
Python基础入门升级版
火热开团
从零开始学 Python

(三)、函数

七月在线 David

2017年12月2日

复习

- 熟悉Python中序列的概念
- 掌握对python容器类型元素的各类操作
- 理解变量与对象之间的引用关系

本节课程目标

- 了解函数的意义，掌握函数创建及使用
- 掌握函数中参数使用
- 了解偏函数，匿名函数
- 理解高阶函数的概念
- 掌握BIFs中的map(),reduce(),filter()函数的使用
- 理解变量的作用域
- 了解闭包，装饰器，函数式编程的概念

Python中的函数

- 函数的意义:
  - 1.对输入进行变换映射后输出
  - 2.过程化 VS 结构化
- 函数的创建及结构:
  - 定义函数名
  - 参数
  - 函数体
  - 返回
    - 有无返回
    - return与yield的区别

函数中的参数

- 参数:
- 语法：func(positional_args,keyword_args,*tuple_nonkw_args,**dict_kw_args)
- 按参数传递方式:
  - 位置参数(定位参数，非关键字参数) :位置顺序十分重要
  - 关键字参数:
  - 位置参数包裹及使用*
  - 关键字参数包裹及使用**
  - 包裹解包顺序
  - 传递参数时使用包裹
- 按参数的类型:
  - 必选（位置参数）
  - 关键字/默认
  - *args可变长位置参数，**kwargs可变长关键字参数
- 函数如何处理传入参数:
  - 值传递参数与指针传递参数

变量作用域

- 标识符的作用域
- 全局变量
  - 定义
  - 经过函数时
  - 然后，除非被del
- 局部变量
  - 函数内部创建与访问
  - 函数被调用结束时，
- 变量的搜索顺序
  - 覆盖问题
  - 局部作用域->全局作用域

□locals()#局部名称空间

□globals()#全局名称空间

偏函数PFA

- 偏函数Partial function application
  - 使用场景：如果一个函数的参数很多，而在每次调用的时候有一些又经常不需要被指定时，就可以使用偏函数（近似理解为默认值）
  - 语法：partical(func,*args,**keywords)
  - 使用：from functools import partial
  - 原理：创建一个新函数，固定住原函数的部分参数（可以为位置参数，也可以是关键字参数）

递归函数

- 定义:
  - 函数在内部调用自身
- 例子:
  - 求一个列表中数值的累加

高阶函数

- 函数的引用与调用
  - 引用：访问，变量别名(多个别名引用)
  - 调用：()
- 函数对象既然可以被引用，那可以作为参数被传入或作为结果被返回吗?
- 高阶函数:
  - 一个函数接收另一个函数作为参数
- 回调函数:
  - 函数作为调用函数的结果返回

BIFs中的高阶函数

- filter
  - 核心点：对每个元素做过滤
- map
  - 核心点：对每个元素做映射
- reduce
  - 核心点：两两传给func
  - Python 3.x中,reduce()函数已经被从全局名字空间里移除了,它和partical一样被放置在fucntools模块中。使用前需要调用.

匿名函数lambda

- 匿名函数lambda:
  - 使用场景:
    - 返回简单结束， 不需要专门定义函数
  - 特点:
    - 简洁，同一行定义体+声明。不用写return
  - 定义:
    - 定义后，赋值给一个变量,做正常函数使用
    - lambda关键字定义在高阶函数的参数位上
  - 语法:
    - lambda(args1,args2, argsN):expression

闭包Closure

- 闭包的概念
  - 涉及嵌套函数时才有闭包问题
  - 内层函数引用了外层函数的变量（参数），然后返回内层函数的情况，称为闭包（Closure）。
  - 这些外层空间中被引用的变量叫做这个函数的环境变量。
  - 环境变量和这个非全局函数一起构成了闭包。
- 闭包的特点:
  - 函数会记住外层函数的变量
- 闭包的实现:

装饰器Decorator

- **定义:**
  - 以函数作参数并返回一个替换函数的可执行函数
- **简单理解:**
  - 装饰器的作用就是为已存在的对象添加额外功能
  - 为一个函数增加一个装饰（用另一个函数装饰)
  - 本质上就一个返回函数的高阶函数
- **应用:**
  - 给函数动态增加功能(函数)
- **定义与使用:**

函数式编程Functional Programming

- **函数式编程思想:**
  - 函数是第一等公民First Class
  - 函数式编程是一种编程范式，是如何编写程度的方法论。把运算过程尽量变成一系列函数的调用。属于结构化编程

- **特点:**
  - 允许函数作为参数，传入另一个函数
  - 返回一个函数

- **理解:**
  - 结合本章知识点，案例进行理解

习题1

- 定义一个函数，接收任意3个数字的输入，并按顺序从小到大输出

习题2

- 要求创建一个函数，它可以接收，位置参数，不定长位置参数，不定长关键词参数，并按要求输出 。
  - 输入班级名，班级特色（如'勤奋','颜值高'）等等不同特色，班级不同同学的姓名与年龄。
  - 要求输出，班级名，班级特色，班级成员，班级成员的平均年龄。

习题3

- 使用**reduce**函数实现找出一组数字列表中的最大值

习题4

- 求1000以内能同时被3和7整除的数有哪些。要求使用map与filter函数

习题4

- 体现闭包的思想，创建一个三层嵌套的函数，并调用。

习题5

- 请以**round**函数，定义一个偏函数**roundN**，调用为输入一个数字**N**，返回圆周率后**N**位的数字
  - 提示:
  - import math
  - math.pi

- 请以**sorted**函数，定义一个偏函数**sortedDESC**，结果为输入一个序列，返回为按降序排列后序列。

习题6

- 要求使用**map**与**filter**函数，输出一个输入字符串里每个字符出现的次数
  - 提示：结合**dict**使用实现

习题7

- 创建一个能够快递排序的递归函数

习题8

- 创建一个能打印一个字典中包含所有字典对象的递归函数

```
dic = {
    "北京": {
        "东城": ["天坛", "东单", "王府井"],
        "西城": ["西单", "军博", "复兴门"],
    },
    "上海": {
        "杨浦":['五角场','政通路','世界路']
 },
    "天津":['天津港']
}
```

习题9

- 编写一个生成器，实现**fib**数列的效果

作业1

- 创建一个能接收不定长位置参数（数字）函数，返回是所有参数的和。
- 现要求在这个函数每次调用时都有进行屏幕打印（该函数被调用）。
- 请在不改动这个函数内部及调用的前提下实现。