

Day1 Go环境搭建&Go基础

tony

目录

1.环境搭建、标识符、关键字

2. 变量和常量

3. 数据类型

4. Go程序基本结构

5. Golang特性

开课介绍

上课时间	每周六9点半到6点半
培训时长	18周
课程内容	golang培训+项目实战

课堂纪律

上课时间	不要迟到，不要玩手机
上课氛围	积极听讲，有问题就问
课后练习	按时做完并提交
个人态度	主动主动再主动

开发环境搭建

1. 安装Go

- a. 打开网址<https://golang.org/dl/>
- b. 根据操作系统选择对应的安装包
- c. 点击安装包进行安装（linux直接解压）
- d. 设置环境变量
 - 1. `export GOROOT=$PATH:/path/to/go/`
 - 2. `export PATH=$PATH:$GOROOT/bin/`
 - 3. `export GOPATH=/home/user/project/go`

开发环境搭建

2. IDE搭建 (vscode)

- a. 打开网址: <https://code.visualstudio.com/>
- b. 根据操作系统选择对应的安装包
- c. 点击安装包进行安装 (linux直接解压)
- d. 选择查看-》扩展-》搜索go, 安装第二个

开发环境搭建

3. 新建项目

- a. 新建目录/home/user/project/go/src/listen1
- b. 用vscode打开目录/home/user/project/go/src/listen1
- c. 右键新建文件hello.go, 保存
- d. vscode会提示你安装一些go的工具, 我们点击install all

工作区介绍

4. GOPATH

- a. 所有项目都需要放到\$GOPATH/src
- b. GOPATH路径最好只设置一个
- c. 目录结构介绍

Go 基本命令介绍

5. 基本命令

- a. `go run` 快速执行go文件，就像执行脚本一样
- b. `go build` 编译程序，生成二进制可执行文件
- c. `go install` 安装可执行文件到bin目录

Go 基本命令介绍

6. 基本命令

- a. `go test` 执行单元测试或压力测试
- b. `go env` 显示go相关的环境变量
- c. `go fmt` 格式化源代码

Go 基本命令介绍

7. Go程序结构

- a. go源码按package进行组织，并且package要放到非注释的第一行
- b. 一个程序只有一个main包和一个main函数
- c. main函数是程序的执行入口

Go 基本命令介绍

8. 注释

a. 单行注释 //

b. 多行注释 /* */

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	fallthrough	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)
}
```


数据类型和操作符

5. 操作符

a. 逻辑操作符, `==`、`!=`、`<`、`<=`、`>=`

b. 算术操作符, `+`、`-`、`*`、`/`、`%`

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

golang语言特性

1. 垃圾回收

- a. 内存自动回收，再也不需要开发人员管理内存
- b. 开发人员专注业务实现，降低了心智负担
- c. 只需要new分配内存，不需要释放

golang语言特性

2. 天然并发

- a. 从语言层面支持并发，非常简单
- b. goroutine，轻量级线程，创建成千上万个goroutine成为可能
- c. 基于CSP（Communicating Sequential Process）模型实现

```
func calc() {  
    //大量计算  
}  
  
func main() {  
    go calc()  
}
```

golang语言特性

3. channel

- a. 管道, 类似unix/linux中的pipe
- b. 多个goroutine之间通过channel进行通信
- c. 支持任何类型

```
func main() {  
    pipe := make(chan int, 3)  
    pipe <- 1  
    pipe <- 2  
}
```

golang语言特性

4. 多返回值

a. 一个函数返回多个值

```
func calc(a int, b int) (int,int) {  
    sum := a + b  
    avg := (a+b)/2  
    return sum, avg  
}
```

第一个golang程序

1. hello world, 在listen1目录下新建hello.go

```
package main

import (
    "fmt"
)

func main() {
    fmt.Println("hello world")
}
```


包的概念

1. 和python一样，把相同功能的代码放到一个目录，称之为包
2. 包可以被其他包引用
3. main包是用来生成可执行文件，每个程序只有一个main包
4. 包的主要用途是提高代码的可复用性

包的实战

1. 在listen1目录下新建calc目录
2. 在calc目录下新建calc.go

```
package calc

func Add(a int, b int) int {
    return a + b
}
```

包的实战

1. 修改hello.go代码, 如下

```
package main

import (
    "fmt"
    "calc"
)

func main() {
    sum := calc.Add(3, 5)
    fmt.Println("hello world, %d", sum)
}
```

课后作业

1. 使用fmt分别打印字符串、二进制、十进制、十六进制、浮点数。

QA