

字典

从字符串键到值的映射。

key: value您可以通过将逗号分隔的对括在括号中来构造字典。这些值不必是同一类型。由于空括号已经产生一个空数组，因此您必须使用特殊(:)语法来创建一个空字典。

字典在概念上类似于数组，但它是通过字符串而不是整数进行索引的。您可以使用该方法访问和创建字典条目.at()。如果您静态地知道键，则可以使用[字段访问符号](#)(.key)来访问值。字典可以用运算符添加+并[连接在一起](#)。要检查字典中是否存在某个键，请使用关键字in。

[您可以使用for 循环](#)迭代字典中的对。这将按照插入/声明对的顺序进行迭代。

例子

```
#let dict = (  
  name: "Typst",  
  born: 2019,  
)  
  
#dict.name \  
#(dict.launch = 20)  
#dict.len() \  
#dict.keys() \  
#dict.values() \  
#dict.at("born") \  
#dict.insert("city", "Berlin ")  
#("name" in dict)
```

```
Typst  
3  
("name", "born", "launch")  
("Typst", 2019, 20)  
2019  
true
```

构造函数

将值转换为字典。

请注意，此函数仅用于将类似字典的值转换为字典，而不是用于从各个对创建字典。请改用字典语法(key: value)。

字典 (模块) -> 字典

```
#dictionary(sys).at("version")
```

0.11.0

value 模块 必需的 位置性 ?

应转换为字典的值。

定义 ?

len

字典中的对数。

```
自己。伦（）-> 整数
```

at

返回与字典中指定键关联的值。如果字典中已存在该键，则可以在作业的左侧使用。如果键不是字典的一部分，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
```

key 斯特 必需的 位置性 ?

检索项目的键。

default 任何

如果键不是字典的一部分，则返回默认值。

insert

将新对插入字典中。如果字典已包含此键，则更新该值。

```
自己。插入（  
    斯特，  
    任何，  
）
```

key 斯特 必需的 位置性 ?

应插入的密钥对的密钥。

value 任何 必需的 位置性 ?

应插入的对的值。

remove

通过键从字典中删除一对并返回值。

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

key 斯特 必需的 位置性 ?

要删除的对的密钥。

default 任何

如果键不存在则返回默认值。

keys

按插入顺序将字典的键作为数组返回。

```
自己。键 () -> 大批
```

values

按插入顺序以数组形式返回字典的值。

```
自己。值 () -> 大批
```

约会时间
上一页

期间
下一页