query

查找文档中的元素。

这些 query 功能使您可以在文档中搜索特定类型或具有特定标签的元素。要使用它，您首先需要确保上下文可用。

寻找元素

在下面的示例中，我们创建一个自定义页面标题，以小写字母显示文本“Typst Academy”和当前部分标题。在第一页上，节标题被省略，因为标题位于第一个节标题之前。

为了实现这种布局，我们打开一个 context，然后查询当前位置之后的所有标题。上下文块中的代码运行两次：每页运行一次。

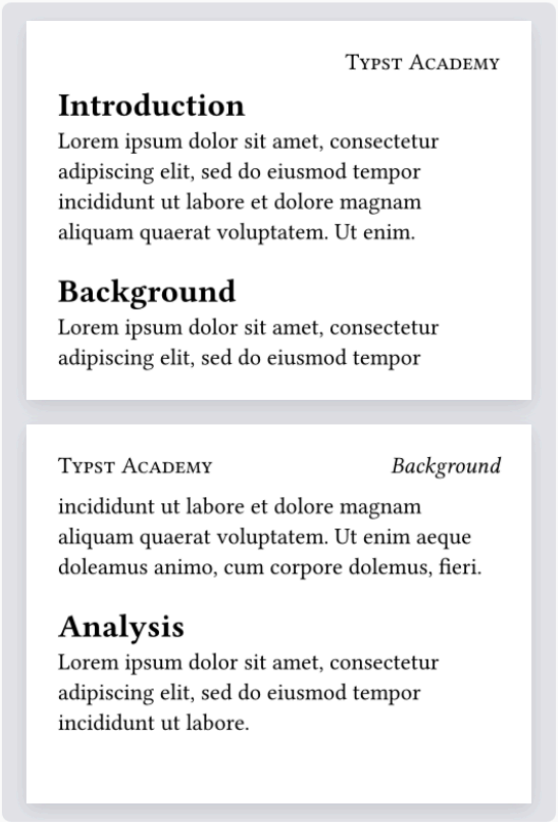
- 在第一页上，查询当前位置之前的所有标题会产生一个空数组：没有以前的标题。我们检查这种情况并只显示“Typst Academy”。
- 对于第二页，我们从查询结果中检索最后一个元素。这是当前位置之前的最新标题，因此，它是我们当前所在部分的标题。我们通过该 body 字段访问其内容并将其显示在“Typst Academy”旁边。

```
#set page(header: context {
  let elems = query(
    selector(heading).before(here()),
  )
  let academy = smallcaps[
    Typst Academy
  ]
  if elems.len() == 0 {
    align(right, academy)
  } else {
    let body = elems.last().body
    academy + h(1fr) + emph(body)
  }
})

= Introduction
#lorem(23)

= Background
#lorem(30)

= Analysis
#lorem(15)
```



query您可以获取with返回的元素的位置location。

一句警告

为了解决您的所有疑问，Typst 会多次评估和布局文档的各个部分。但是，不能保证您的疑问实际上可以得到完全解决。如果您不小心，查询可能会影响自身，从而导致结果永远不稳定。

在下面的示例中，我们查询文档中的所有标题。然后我们生成尽可能多的标题。一开始，只有一个标题，名为Real。因此，生成了count一个1标题。 FakeTypst 发现查询结果已更改并再次处理它。这次，生成了count两个2标题。Fake这种情况一直持续下去。正如我们所看到的，输出的标题数量是有限的。这是因为 Typst 在几次尝试后就放弃了。

一般来说，您应该尽量不要编写影响自身的查询。同样的警告也适用于其他内省功能，例如计数器和状态。

```
= Real
#context {
  let elems = query(heading)
  let count = elems.len()
  count * [= Fake]
}
```

Real

Fake

Fake

Fake

Fake

命令行查询

您还可以使用命令从命令行执行查询`typst query`。此命令对文档执行任意查询并以序列化形式返回结果元素。考虑以下`example.typ`文件，其中包含一些不可见的元数据：

```
#metadata("This is a note") <note>
```

您可以使用 Typst 的 CLI 对其执行查询，如下所示：

```
$ typst 查询 example.typ "<note>"
 [
   {
 "func" : "metadata" ,
 "value" : "这是一条注释" ,
 "label" : "<note>"
   }
 ]
```

通常，您只对结果元素的一个特定字段感兴趣。就元素而言`metadata`，`value`字段是最有趣的。您可以使用参数仅提取该字段`--field`。

```
$ typst 查询 example.typ "<note>" --field value
 [ "This is a note" ]
```

如果您只对单个元素感兴趣，则可以使用该`--one`标志来提取它。

```
$ typest query example.typ "<note>" --field value --one
 "这是一条注释"
```

参数

```
询问（
  标签 选择器 地点 功能，
  没有任何 地点，
）-> 大批
```

target 标签 或者 选择器 或者 地点 或者 功能 *必需的* *位置性*

可

- 一个元素函数，如heading或 figure，
- A <label>，
- 更复杂的选择器，例如heading.where(level: 1)
- 或者。selector(heading).before(here())

仅支持可定位元素函数。

location 没有任何 或者 地点 *位置性*

*兼容性：*此参数仅用于与 Typst 0.10 及更低版本兼容，不应再使用。

默认：none