目录

[1 概述 2](#_Toc8982)

[1.1 参考教程 2](#_Toc12251)

[1.2 优秀框架 2](#_Toc15252)

[2 安装环境 2](#_Toc20158)

[3 数据类型 3](#_Toc19694)

[3.1 字典 3](#_Toc23372)

[3.1.1 bool 3](#_Toc17814)

[3.1.2 整数 3](#_Toc30201)

[3.1.3 浮点 4](#_Toc18620)

[3.1.4 字符串 4](#_Toc20378)

[3.1.5 字符 5](#_Toc31933)

[3.1.6 数组 5](#_Toc12938)

[3.1.7 Dictionary 5](#_Toc4656)

[3.2 技巧 5](#_Toc1545)

[4 基础 6](#_Toc28753)

[4.1 常用方法 6](#_Toc9584)

[4.1.1 Print 6](#_Toc6619)

[4.2 内置函数 6](#_Toc29236)

[4.3 类型转化 6](#_Toc13707)

[4.4 自省 6](#_Toc32462)

[4.5 异常 6](#_Toc16356)

[4.6 初始化 7](#_Toc25079)

[4.7 指针 7](#_Toc22508)

[4.8 面向对象 7](#_Toc31981)

[4.8.1 基本类 7](#_Toc26364)

[4.8.2 私有属性，方法 8](#_Toc5260)

[4.8.3 调用父类 8](#_Toc9171)

[4.8.4 与java区别 8](#_Toc23683)

[4.9 常用知识 8](#_Toc13203)

[4.9.1 格式化数字 8](#_Toc19700)

[5 并发 8](#_Toc32591)

[5.1 基础 8](#_Toc7734)

[6 经验总结 8](#_Toc19276)

[6.1 常见陷阱 8](#_Toc25497)

[6.1.1 nil 8](#_Toc17714)

[6.1.2 指针 9](#_Toc17756)

[7 常用 9](#_Toc10415)

[7.1 常用模块 9](#_Toc6961)

[7.1.1 Print 9](#_Toc22119)

[7.2 常用代码 9](#_Toc29701)

[7.2.1 调用DLL 9](#_Toc22303)

[8 第三方模块 9](#_Toc15589)

[8.1.1 beego 9](#_Toc21893)

[8.1.2 Beego 9](#_Toc2107)

[8.1.3 工具 10](#_Toc11019)

# 概述

## 参考教程

## 优秀框架

# 安装环境

GOROOT

值：Go的安装位置

系统变量名：

Path

追加值：;%GOROOT%\bin

Install ssh:

Git clone [https://github.com/golang/crypto.git](https://github.com/golang/crypto.git" \t "http://www.jianshu.com/p/_blank)

git clone出来后，放到$GOPATH/src/golang.org/x/目录下面即可。

Install sftp:

Go get github.com/pkg/sftp

# 数据类型

## 字典

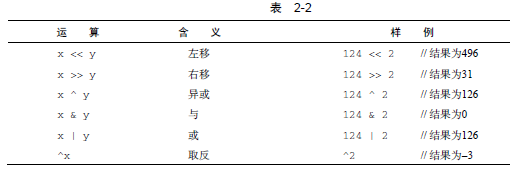
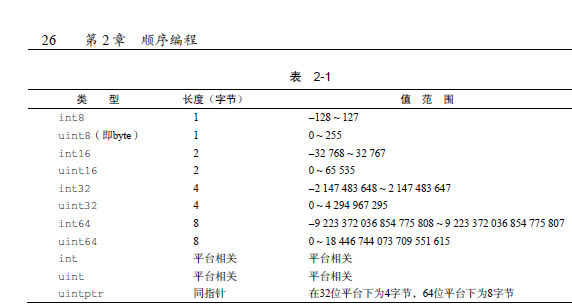
### bool

**var** v1 *bool*

v1 = ***true***

v2 := (1 == 2) // v2也会被推导为bool类型

### 整数



### 浮点

float32, float64

强转化： float32(12)

比较：

// p为用户自定义的比较精度，比如0.00001

**func** IsEqual(f1, f2, p *float64*) *bool* {

**return** math.Fdim(f1, f2) < p

}

### 字符串

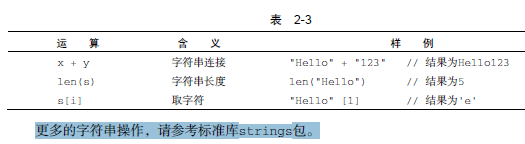
**var** str *string* // 声明一个字符串变量**1**

str = "Hello world" // 字符串赋值

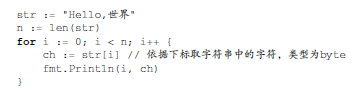
ch := str[0] // 取字符串的第一个字符

fmt.Printf("The length of \"%s\" is %d \n", str, len(str))

fmt.Printf("The first character of \"%s\" is %c.\n", str, ch)



遍历：



### 字符

在Go语言中支持两个字符类型，一个是byte（实际上是uint8的别名），代表UTF-8字符串的单个字节的值；另一个是rune，代表单个Unicode字符。

关于rune相关的操作，可查阅Go标准库的unicode包。另外unicode/utf8包也提供了

UTF8和Unicode之间的转换。

### 数组

[32]*byte* // 长度为32的数组，每个元素为一个字节

[2\*N] **struct** { x, y *int32* } // 复杂类型数组

[1000]\**float64* // 指针数组

[3][5]*int* // 二维数组

[2][2][2]*float64* // 等同于[2]([2]([2]float64))

### Dictionary

id=sa;database=master;pwd=secret']

## 技巧

# 基础

## 常用方法

### Print

print pwd + " is not a good password for " + uid

print "%s is not a good password for %s" % (pwd, uid)

print "Users connected: %d" % (userCount, )

print "Users connected: " + userCount //error

## 内置函数

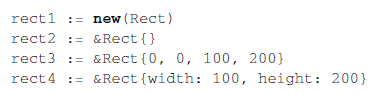
## 类型转化

字符串转字节组： subslice1 := []**byte**("go")

## 自省

## 异常

## 初始化



## 指针

**var** a = [3]int{1, 2, 3}

**var** c = a

    c[1]++

    fmt.Println(c)

**var** b = &a

    b[1]++

    fmt.Println(a, \*b, c)

## 面向对象

### 基本类

### 私有属性，方法

·

### 调用父类

### 与java区别

## 常用知识

### 格式化数字

# 并发

## 基础

# 经验总结

## 常见陷阱

### nil

[golang的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。nil是预先说明的标识符，也即通常意义上的关键字。在golang中，nil只能赋值给指针、channel、func、interface、map或slice类型的变量。如果未遵循这个规则，则会引发panic。](http://10.139.32.180:8888/subversion/sqlscript/branch/201511)

### 指针

# 常用

## 常用模块

### Print

## 常用代码

### 调用DLL

<http://studygolang.com/topics/78>

### 文件操作

http://www.open-open.com/lib/view/open1476274079468.html

# 第三方模块

### beego

### Beego

### 工具

go get -u -v github.com/nsf/gocode

go get -u -v github.com/rogpeppe/godef

go get -u -v github.com/golang/lint/golint

go get -u -v github.com/lukehoban/go-find-references

go get -u -v sourcegraph.com/sqs/goreturns

go get -u -v golang.org/x/tools/cmd/gorename

go get -u -v github.com/derekparker/delve/cmd/dlv

go get -u -v github.com/rogpeppe/gurn