GO语言学习笔记

变量：一段或多段用来存储数据的内存，类型决定了变量内存的长度和存储格式。简短模式并不总是重新定义变量，可能是部分退化的赋值操作（P19），前提：最少有一个新变量被定义且必须是同一作用域，err变量可以重复使用。

常量：不使用也不会引起编译错误，在常量组中如不指定类型和初始化值，则与上一行非空常量右值相同（P25）。

枚举：可借助iota标识符实现一组自增常量来实现枚举类型（P25）。

常量与变量的区别：变量在运行期分配存储内存，常量通常会被编译器在预处理阶段直接展开，作为指令数据使用。

引用类型：slice、map、channel，使用make函数创建。

未命名类型：数组、切片、字典、通道等类型与具体元素类型或长度属性有关，可用type为其提供具体名称（P35）。

25个关键字：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 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 |

指针：是一个实体，用来保存地址的整型变量（P42）。在 Go 语言中，通过指向结构体的指针获取结构体中的值时，可以使用点操作符直接访问字段，而无需显式地使用 \* 运算符进行解引用。这是因为 Go 语言在访问指针指向的结构体字段时，会自动解引用指针并访问字段。

函数：结构化变成的最小模块单元，函数只能判断其是否为nil（P59）。形参是指函数定义中的参数，实参则是函数调用时所传递的参数。

变参：本质上是一个切片。只能接受一个到多个同类型参数，且必须放在列表尾部（P66）。

闭包：在其词法上下文中引用了自由变量的函数，或者是函数和其引用的环境的组合体（P71）。