

HelloWorld程序解释及Go编码规范

目录：

1. 第一个HelloWorld程序解释
2. Go语言编码规范
3. Go语言关键字及保留字
4. Go程序结构组成

一、HelloWorld程序解释

(一)、第一个HelloWorld程序

```
package main
import "fmt"
func main() {
    /* 这是第一个简单的程序 */
    fmt.Println("Hello, World!")
}
```

(二)、程序解释

1. 第一行代码 package main 定义了包名。必须在源文件中非注释的第一行指明这个文件属于哪个包，如：package main。package main表示一个可独立执行的程序，每个 Go 应用程序都包含一个名为 main 包。
2. 下一行 import "fmt" 告诉 Go 编译器这个程序需要使用 fmt 包，fmt 包实现了格式化 IO（输入/输出）的函数。
3. 下一行 func main() 是程序入口。main 函数是每一个可执行程序所必须包含的，一般来说都是在启动后第一个执行的函数，如果有 init() 函数则会先执行init()函数。
4. 下一行 /*...*/ 是注释，在程序执行时将被忽略。单行注释是最常见的注释

形式，你可以在任何地方使用以 `//` 开头的单行注释。多行注释也叫块注释，均已以 `/*` 开头，并以 `*/` 结尾，且不可以嵌套使用，多行注释一般用于文档描述或代码片段。

5. 下一行 `fmt.Println(...)` 可以将字符串输出到控制台，并在最后自动增加换行字符 `\n`。使用 `fmt.Print("hello, world\n")` 可以得到相同的结果。

二、Go语言编码规范

（一）、注释

- 单行注释是最常见的注释形式，你可以在任何地方使用以 `//` 开头的单行注释
- 多行注释也叫块注释，均已以 `/*` 开头，并以 `*/` 结尾，且不可以嵌套使用，多行注释一般用于文档描述或注释成块的代码片段

（二）、标识符

- 1、标识符用来命名变量、类型等程序实体。一个标识符实际上就是一个或是多个字母(A~Z和a~z)数字(0~9)、下划线_组成的序列，但是第一个字符必须是字母或下划线而不能是数字。
- 2、Go不允许在标识符中使用@、\$和%等标点符号。
- 3、Go是一种区分大小写的编程语言。因此，Manpower和manpower是两个不同的标识符。
- 4、以下是无效的标识符：
 - 1xy（以数字开头）
 - case（Go 语言的关键字）
 - chan（Go 语言的关键字）
 - x+y（运算符是不允许的）

（三）、Go 语言的空格

- 1、Go 语言中变量的声明必须使用空格隔开，如：`var age int;`
- 2、语句中适当使用空格能让程序更易阅读。
- 3、在变量与运算符间加入空格，程序看起来更加美观，如：
 - `a = x + y;`

（四）、语句的结尾

- 1、在 Go 程序中，一行代表一个语句结束。Go 语言中是不需要类似于Java需要分号结尾，因为这些工作都将由 Go 编译器自动完成；
- 2、如果打算将多个语句写在同一行，它们则必须使用 分号“;” 人为区分，但在实际开发中并不鼓励这种做法。

（五）、可见性规则

- 1、Go语言中，使用大小写来决定标识符（常量、变量、类型、接口、结构或函数）是否可以被外部包所调用。
- 2、以一个大写字母开头，那么使用这种形式的标识符的对象就可以被外部包的代码所使用（使用时程序需要先导入这个包），如同面向对象语言中的 public。
- 3、如果以小写字母开头，则对包外是不可见的，但是他们在整个包的内部是可见并且可用的，像面向对象语言中的 private 。

三、Go语言关键字及保留字

下面列举了 Go 代码中会使用到的 25 个关键字或保留字：

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

除了以上介绍的这些关键字，Go 语言还有 36 个预定义标识符：

append	bool	byte	cap	close	complex	complex64
copy	false	float32	float64	imag	int	int8
int32	int64	iota	len	make	new	nil
print	println	real	recover	string	true	uint

四、Go 程序结构组成

(一)、Go一般结构

```
// 当前程序的包名
package main

// 导入其他包
import . "fmt"

// 常量定义
const PI = 3.14

// 全局变量的声明和赋值
var name = "gopher"

// 一般类型声明
type newType int

// 结构的声明
type gopher struct{}

// 接口的声明
type golang interface{}

// 由main函数作为程序入口点启动
func main() {
    Println("Hello World!")
}
```

(二)、Go 文件的基本组成部分

1. 包声明
2. 引入包

3. 函数
4. 变量
5. 语句 & 表达式
6. 注释

(三)、Go文件结构组成

- 1、Go 程序是通过 package 来组织的。
 - 只有 package 名称为 main 的包可以包含 main 函数。
 - 一个可执行程序有且仅有一个 main 包。
 - 通过 import 关键字来导入其他非 main 包。
 - 可以通过 import 关键字单个导入，也可以同时导入多个
- 2、程序一般由关键字、常量、变量、运算符、类型和函数组成。
- 3、程序中可能会使用到这些分隔符：括号 (), 中括号 [] 和大括号 {}。
- 4、程序中可能会使用到这些标点符号：
 - 点.
 - 逗号,
 - 分号;
 - 冒号:
 - 省略号(三个点) ...
- 4、通过 const 关键字来进行常量的定义。
- 5、通过在函数体外部使用 var 关键字来进行全局变量的声明和赋值。
- 6、通过 type 关键字来进行结构(struct)和接口(interface)的声明。
- 7、通过 func 关键字来进行函数的声明。

