链滴

GO 语言学习 -1 变量

作者：rolayyalor

原文链接：https://ld246.com/article/1593424041060

来源网站：链滴

变量

Go 是静态类型语言，不能在运行期改变变量类型。

使用关键字 var 定义变量，自动初始化为零值。如果提供初始化值，可省略变量类型，由编译器自动断。

```
var x int
var f float32 = 1.6
var s = "abc"
```

在函数内部，可用更简略的 ":=" 方式定义变量。

```
func main() {
x := 123 // 注意检查，是定义新局部变量，还是修改全局变量。该方式容易造成错误。
}
```

可一次定义多个变量。

```
var x, y, z int
var s, n = "abc", 123
var (
a int
b float32
)
func main() {
n, s := 0x1234, "Hello, World!"
println(x, s, n)
}
```

多变量赋值时，先计算所有相关值，然后再从左到右依次赋值。

```
data, i := [3]int{0, 1, 2}, 0
i, data[i] = 2, 100 // (i = 0) -> (i = 2), (data[0] = 100)
```

特殊只写变量 "_"，用于忽略值占位。

```
func test() (int, string) {
return 1, "abc"
}
func main() {
_, s := test()
println(s)
}
```

编译器会将未使用的局部变量当做错误。

```
var s string // 全局变量没问题。
func main() {
i := 0 // Error: i declared and not used。(可使用 "_ = i" 规避)
}
```

注意重新赋值与定义新同名变量的区别。

```
s := "abc"
println(&s)
s, y := "hello", 20 // 重新赋值: 与前 s 在同一层次的代码块中，且有新的变量被定义。
println(&s, y) // 通常函数多返回值 err 会被重复使用。
{
s, z := 1000, 30 // 定义新同名变量: 不在同一层次代码块。
println(&s, z)
}
```

输出：

0x2210230f30

0x2210230f30 20

0x2210230f18 30