Day-5

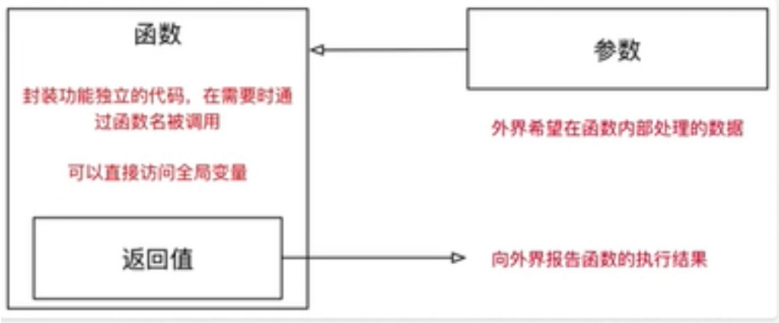
- Video-226-242变量进阶
  - 变量的引用
    - 变量和数据都是保存在内存中的
    - 在Python中函数的参数传递以及返回值都是靠引用传递的
  - 引用的概念
    - **变量** 和 **数据** 是分开存储的
    - **数据** 保存在内存中的一个位置
    - **变量** 中保存着数据在内存中的地址
    - **变量** 中 **记录数据的地址**，就叫做 **引用**
    - 使用 id() 函数可以查看变量中保存数据所在的 **内存地址**
    - 注意⚠️
      - 当给变量赋值时，本质上是**修改了数据的引用**
      - 变量 **不再** 对之前的数据引用；变量 **改为** 对新赋值的数据引用
  - 函数的参数和返回值的传递

    - 参数和返回值都是靠**引用**来传递的
  - 数据类型：可变 & 不可变类型
    - 可变：列表、字典，可以修改数据内容，但是**不改变内存地址**
      - 修改列表&字典的内容需要通过**调用相关方法实现**，**赋值的话是开一段新的内存地址，修改引用**
      - 字典key只能使用不可变类型的数据
    - 不可变：数字类型、字符串、元组
  - 哈希（hash）
    - hash函数：接收一个不可变类型的数据作为参数，返回结果是一个整数
    - 作用：**提取数据的特征码（指纹）**
      - 1.相同的内容得到相同的结果
      - 2.不同的内容得到不同的结果

- 字典的键值对设置，**首先会对key进行hash，以决定如何在内存中保存字典的数据**
  - 1.key必须是不可变类型数据
  - 2.value可以是任意类型的数据

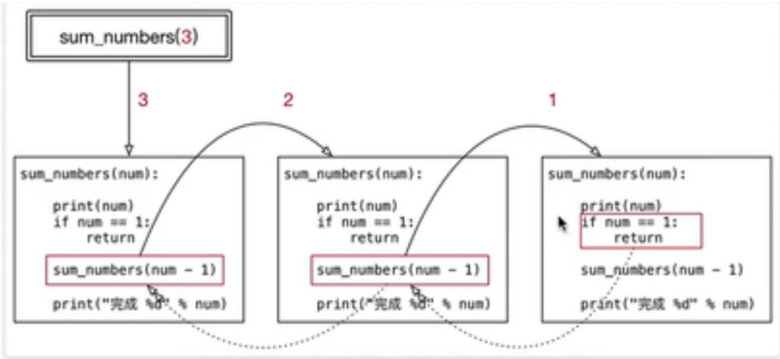
- 局部变量 & 全局变量
  - 局部变量：函数内部定义的变量，只能在函数内部使用
    - 函数内部的局部变量在该函数执行完之后，会被系统回收
    - 作用：临时保存函数内部需要的数据
    - 局部变量的生命周期
      - 1.生命周期：从被创建到被系统回收的过程
      - 2.**局部变量在函数执行时才会被创建**
      - 3.函数执行结束后，局部变量被系统回收
      - 4.局部变量在生命周期内，可以**用来存储函数内部临时使用的数据**
    - 不同函数可以定义相同名字的局部变量，彼此之间不会产生影响
  - 全局变量：在函数外部定义的变量，所有函数内部都可以使用这个变量
    - **函数不能直接修改全局变量的引用**
    - **全局变量可变范围太大会导致程序不好维护**
    - global关键字：**告诉解释器后面的变量是一个全局变量**，再使用赋值语句时，就不会创建局部变量
    - 全局变量的位置：为了保证所有的函数都能够正确使用到全局变量，应该 **将全局变量定义在其他函数的上方**
    - 代码结构示意图

        - 全局变量命名的建议：避免与局部变量出现混淆
- Video-243-259函数进阶
    - 函数参数和返回值的作用



        - 定义函数时，是否接收参数或者是否返回结果，是根据实际的功能需求来决定的
            - 如果函数 **内部处理的数据不确定**，就可以将外界的数据以参数传递到函数内部
            - 如果希望一个函数 **执行完成后，向外界汇报执行结果**，就可以增加函数的返回值
    - 函数的返回值：函数完成工作后，最后给调用者的一个结果
        - 调用函数一方，可以使用变量来接收函数的返回结果
    - 函数的参数
        - 不可变 & 可变的参数
            - **针对参数使用赋值语句，会在函数内部修改局部变量的引用，不会影响到外部变量的引用**
            - 可变类型，函数内部**使用方法修改了数据内容，则会影响外部变量的引用**
        - **+=：在 python 中，列表变量调用 += 本质上是在执行列表变量的 extend 方法，不会修改变量的引用**
    - 缺省参数：定义函数时，可以给某个参数指定一个默认值，**具有默认值的参数就叫做缺省参数**
        - 调用函数时，如果没有传入缺省参数的值，则在函数内部使用定义函数时指定的参数默认值
        - **将常见的值设置为参数的缺省值，从而简化函数的调用**
        - 注意⚠️
            - 缺省参数的定义必须放在参数列表的末尾
            - 在 **调用函数时**，如果有 **多个缺省参数**，**需要指定参数名**，这样解释器才能够知道参数的对应关系！
    - 多值参数
        - 有时**可能需要一个函数能够处理的参数个数是不确定**的，这个时候，可以使用多值参数
        - 两种多值参数
            - 1.参数名前加一个*，可以接收元组，*args

        - 2.参数名前加两个**，可以接收字典，**kwargs
- 元组 & 字典的拆包
    - 在调用带有多值参数的函数时，如果希望：
        - 将一个 **元组变量**，直接传递给 args
        - 将一个 **字典变量**，直接传递给 kwargs
    - 就可以使用 **拆包**，简化参数的传递，**拆包** 的方式是：
        - 在 **元组变量前**，增加 **一个** *
        - 在 **字典变量前**，增加 **两个** *
- 递归
    - 特点：一个函数内部调用自己
    - 代码特点
        - 1.函数内部的代码是相同的，只是针对参数不同，处理的结果不同
        - 2.当参数满足一个条件时，函数不再执行（**这个参数非常重要，否则会出现死循环**）
        - 递归函数执行流程



以上内容整理于 幕布文档