

arguments

捕获函数的参数。

论点沉没

与内置函数一样，自定义函数也可以采用可变数量的参数。您可以指定一个参数接收器，它将所有多余的参数收集为`..sink`。结果 `sink` 值是类型 `arguments`。它公开了访问位置参数和命名参数的方法。

```
#let format(title, ..authors) = {  
  let by = authors  
    .pos()  
    .join(", ", last: " and ")  
  
  [*#title* \ _Written by #by;_]  
}  
  
#format("ArtosFlow", "Jane", "Joe")
```

```
ArtosFlow  
Written by Jane and Joe
```

传播

与参数接收器相反，您可以使用运算符将参数、数组和字典传播到函数调用中`..spread`:

```
#let array = (2, 3, 5)  
#calc.min(..array)  
#let dict = (fill: blue)  
#text(..dict)[Hello]
```

```
2 Hello
```

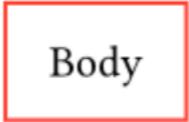
构造函数

构建可传播的论据。

该函数的行为类似于 `.let args(..sink) = sink`

参数 (.. 任何) -> 论据

```
#let args = arguments(stroke: red, inset: 1em, [Body])
#box(..args)
```



arguments 任何 必需的 位置性 ? 可变参数 ?

定义 ?

pos

以数组形式返回捕获的位置参数。

自己。位置 () -> 大批

named

以字典形式返回捕获的命名参数。

自己。命名为 () -> 字典

基础
上一页

大批
下一页