go 语言学习笔记

1. 基础语法

- 导入： `import ( ... )`
- 方法： `func`
- 多返回值函数：

```
func swap(x, y string) (string, string){
     return y, x;
}
a, b := swap("a","b");
```

- 变量声明：

```
var $var_name $var_type
var $var_name1, $var_name2 $var_type = $var_init1, $var_init2

//短声明
// := go 会进行类型推导？？ 只能在函数内部使用
```

- 基本数据类型

```
bool
string
int int8 int16 int32=rune int64
uint uint8=byte uint16 uint32 uint64
float32 float64
complex64 complex128
```

- 常量定义： `const`
- `defer` 函数：将函数进行延迟压栈处理， 详细细节

```
func test_defer(){
    defer fmt.Println("2");
    defer fmt.Println("4");
    fmt.Println("1");
    fmt.Println("3");
}

//result: 1342
```

> defer 规则：
>
> 1. *A deferred function's arguments are evaluated when the defer statement is evaluated.*

> 2. *Deferred function calls are executed in **Last In First Out (method stack)** order after the surrounding function returns.*
> 3. *Deferred functions may read and assign to the returning function's named return values.*

- 指针：默认指针值为 `nil` ，跟 c 语言不同， **不支持指针运算**

```
var p *T
```

- 结构体

```
type $name struct{
    ...
    ...
}

//使用时 $name{...}，注意是大括号
```

- 数组

```
var $name [$length]$type
var a [10]int
var b [2]string
$name := [...]$type{$item1,$item2} // [...]可以由编译器计算出长度
```

- `slice` 切片，跟 python 类似， **拓展阅读**

```
//使用 make 函数创建 slice
func make([]T, len, cap) []T
//len 初始长度
//cap 最大容量

func copy(dst, src []T) int
```

- 内建函数： https://golang.org/pkg/builtin

```
//常用内建函数
make
delete
copy
```

- `map`

```
map_name = make(map[T]I)
```

- 函数值，即函数指针
- 复写已有函数：

```
type Postion struct {
    x float32
    y float32
}

//类似于 java 的 toString 函数
func (p Postion) String() string {
    return fmt.Sprintf("x is: %v, y is: %v", p.x, p.y)
}
```

- `goroutine` go 语言的轻量级线程
- `channel` 通道
- 常用链接：

> 标准库： https://golang.org/pkg/
>
> 语言规范： https://golang.org/ref/spec
>
> 内建函数： https://golang.org/pkg/builtin
>
> effective go： https://golang.org/doc/effective_go.html