

day09: 函数

笔记本: Python基础
创建时间: 2018/6/13 16:33 更新时间: 2018/7/4 19:52
作者: liuchang_0412@163.com
URL: <https://www.zhihu.com/question/57726430>

#####

##装饰器

#####

- 1.定义: 装饰器是一个函数用来包装另一个函数
- 2.作用: 用来装饰别的函数, 给其他函数增加新功能
- 3.原则:

- ①不能修改被装饰的源函数代码
- ②不能修改被装饰函数的调用方式

4.三句话:

- ①函数即变量
- ②高阶函数: 以函数作为参数, 或者返回值是函数的函数
- ③函数嵌套: 函数里面定义函数
- ④结论: 装饰器 = 函数嵌套 + 高阶函数

#练习: 现有一个求和函数

```
def my_sum(*args):  
    return sum(args)
```

现需要增加功能, 如果不传入参数时显示什么也没有, 如果输入非int类, 显示输入错误。

```
def my_sum(*args):  
    if len(args) == 0:  
        return '什么也没有'  
    for value in args:  
        if not isinstance(value, int):  
            return '输入错误'  
    return sum(args)
```

```
print(my_sum())
```

问题:

- 1.装饰器与函数调用的区别?

#####

1.函数 function: 定义用户级的函数

①定义：函数是可以重复执行的代码块，可以重复使用

②作用：定义用户级的函数

③语法：

```
def 函数名(参数列表):  
    语句块（代码块） ...
```

④语法说明：

函数名是语句块的名称

函数名的命名规则与变量名相同

函数的名字是一个变量，是绑定代码块的名称

函数有自己的名字空间，要让函数处理外部数据需要用参数给函数传入一些数据，
如果不需要传入参数，参数列表可以为空

语句块部分不能为空，如果为空则需要填充pass语句

2.函数的调用：

函数名（实际参数）

①函数调用是一个表达式

②如果没有return语句，函数执行完毕返回None值对象

③如果函数需要返回其他的值对象，需要用return语句

3.return语句

语法：return [表达式] []代表可以省略

作用：结束当前函数的执行，返回到调用该函数的地方，同时返回一个值对象的引用关系。

说明：

1.return语句后跟的表达式可以省略，省略后相当于return None

2.如果函数内没有return语句，则函数执行完最后一条语句后返回None

3.函数的调用可以返回一个值或一组值

#练习：定义两个函数：

sum3(a, b, c) 用于返回三个数的和；

pow3(x) 用于返回x的三次方（立方）

用以上两个函数计算：

1.计算1的立方+2的立方+3的立方

2.计算1+2+3的立方

4.函数的参数传递

①传递方式：

位置传参

* 序列传参

关键字传参

②位置传参：实际参数的对应关系与形式参数的对应关系是以位置来依次对应的

示例：

```
def sum3(a, b, c):  
    return a + b + c  
# 1-->a, 2-->b, 3-->c  
x = sum3(1, 2, 3)
```

位置传参说明：

实参与形参通过位置对应和传递

实参和形参的个数必须完全相同

③* 序列传参：序列的元素个数必须与参数列表的个数相同

示例：见 args.py

④关键字传参：是指传参时，按着形参的名称给形参赋值。形参和实参按照名称匹配。

示例：见 args.py

⑤字典关键字传参：实参和形参通过字典进行传递和匹配，字典的值传递给键对应的形参。

示例：见 args.py

说明：字典传参的键名和形参名必须一致

键名必须为字符串

键名必须要在形参中存在

——以下讲函数的定义——

5.函数的缺省参数

语法：

```
def 函数名(形参1=默认参数1, 形参2=默认参数2, ...):
```

语句

示例：

```
def info(name, age=1, addr="不详"):  
    print("姓名", name, "年龄：", age, "住址：", addr)
```

说明：

缺省参数必须自右至左依次存在，如果一个参数有缺省值则其右侧的所有参数必须有缺省值。

缺省参数可以有0个或者多个，甚至全部参数都有缺省值

#练习：写一个函数sum4(a,b,c,d)来计算4个参数的和，可以用如下方法调用：

```
print(sum4(1,2))  
print(sum4(1.1,2.2,3.3))  
print(sum4(100,200,300,400))
```

6.函数的不定长参数：* 元组形参，** 字典形参

①* 元组形参：

语法:

```
def 函数名(*元组形参):  
    语句
```

示例: 见star_args.py

②命名关键字形参 (命名参数配合默认参数使用可以简化代码)

语法:

```
def 函数名(*, 命名关键字形参)  
    语句块  
def 函数名(*args, 命名关键字形参)  
    语句块
```

示例: 见named_args.py

③** 字典形参:

语法:

```
def 函数名(**字典形参)  
    语句
```

示例:

```
def myfun(**kwargs):  
    print("参数个数:", len(kwargs))  
    for k,v in kwargs.items():  
        print(k,">>",v)  
#调用  
myfun(name="tarena",age=15)  
myfun(a=1,b="bbb",c=[1,2,3],d=True)
```

#练习: 写一个函数minmax(), 有不定长个参数, 返回这些参数的最大值和最小值, 形成元组, 最小值在前, 最大值在后, 调用次函数得到最大值和最小值并打印。

7.函数的参数列表顺序:

①位置形参, 缺省参数, *元组形参, **字典形参, 命名关键字参数可以混合使用

②函数参数自左至右的顺序为:

位置形参, *元组形参, 命名关键字参数, **字典形参

示例: def fn(a, b, *arg, c, **kwargs):

```
    pass  
fn(100, 200, 300, 400, c="C", d="D", e="E")
```

#练习:

prinme(素数) :

1.写一个函数isprime(x), 判断x是否为素数, 如果为素数返回True, 否则返回False。

- 2.写一个函数prime_m2n(m,n)，返回从m开始到n结束范围内的素数列表，并打印。
- (如L=prime_m2n(5,10) print(L) #[5,7])
- 3.写一个函数primes(n)，返回指定范围内的全部素数的列表，并在主程序中打印。
- (如L=primes(100) print(L) #[2,3,5,7,...,97])