GOLANG介绍

讲师：kk

课程内容

- 初识GO语言
- 环境安装
- 程序结构
- 基本组成元素
- 基础知识点（一）

初识GO语言

Go是一门开放源码的编程语言，可容易的构建简单、可靠和高效的软件

- 开发者使用编程语言的三大分类（执行速度、简易程度、开发难度）：
  - 执行速度快、编译速度慢(编译型)：C，C++
  - 执行速度较慢、编译速度快(解释型): JAVA，.NET
  - 执行速度慢、开发难度小(动态脚本)：Python，PHP

Go语言在3个条件做了平衡：易于开发、快速编译、高效执行

初识GO语言

特性

- 静态类型并具有丰富的内置类型
- 函数多返回值
- 错误处理机制
- 语言层并发
- 面向对象：使用类型、组合、接口来实现面向对象思想
- 反射
- CGO：用于调用C语言实现的模块
- 自动垃圾回收
- 静态编译
- 交叉编译
- 易于部署
- 基于BSD协议完全开放

初识GO语言

落地应用场景

- Go语言主要用于服务端开发，其定位是开发大型软件，常用于：
  - 服务器编程：日志处理、虚拟机处理、文件系统、分布式系统等
  - 网络编程：Web应用、API应用、下载应用等
  - 内存数据库
  - 云平台
  - 机器学习
  - 区块链
- 使用Go开发的项目：
  - Go
  - Docker
  - kubernetes
  - ……

初识GO语言

在工业界中的应用

- 使用Go开发的组织
- 国外：Google 、 Microsoft 、 Amazon 、 FaceBook
- 国内：阿里、腾讯、百度、京东、爱奇艺、今日头条、滴滴、美团、B站

环境安装

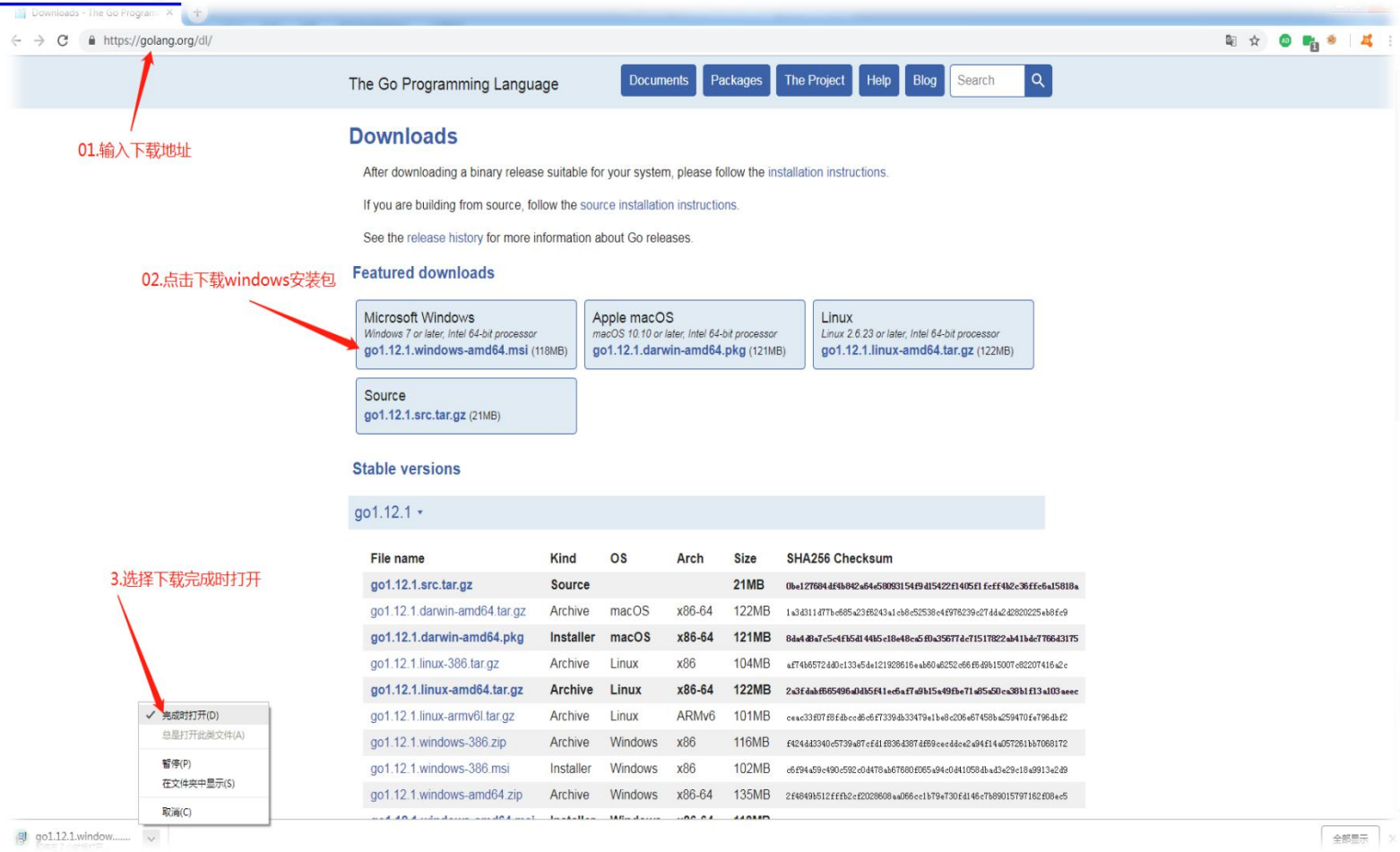
◆ 下载地址：

https://golang.org/dl/

https://golang.google.cn/dl/

Windows：

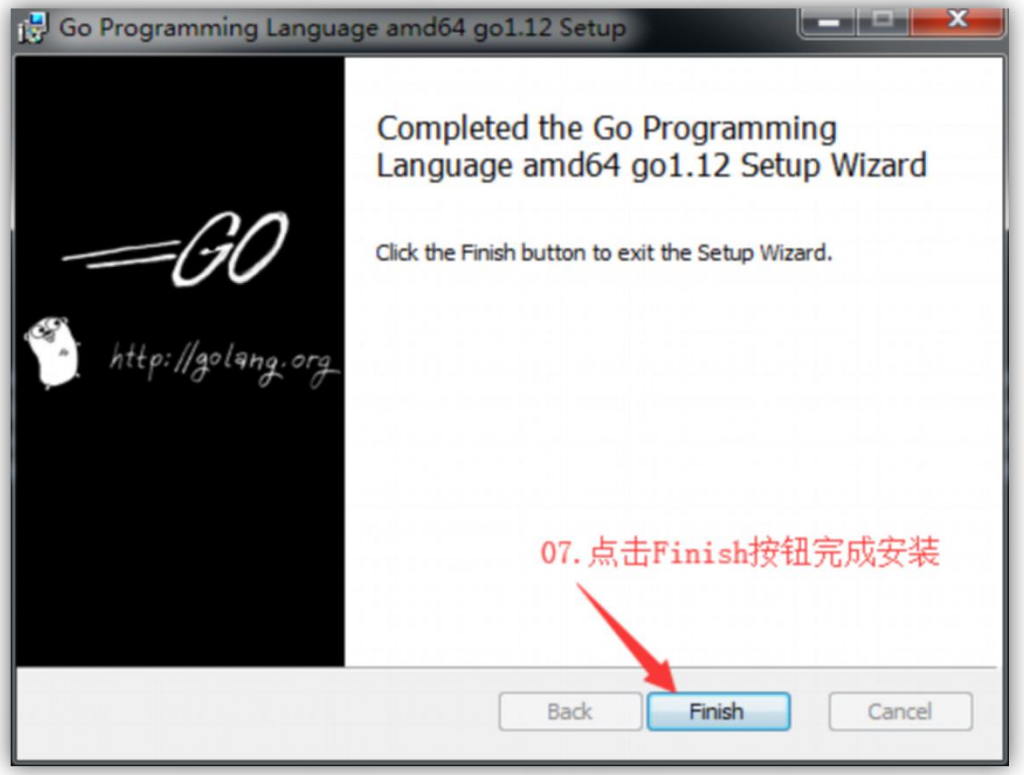
a) 下载安装包



环境安装

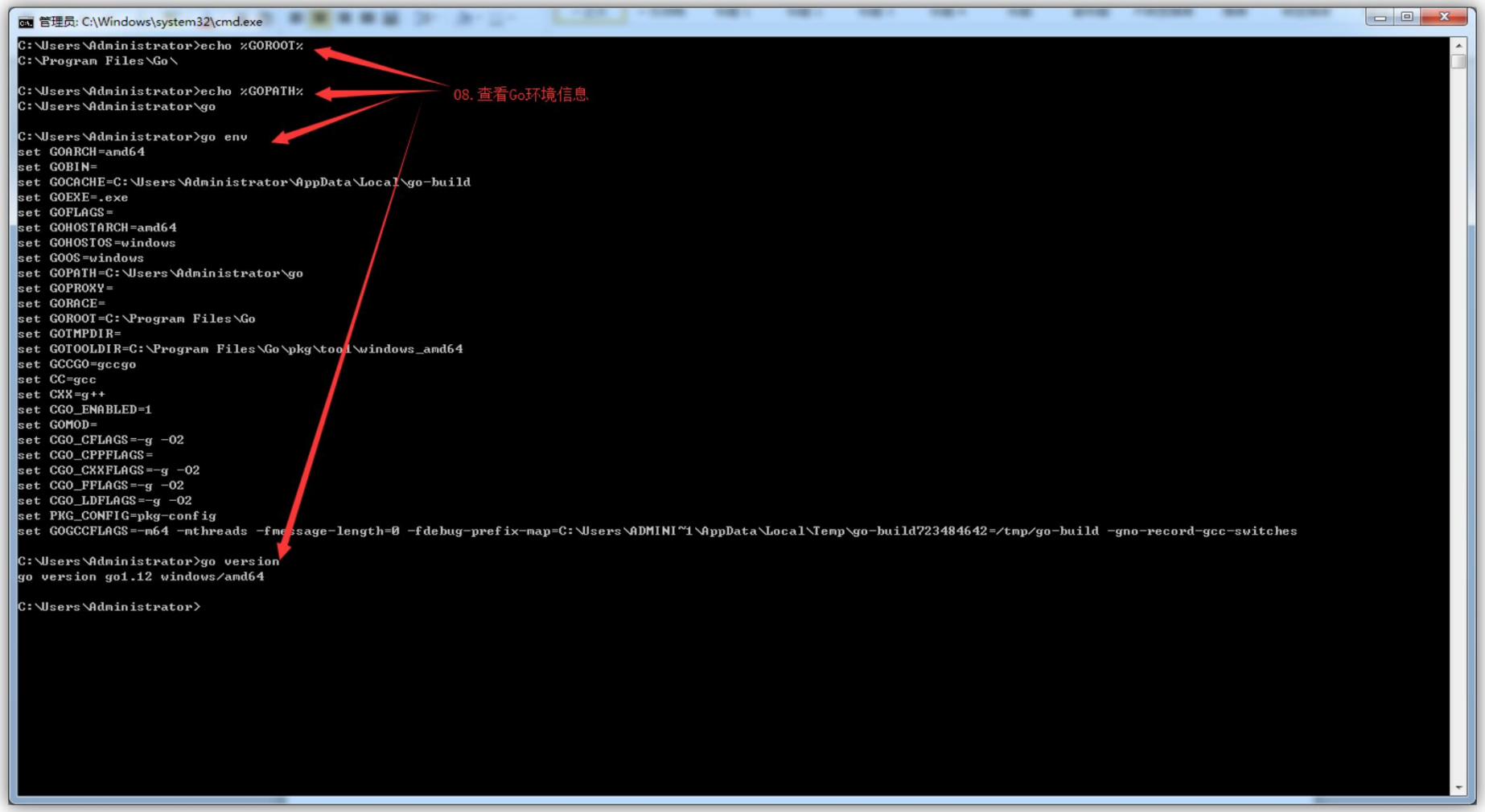
b）安装


Go Programming Language amd64 go1.12 Setup
Welcome to the Go Programming Language amd64 go1.12 Setup Wizard
The Setup Wizard will install Go Programming Language amd64 go1.12 on your computer. Click Next to continue or Cancel to exit the Setup Wizard.
http://golang.org
01.点击Next按钮进行安装
Back
Next
Cancel


Go Programming Language amd64 go1.12 Setup
Completed the Go Programming Language amd64 go1.12 Setup Wizard
Click the Finish button to exit the Setup Wizard.
http://golang.org
07.点击Finish按钮完成安装
Back
Finish
Cancel

环境安装

c）效果查看



程序结构

第一个Go程序

```
package main

import “fmt”

func main() {
        fmt.Println(“Hello World”)
}
```

- Go源文件以package声明开头，说明源文件所属的包。
- 接着使用import导入依赖的包，其次为包级别的变量、常量、类型和函数的声明和赋值。
- 函数中可定义局部的变量、常量。

运行：

```
go build -work -x -o helloworld.exe main.go
go run -work -x main.go
```

基本组成元素

标识符

- 标识符是编程时所使用的名字，用于给变量、常量、函数、类型、接口、包名等进行命名，以建立名称和使用之间的关系。
- Go语言标识符的命名规则：

1. 只能由非空字母(Unicode)、数字、下划线(_)组成
2. 只能以字母或下划线开
3. 不能Go语言关键字
4. 避免使用Go语言预定义标识符
5. 建议使用驼峰式
6. 标识符区分大小写

基本组成元素

标识符

- Go语言提供一些预先定义的标识符用来表示内置的常量、类型、函数。
- 在自定义标识符时应避免使用：

1. **内置常量**：true、false、nil、iota
2. **内置类型**：bool、byte、rune、int、int8、int16、int32、int64、uint、uint8、unit16、unit32、unit64、uintptr、float32、float64、complex64、complex128、string、error
3. **内置函数**：make、len、cap、new、append、copy、close、delete、complex、real、imag、panic、recover
4. **空白标识符**:_

基本组成元素

关键字

- 关键字用于特定的语法结构
- Go语言定义25关键字：

- **声明**：import、package
- **实体声明和定义**：char、const、func、interface、map、struct、type、var
- **流程控制**：break、case、continue、default、defer、else、fallthrough、for、go、goto、if、range、return、select、switch

基本组成元素

字面量

- 字面量是值的表示方法，常用与对变量/常量进行初始化。
- 主要分为：

- 标识基础数据类型值的字面量，例如：0, 1.1, true, 3 + 4i, 'a', "我爱中国"
- 构造自定义的复合数据类型的类型字面量，例如：type Interval int
- 用于表示符合数据类型值的复合字面量，用来构造array、slice、map、struct的值，例如：{1, 2, 3}

基本组成元素

操作符

- **算术运算符**：+、-、*、/、%、++、--
- **关系运算符**：>、>=、<、<=、==、!=
- **逻辑运算符**：&&、||、!
- **位运算符**：&、|、^、<<、>>、&^
- **赋值运算符**：=、+=、-=、*=、/=、%=、&=、|=、^=、<<=、>>=
- **其他运算符**：&(单目)、*(单目)、.(点)、-(单目)、...、<-

分隔符

- 小括号(), 中括号[]，大括号()，分号;，逗号,

基础知识点（一）

声明

- 声明语句用于定义程序的各种实体对象，主要有：

- 声明变量的var
- 声明常量的const
- 声明函数的func
- 声明类型的type

基础知识点（一）

变量

- 变量是指对一块存储空间定义名称，通过名称对存储空间的内容进行访问或修改，使用var进行变量声明，常用的语法为:

1. var 变量名 变量类型 = 值
   定义变量并进行初始化，例如：var name string = "silence"
2. var 变量名 变量类型
   定义变量使用零值进行初始化，例如：var age int
3. var 变量名 = 值
   定义变量，变量类型通过值类型进行推导 例如： var isBoy = true
4. var 变量名1, 变量名2 , ..., 变量名n 变量类型
   定义多个相同类型的变量并使用零值进行初始化 例如：var prefix, suffix string
5. var变量名1, 变量名2 , ..., 变量名n 变量类型 = 值1, 值2, ..., 值n
   定义多个相同类型的变量并使用对应的值进行初始化，例如：var prev, next int = 3, 4
6. var变量名1, 变量名2 , ..., 变量名n = 值1, 值2, ..., 值n
   定义多个变量并使用对应的值进行初始化，变量的类型使用值类型进行推导，类型可不相同，例如：var name, age = "silence", 30

基础知识点（一）

变量

7. 批量定义

```
var (
    变量名1 变量类型1 = 值1
    变量名2 变量类型2 = 值2
)
```

- 定义多个变量并进行初始化，批量复制中变量类型可省略。例如：

```
var (
name string = "silence“
age int = 30
)
```

- 初始化表达式可以使用字面量、任何表达式、函数

vars.go

```go
package main

import "fmt"

var v0 int
var v1 int = 1
var v2 = 2

var v3, v4 int = 3, 4

var v5, v6 = "5", 6

func main() {
    var (
        v7 int = 2 + 5
        v8 int = v2 + v6
    )

    fmt.Println(v0, v1, v2, v3, v4, v5, v6, v7, v8)
}

```

问题 8　输出　调试控制台　终端

```
e:\project\goproject\htgolang\chapter03>go build vars.go

e:\project\goproject\htgolang\chapter03>vars.exe
0 1 2 3 4 5 6 7 8

e:\project\goproject\htgolang\chapter03>
```

基础知识点（一）

赋值

- 可以通过赋值运算=更新变量的值，Go语言支持通过元组赋值同时更新多个变量的值。

例如：

n1, n2 = 1, 2，可用于两个变量的交换x, y = y, x

swap.go

```go
package main

import "fmt"

func main() {
    n1, n2 := 1, 2

    fmt.Println(n1, n2)

    n1, n2 = n2, n1

    fmt.Println(n1, n2)
}
```

问题 11　输出　调试控制台　终端

```
e:\project\goproject\htgolang\chapter03>go build swap.go

e:\project\goproject\htgolang\chapter03>swap.exe
1 2
2 1

e:\project\goproject\htgolang\chapter03>
```

基础知识点（一）

常量

- 常量用于定义不可被修改的的值，需要在编译过程中进行计算，只能为基础的数据类型布尔、数值、字符串，使用const进行常量声明，常用语法：

1. const 常量名 类型 = 值

   定义常量并进行初始化，例如：const pi float64 = 3.1415926

2. const 常量名 = 值

   定义常量，类型通过值类型进行推导，例如：const e = 2.7182818

3. 批量定义

```
const (
常量名1类型1 = 值1
常量名2类型2 = 值2
)
```

基础知识点（一）

常量

定义多个变量并进行初始化，批量复制中变量类型可省略，并且除了第一个常量值外其他常量可同时省略类型和值，表示使用前一个常量的初始化表达式。

例如：

```
const (
name string = "silence"
age int = 30
)
const (
name string = "silence"
desc
)
```

- 常量之间的运算，类型转换，以及对常量调用函数len、cap、real、imag、complex、unsafe.Sizeof得到的结果依然为常量

consts.go

```go
package main

import "fmt"

const c1 int = 1
const c2 = 3 - 1

const c3, c4 int = 3, 4

const c5, c6 = "5", c2 + c4

func main() {
    const (
        c7 int = 7
        c8
    )

    fmt.Println(c1, c2, c3, c4, c5, c6, c7, c8)
}
```

```
问题 8  输出  调试控制台  终端

e:\project\goproject\htgolang\chapter03>go build consts.go

e:\project\goproject\htgolang\chapter03>consts.exe
1 2 3 4 5 6 7 7

e:\project\goproject\htgolang\chapter03>
```

基础知识点（一）

作用域

作用域指变量可以使用范围。go语言使用大括号显示的标识作用域范围，大括号内包含一连串的语句，叫做语句块。语句块可以嵌套，语句块内定义的变量不能在语句块外使用

常见隐式语句块：

- 全语句块
- 包语句块
- 文件语句块
- if、switch、for、select、case语句块

◆ 作用域内定义变量只能被声明一次且变量必须使用，否则编译错误。在不同作用域可定义相同的变量，此时局部将覆盖全局

```go
package main

import "fmt"

func main() {
    outer := 1
    {
        inner := 2
        fmt.Println(outer, inner)
        outer := 3
        fmt.Println(outer, inner)
    }

    //fmt.Println(outer, inner)
    fmt.Println(outer)
}
```

基础知识点（一）

注释

Go支持两种注释方式，行注释和块注释：

- 行注释：以//开头，例如： //我是行注释
- 块注释：以/*开头，以*/结尾，例如：/*我是块注释*/

comment.go

```go
package main

import "fmt"

func main() {
    // 我是行注释
    // 使用fmt包的Println函数将Hello World字符串打印到控制台
    fmt.Println("Hello World")

    /*
        我是块注释
        使用fmt包的Println函数将Hello World字符串打印到控制台
    */
    fmt.Println("Hello World")
}
```

问题 4　输出　调试控制台　终端

```
e:\project\goproject\htgolang\chapter03>go build comment.go

e:\project\goproject\htgolang\chapter03>comment.exe
Hello World
Hello World

e:\project\goproject\htgolang\chapter03>
```

祝大家学业有成

谢　谢