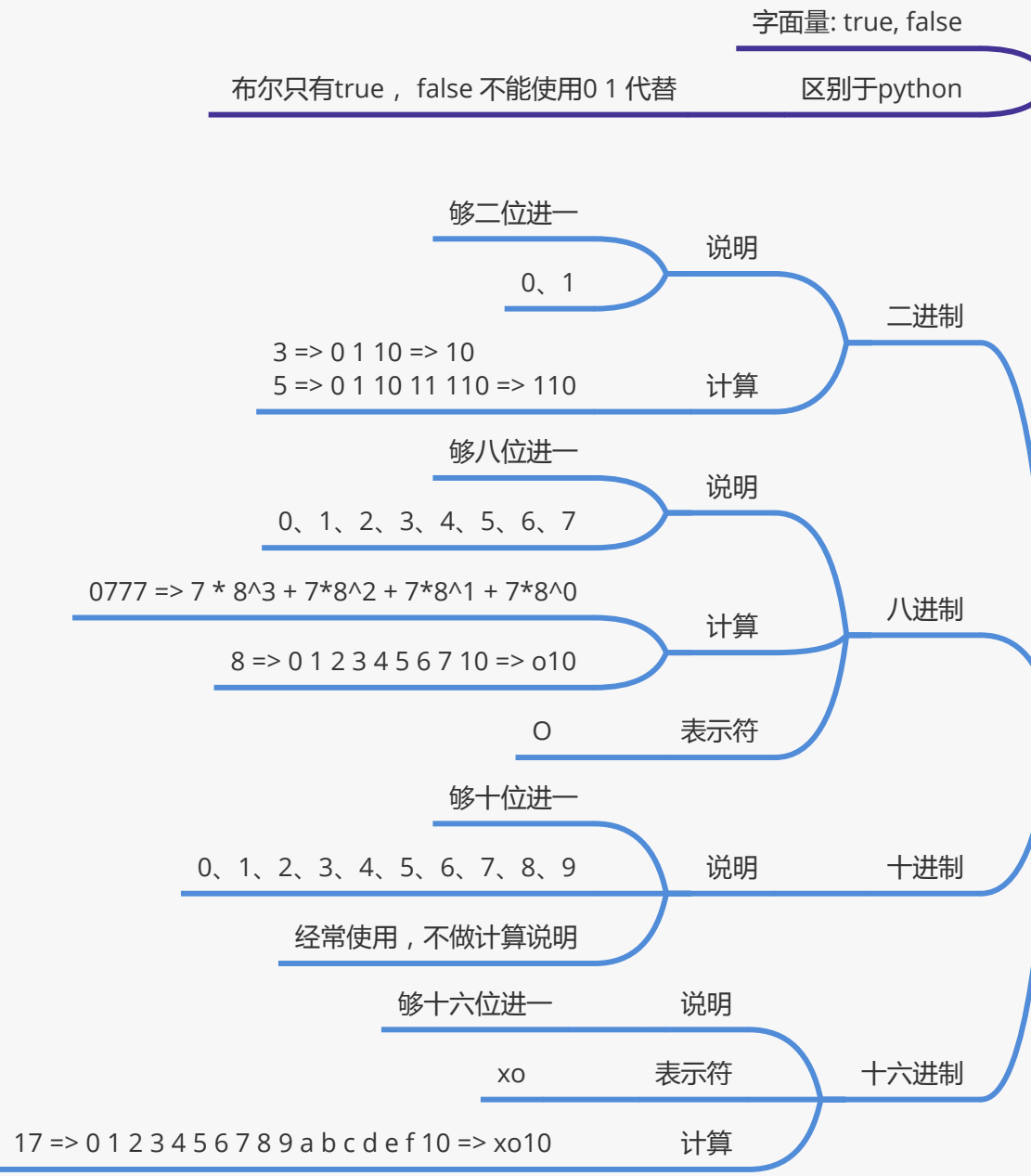


chapter01

布尔

布尔只有true, false 不能使用0 1 代替  
字面量: true, false  
区别于python

进制



作用域

作用域

- 分为: 包、函数、语句块作用域
- go语言中使用{}定义作用域的范围
- 如果同一个变量名称在父子作用域内声明赋值时，在各自作用域内打印值不同
- 作用域没有强制代码上下关联，代码靠前的函数内引用代码靠后的函数不会报错。python是不可以这么写的。
- 定义标识符可以使用的范围
- 使用原则
  - 父作用域内的变量可被子作用域变量调用，反之不可以

指针

指针

- 定义指针
  - var \*int
- 取指针地址
  - &pointer
- 为何使用指针
  - 1、当需要处理的数据庞大时，为了节省传输带宽，只传递地址相对好一些。比如接口嵌套、结构体嵌套等等
  - 2、基于最底层数据变更时使用。如果传递该变量参数时，go只是将其copy一份存入新的地址中，如果再将其改值时，只是改变新地址的那个值，而原来的参数值不变。这样一来再调用原来参数时值未变。

变量

变量

- 关键字: var
- 定义方式
  - var d int // 定义整型d并初始化0
  - var d int = 32 // 定义整型d并赋值32
  - var ( d int = 32 sd int = 6 ) // 多定义
  - var ( d int = 32 s string = "ligz" ) // 多定义不同类型的
  - var c = 32 // 根据后边值类型推导属于哪种类型的
- 只能在函数内定义的方法
  - d := 32

常量

常量

- 关键字: const
- 定义方式
  - const NAME int = 32
  - const ( NAME string = "张三" AGE int = 64 )
- 常量是不可变更的，定义值后无法在进行赋值操作
- 常量规范定义应该以全字母大写
- 必须设定默认值，否则报错
  - const NAME int // 报错

字符串

字符串

- 字面量: ""、``
- 字符串可 拼接
  - "张三" + "是小狗"
- 算数运算符
  - +
- 赋值
  - s := "我叫"

if、for、switch

if、for、switch

- for
  - for {} // 死循环。相当于别的语言中 while true
  - for 初始化子语句;条件子语句;后置子语句{ }
  - for range 多用于 切片、数组、管道、映射、字符串
- if
  - if 条件表达式 { }else if 条件表达式 { }else{ }
  - if 语句块中可嵌套if
- switch
  - switch 变量 { case 值: 内容 }
  - switch { case 条件表达式: 内容 case 条件表达式, 条件表达式2 ... : 内容 }

整型

整型

- 有符号
  - int\int8\int32\int64
  - 128 ~ 127      -2^7 - 2^7-1
- 无符号
  - uint\uint8\uint32\uint64
  - 0 ~ 255      2^8-1
- 定义
  - var int
  - var int8
- 八字节中 第一位为正负号描述，所以带符号的看作7位 符号占一位
- 类型间强转
  - int => int(int8)
- 算数运算
  - +, -, \*, /, %
- 关系运算
  - ==, !=, >, >=, <, <=
- 位运算
  - 十进制 => 二进制      7/(除以)2 => 余1 3(等于)2 => 1 1/2 => 1 0 => 0111
  - &, |, ^, <<, >>, &^

浮点型

浮点型

- float32
- float64
- 定义
  - var float4 = 3.14
  - var float64 = 3.144
- 字面量: 十进制表示法, 科学计算表示法
  - 十进制: 30 40
  - 科学计算
    - E的n次方。MEN => M \* 10^n
    - 1.9 E -1 ==> 0.19
- 算数 运算符
  - +, -, \*, /, ++, --
- 关系运算符
  - >, >=, <, <=
  - 没有等于，但可以写成      1.3-1.2 <= 0.05 0.05为该浮点值在0.05范围内
- 赋值
  - =, +=, -=, /=, \*=
- 复数
  - ii := 1 + 2i

goto

goto

- 跳转到标签位置
- 标签
  - 一般标签以全字母大写方式
  - TAG:
- 如果for能搞定的事情尽量不使用goto去处理
- 例子说明
  - 可在多循环时，最内层循环得到结果后想直接退出时，可用brack TAG 跳转标签处
  - 如果使用brack TAG 跳出for循环情况下，需要确保TAG: 标签一定在brack所处的for循环上方。否则报错。