变量

go语言使用var对变量进行声明。也可以使用:=来对变量直接进行初始化。 例如：

```
1：变量的定义
var a int
var b string
v3 :=5
var d[5] int
var e[] int
2：相互交换
i := 2
j := 3
i, j = j, i  //交换i和j的值，此时i == 3，j == 2
3：Go语言的数据类型
a.整形
类型    说明
byte    等同于 uint8
int    依赖于不同平台下的实现，可以是 int32 或者 int64
int8    [-128, 127] （相当c语言的char）
int16   [-32768, 32767]（c语言的short）
int32  [-2147483648, 2147483647]
int64  [-9223372036854775808, 9223372036854775807]
rune 等同于 uint32
uint 依赖于不同平台下的实现，可以是 uint32 或者 uint64
uint8   [0, 255]
uint16  [0, 65535]
uint32  [0, 4294967295]
uint64  [0, 18446744073709551615]
uintptr  一个可以恰好容纳指针值的无符号整型（对 32 位平台是 uint32，对 64 位平台是
uint64）
在c语言下我们使用的是sizeof操作符来查看类型的字节长度，在Go语言中可以通过unsafe.Sizeof函
数进行。
b.浮点数
类型  说明
float32   ±3.402 823 466 385 288 598 117 041 834 845 169 254 40x1038 计算精度大概是
小数点后 7 个十进制数
float64   ±1.797 693 134 862 315 708 145 274 237 317 043 567 981x1038 计算精度大概
是小数点后 15 个十进制数
complex32  复数，实部和虚部都是 float32
complex64  复数，实部和虚部都是 float64
c.布尔类型
var a bool
a = true
b := (2 == 3) //b也会被推导为bool类型
//错误示范
var b bool
b = 1 //编译错误
b = bool(1) //编译错误
d.字符串
例如：
t1 := "\"hello\""
```

```
t2 :='"hello"'
字符串支持的操作：
语法  描述
s += t  将字符串 t 追加到 s 末尾
s + t   将字符串 s 和 t 级联
s[n]    从字符串 s 中索引位置为 n 处的原始字节
s[n:m]  从位置 n 到位置 m-1 处取得的字符（字节）串
s[n:]   从位置 n 到位置 len(s)-1 处取得的字符（字节）串
s[:m]   从位置 0 到位置 m-1 处取得的字符（字节）串
len(s)  字符串 s 中的字节数
len([]rune(s))    字符串 s 中字符的个数，可以使用更快的方法 utf8.RuneCountInString()
[ ]rune(s)        将字符串 s 转换为一个 unicode 值组成的串
string(chars)     chars 类型是[]rune 或者[]int32，将之转换为字符串
[ ]byte(s)        无副本的将字符串 s 转换为一个原始的字节的切片数组，不保证转换的字节是
合法的 UTF-8 编码字节
e：数组
使用的语法创建如下：
[length]Type
[N]Type{value1,value2,...valueN}
[...]Type{value1,value2,...,valueN}
```