(1) \
#float.is-infinite(calc.inf)
```

```
false
false
true
```

signum

计算浮点数的符号。

- 如果数字为正数（包括），则返回。+0.01.0
- 如果数字为负数（包括），则返回。-0.0-1.0
- 如果数字为，则返回。calc.nan

自己。符号() -> 漂浮

```
#(5.0).signum() \
#(-5.0).signum() \
#(0.0).signum() \
```

```
1
-1
1
```

< 评价
上一頁

功能 >
下一頁