数据类型、变量、常量

作者：少林之巅

目录

标识符、关键字

1. 标识符是用来表示Go中的变量名或者函数名，以字母或_开头。后面跟着字母、_或数字

   A. 88ab　　B. _ab28　　C. ab_28

2. 关键字

   关键字是Go语言预先定义好的，有特殊含义的标识符。

break	default	func	interface	select
case	defer	go	map	struct
chan	else	goto	package	switch
const	fallthough	if	range	type
continue	for	import	return	var

变量

1. 语法：var identifier type

举例1：

```
var a int
var b string
var c bool
var d int = 8
var e string = “hello”
```

变量

举例2：

```
var (
    a int                   //0
    b string                //“”
    c bool                  //false
    d int = 8               // 8
    e string = “hello”      //hello
)
```

常量

1. 常量使用const 修饰，代表永远是只读的，不能修改。

2. 语法：const identifier [type] = value，其中type可以省略。

举例：

const b string = “hello world”

const b = “hello world”

const Pi = 3.1414926

const a = 9/3

常量

4. 比较优雅的写法：

```
const(
    a = 1
    b = 2
    c = 3
)
```

5. 更加专业的写法：

```
const (
    a = iota
    b
    c
)
```

```
const(
    a = 1 << iota
    b
    c
)
```

数据类型

1. 布尔类型

2. 整数和浮点数类型

3. 字符串类型

数据类型和操作符

1. 布尔类型

    a. var b bool　和 var b bool = true 和 var b = false

    b. 操作符 == 和 !=

    c. 取反操作符：　!b

    d. && 和 || 操作符

    e. 格式化输出占位符: %t

数据类型和操作符

2. 整数和浮点数类型

a. int8、int16、int32、int64

b. uint8、uint16、uint32、uint64

c. int 和 uint，和操作系统平台相关

d. float32 和 float64浮点类型

e. 所有整数 初始化为0，所有浮点数初始化为0.0，布尔类型初始化为false

数据类型和操作符

3. 整数和浮点数类型

a. Go是强类型语言，不同类型相加以及赋值是不允许的

b. 那怎么样才能实现，不同类型相加呢?

c. 输出占位符：整数%d，%x十六进制，%f浮点数

数据类型和操作符

4. 字符串类型

a. var str string

b. var str string = “hello world”

c. 字符串输出占位符%s

d. 万能输出占位符: %v

字符串

5. 字符串的两种表示方式

a. 双引号，“”，可以包含控制字符
b. 反引号，``，所有字符都是原样输出

```
package main
import “fmt”
func main() {
    var str = “hello world\n\n”
    var str2 = `hello \n \n \n`

    fmt.Println(“str=“, str)
    fmt.Println(“str2=“, str2)
}
```

字符串

6. 字符串常用操作

a. 长度：len(str)
b. 拼接：+，fmt.Sprintf
c. 分割：strings.Split
d. 包含： strings.Contains
e. 前缀或后缀判断：strings.HasPrefix, strings.HasSuffix
f. 子串出现的位置：strings.Index(), strings.LastIndex()
g. join操作： strings.Join(a[]string, sep string)

数据类型和操作符

5. 操作符

a. 逻辑操作符， == 、 != 、<、<=、>=

b. 算数操作符， +、-、*、/、%

go程序的基本结构

```
package main

import “fmt”

func main() {

    fmt.Println(“hello, world”)

}
```

1. 任何一个代码文件隶属于一个包

2. import 关键字，引用其他包：

```
import(“fmt”)
import(“os”)
```

通常习惯写成：

```
import (
      “fmt”
       “os”
)
```

go程序的基本结构

```
package main

import “fmt”

func main() {

    fmt.Println(“hello, world”)

}
```

3. 开发可执行程序，package **main**，
   并且有且只有一个main入口函数

4. 包中函数调用：

   a. 同一个包中函数，直接用函数名调用

   b. 不同包中函数，通过包名+点+
   函数名进行调用

5. 包访问控制规则：

   a. 大写意味着这个函数/变量是可导出
   b. 小写意味着这个函数/变量是私有的
      包外部不能访问