链滴

Go 语言基础语言之变量定义

作者：superstonne

原文链接：https://ld246.com/article/1568534452758

来源网站：链滴

Go 语言基础定义变量

上一节我们学习了 Go 语言开发环境的安装，这一节我们一起来学习一下 Go 语言的基础语法

VSCode Go 开发环境搭建

打开 VsCode，使用 Ctrl+Shift+X or Cmd+Shift+X 来打开扩展安装视图，安装 Go 插件，安装完后即可开始你的 Go 编程咯。插件定制化配置详情参考：

Go VS Code 插件

定义变量

1. 定义单个变量和多个变量

如下我们使用 var varibleName varibleType 来定义一个变量，定义多个变量使用逗号进行分割。如我们定义了 2 个整型变量 num1 和 num2，并且对它们进行了赋值操作，最后将它们打印在控制台。

```
var num1, num2 int
num1 = 1
num2 = 2
fmt.Printf("num1 = %d, num2 = %d", num1, num2)
```

2. 定义变量并且初始化

如下我们定义了 2 个字符串变量，并且直接初始化了它们。

```
var str1, str2 string = "string 01", "string 02"
```

3. 定义变量省去类型

上面的定义方法是不是有些繁琐，Go 支持自动识别变量类型。下面定义了一个整型和一个字符串，是我们没有指定类型，Go 编译器自动帮助我们识别了类型。

```
var num3, str3 = 100, "string 03"
```

4. 简短声明

如果你觉得上面的定义还是繁琐，Go 提供了更简单的方法。如下简短声明可以达到同样的效果，但你要注意，这种方式只适用于在函数内部声明局部变量，在函数外使用会发生编译错误 syntax error: on-declaration statement outside function body。

```
num4, str4 := 200, "string 04"
```

5. 声明但是未使用

go 语言不允许存在定义了但是未曾使用的变量，在编译阶段会直接报错 unUsed declared and not u ed

经过上述的介绍相信你已经掌握了如何使用多种方式来定义 Go 语言的变量，并且你应该注意到了所的代码都没有分号终止符号。是的，Go 语言不需要这些多余的终止分号。

定义常量

Go 语言支持常量类型的定义，使用 const constantVarible = constantValue 即可完成一个常量的义。同样定义常量我们既可以明确指定类型，也可以省略类型。如下展示我们定义的常量：

```
const PI = 3.141592653
const price float32 = 0.01
const name = "Nick"
const age = 21
fmt.Printf("PI = %f, price = %f, name = %s, age = %d", PI, price, name, age)

// OutPut
// PI = 3.141593, price = 0.010000, name = Nick, age = 21
```

内置基础类型

1. 布尔类型（默认值 false）

2. 数字类型

- 整数类型（见名知意）

int（32 位），uint（32 位），rune（int32 的别称），int8，int16，int32，int64，byte（uint8 别称），uint8，uint16，uint32，uint64

- 浮点类型

float32，float64

- 复数类型：complex64，complex128

3. 字符串（通通采用 utf-8 编码，彻底解决乱码问题）

4. 错误类型（error）

定义数组

1. array

采用 var arrName [length] arrType 即可完成一个数组的定义，arrName 是数组变量的名字，lengt

是数组的长度，arrType 是数组的类型，如下我们定义了一个长度为 2 的 整型数组，数组中每个元的默认值是 0：

```
var arr1 [2] int
```

当然我们可以通过数组下标来对它进行赋值，或者定义和初始化放在一起。如下：

```
arr1[0] = 0
arr1[1] = 1

var arr2 = [2]int{1, 2}
```

当然也可以使用简化模式，同时省略了数组的长度，编译器自动识别长度：

```
arr3 := []int {1, 2, 3}
```

数组的长度定义后便不会更改，假设我像增加一个数组的长度，那我们我可以通过 append 函数来添，操作后将返回一个新的数组，但是不会对旧的数组产生影响。如下：

```
var slice2 = append(arr3, 1)
```

如上所示，此时 arr3 还是有三个元素，但是 slice2 是有四个元素的数组。

定义 map

map 有编程经验的小伙伴一定理解，它是一个 key 和 value 键值对结构。我们可以使用采用 2 种方来定义 map：

- 定义
  - 先定义，然后初始化内存。如下我们定义了一个 key 为 string 类型，值为整型的 map。

```
var myMap map[string]int
myMap = make(map[string]int)
```

  - 定义和初始化内存一起，如下是第一种方式的简化版本。

```
var myMap = make(map[string]int)
```

注意我们必须使用 make 来分配内存，否则会收到编译错误 panic: assignment to entry in nil map

- 赋值，获取 map 中的值

```
myMap["Jack"] = 12
myMap["nick"] = 11

value, contain := myMap["Jack"]
if contain {
  fmt.Printf("myMap contains Jack, age = %d\n", value)
} else {
  fmt.Printf("myMap doesn't contain Jack\n")
}

value1, contain1 := myMap["Jenny"]
if contain1 {
  fmt.Printf("myMap contains Jenny, age = %d\n", value1)
} else {
  fmt.Printf("myMap doesn't contain Jenny\n")
```

```
}
```

赋值操作操作比较简单，如上我们添加了 2 个 key： Jack , nick

和其他编程语言区别较大的地方是获取 map 元素的时候返回值有 2 个，第一个代表 value，第二个一个 bool 值，代表你想要获取的 key 是否存在于 map 中，如上展示体验一下。

约定由于配置

1. 大写字母开头的变量为 public 变量，外部可用。小写开头反之为私有 private 变量
2. 上述规则同样适用于函数的声明

总结

今天我们学习了在 Go 语言中如何定义个一个变量，以及定义变量的几种方式。同时学习了如何定义量，数组，map 类型。最后介绍 Go 语言编程的基本约定，相信你已经掌握了 Go 语言的变量知识，去体验一下吧！下一节，我们将学习 Go 语言的流程控制。