

指针

1. 指针不能赋值
2. go语言只有值传递

go语言的取地址符是&，放到一个变量前使用就会返回相应变量的内存地址

```
1 package main
2
3 import "fmt"
4
5 func main(){
6     var a int = 10
7     fmt.Println("变量的地址: %x\n", &a)
8 }
```

执行以上代码输出结果为：

```
1 变量的地址: %x
2 0xc00007a008
```

什么是指针

一个指针变量指向了一个值的内存地址

在指针类型前面加上*号来获取所指向的内容

```
1 package main
2
3 import "fmt"
4
5 func main(){
6     var a := 20      //声明实际变量
7     var ip *int      //声明指针变量
8
9     ip = &a //指针变量的存储地址
10
11     fmt.Printf("a 的变量地址是: %x\n", &a)
12
13     fmt.Printf("ip 变量的存储的指针地址: %x\n", ip)
14
15     fmt.Printf("*ip 变量的只: %d\n", *ip)
```

```
16 | }
```

以上实例执行输出结果为：

```
1 | a 的变量地址是：c00006e010
2 | ip 变量的存储的指针地址： c00006e010
3 | ip 变量的只：20
```

Go空指针

nil指针也被称为空指针

nil在概念上和其它语言的**null**、**none**、**nil**、**NULL**一样，都指代零值或空值