TYPST SCRIPTS

Interpret Mode

Typst 在处理文档时，在一下的三种模式之间切换, 其实对于 TeX 来说，在 TeX 的内部，也有类似的模式切换:

- 标记模式, 默认就在标记模式
- 脚本模式, 用 `#` 开始: Typst 总是倾向于更快地退出脚本模式
- 数学模式, 用 `$` 开始

> **Note:** 在 Typst 中， `[]` (内容块) 可以看成是 LaTeX 中的 `{}` . 一个关于函数参数的语法糖, 也就是说这三个形式等价:
>
> - #emph()[内容]
> - #emph[内容]
> - #emph([内容])
> - #emph(“内容”)

> **Note:** Typst 遇到 `[]` ，自动化离开脚本模式，进入标记模式，所以在 `[]` 中要进入脚本模式，需要用 `#`

Code/Content Block

内容块（标记模式）内部没有语句的概念. `代码块` 内部有语句概念, 每个语句可以是 `换行` 分隔，也可以是「分号」`;` 分隔.

Foldable

字符串的折叠实际上折叠操作基本就是「加号」（+）操作. 但是布尔值、整数和浮点数都不能相互折叠，下面的语句无法运行:

```Typst
// 不能编译
#{ false; true }; #{ 1; 2 }; #{ 1.; 2. }
```

> **Note:** 内容也可以作为代码块的语句结果，这时候内容块的结果是每个语句内容的“折叠”。

3

List and Dict

数组和字典的创建方式:

- 数组: `(1, 2, 3)`, 不必要求数组元素的同质化

> **Note:** 构造数组字面量时，允许尾随一个多余的逗号而不造成影响

- 字典: `{a: 1, b: 2, c: 3}`, 字典的键可以是任意类型. 一个示例如下：

```Typst
#let cat = (
neko-mimi: 2,
"utterance": "喵喵喵",
attribute: [kawaii\~]
)
#cat
```

访问数组和字典，你都可以使用 `.at` 方法. 有一个所谓的 `典型赋值方法`，代码示例如下:

```Typst
#let cat = (
  neko-mimi: 2,
  "utterance": "喵喵喵",
  attribute: [kawaii\~]
)
#let (utterance: u, attribute: a) = cat
#u \
#a
```

while/for/break/continue

和其他语言中的用法类似, 这里不多介绍.

Function

Introduction

Typst 对内置实现的所有函数都有良好的自我管理，但总免不了用户打算写一些逆天的 函数。为了保证缓存计算仍较为有效，Typst 强制要求用户编写的所有函数都是`纯函数`.

> **Definition:** 一个函数是纯的，如果：
>
> 1. 对于所有相同参数，返回相同的结果。
> 2. 函数没有副作用，即局部静态变量、非局部变量、可变引用参数或输入/输出流等状态不会发生变化

> **Note:** 因为纯函数不允许产生带有副作用的操作, 所以函数的函数体表达式不允许涉及到函数体外的变量修改. 所以,传递进函数的数组和字典参数都会被拷贝。这将导致对参数数组或参数字典的修改不会影响外部变量的内容

为了“修改”外部变量，你必须将修改过的变量设法传出函数，并在外部更新外部变量。下面就是一个示例:

```typst
#let a = (1, );
#let add-array(a) = {
  a += (2, )
  return a
};
#a \ // 初始值
#(a = add-array(a));
#a \ // 返回值更新数组
```

Closure

首先引入一个函数闭包的样例, 以此来初步认识闭包的概念:

```typst
#let f = (x, y) => [两个值 #(x)和 #(y)偷偷
混入了我们内容之中。]
#f("a", "b")
```

两个值 a 和 b 偷偷 混入了我们内容之中。

可以看出，具有基本的三个部分:

- 函数名: `f`
- 参数列表: `(x, y)`
- 函数体: `[两个值 #(x)和 #(y)偷偷混入了我们内容之中。]`

传统的`函数声明`式写法就像下面这样:

```typst
#let f(loc) = [
  当前页面为: #loc.page()
]
#locate(f)
```

当前页面为: 3

然而，写为 `闭包` 的形式，就像下面这样:

```typst
#locate( loc => [
  当前页面为: #loc.page()
])
```

当前页面为: 3