

2.如何学习一门程序语言

程序的本质

程序的本质是计算机指令

程序语言就是人与计算机交互的桥梁

程序语言是有严格的逻辑的，逻辑清晰有规律可循

低级语言和高级语言 0 1

程序语言的通性

一切来源于生活，程序设计的初衷就是为了解决生活中遇到的问题。

所有的程序语言都是由**数据结构+算法**组成的。

数据结构决定了我们数据在计算机当中是如何存储的，以及各项数据之间的联系。

算法决定了我们解决问题的步骤。

三个概念：

1. **数据结构**：是相互之间存在一种或多种特定关系的数据元素的集合，包括逻辑结构和物理结构。
2. **数据类型**：是一个值的集合和定义在这个值集上的一组操作的总称。
3. **抽象数据类型**：是指一个数学模型以及定义在该模型上的一组操作。

大部分程序语言都有的东西

变量、常量

age = 11 (字面量)

rmb = 11

变量和常量就是给字面量命名，赋予含义。变量就是它的值是可以变化的，常量就是它的值不能变化

数据类型：

- 数字类型
- 字符串
- 列表
- 字典 (map)
- 布尔类型
- 集合?

条件判断:

- if条件
- switch条件 (python没有)

循环:

- for循环
- while循环

函数:

- 函数声明、定义
- 函数参数与返回值
 - 位置参数
 - 关键字参数
 - 可变参数
 - 参数默认值
 - 单返回值、多返回值
- 匿名函数
- 函数递归
 - 尾递归的优化
- 闭包

运算符、表达式和语句:

- 算术运算符
 - 加
 - 减
 - 乘
 - 除
 - 取余
 - 幂运算
- 比较运算符
 - 大于
 - 等于

- 小于
 - 不等于
- 逻辑运算符
 - 与
 - 或
 - 非
 - 异或
- 位运算符
 - 按位与
 - 按位或
 - 按位异或
 - 按位取反
 - 左移
 - 右移
- 什么是表达式?
- 什么是语句
 - 声明语句
 - 赋值语句

错误和异常

- 什么是错误
 - 语法错误
 - 逻辑错误
- 什么是异常
 - 常见异常
- 异常捕获和处理

面向对象（OOP语言才有）：

- 类
- 父类、子类
- 类继承
- 类的实例
- 类方法和实例方法
- 方法的重写，重载重定义
- 静态方法
- 属性
- 属性的访问权限
- 方法的访问权限

结构体（非oop语言）：

- 属性
- 嵌套