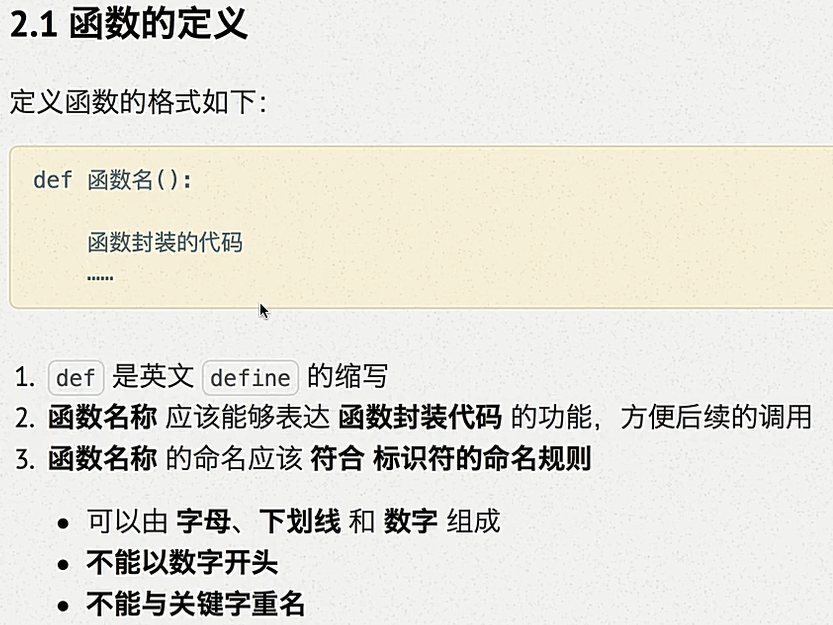
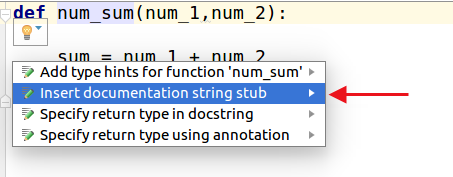
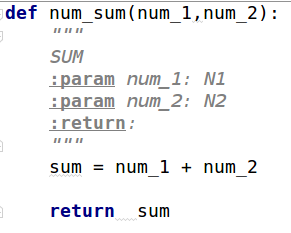
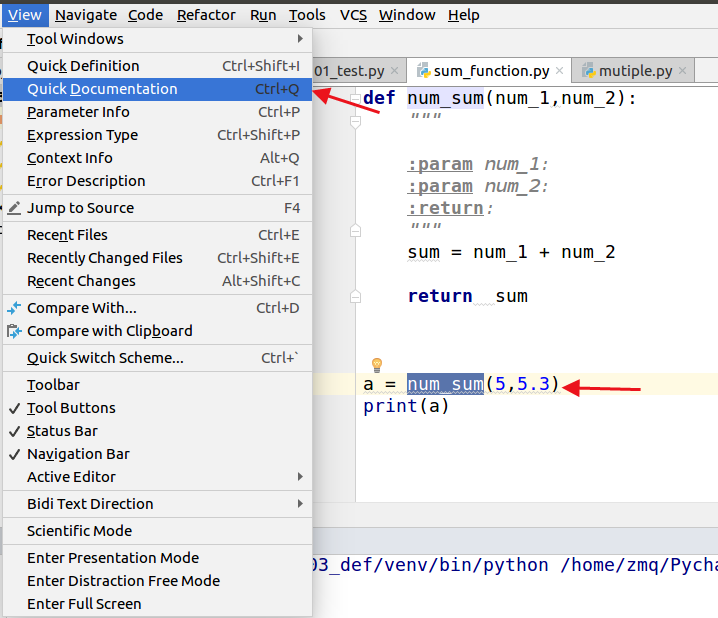
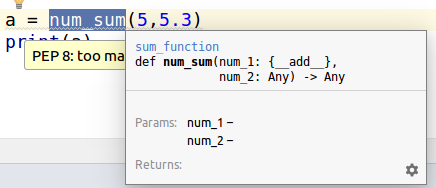
1. 转义字符
2. 函数的定义

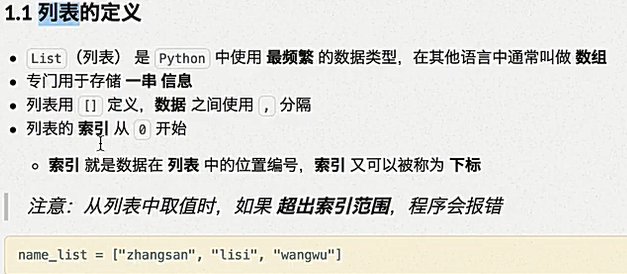


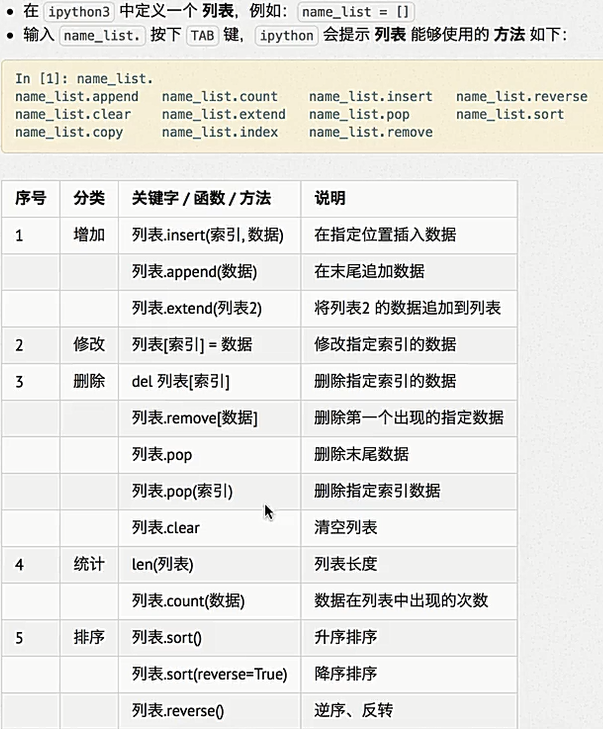
1. 函数注释方法

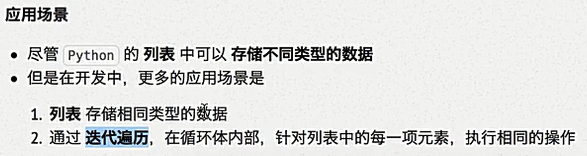
 查看注释方法：ctrl + Q



1. 列表







1. 关键字



1. 元组与列表区别



1. 字典

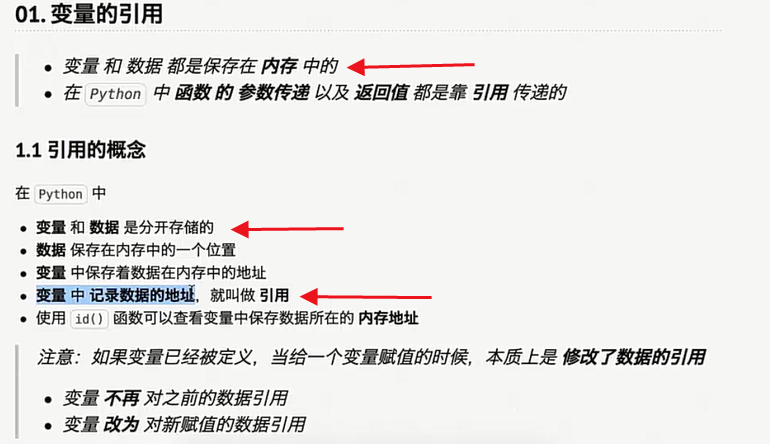


8、字符串函数

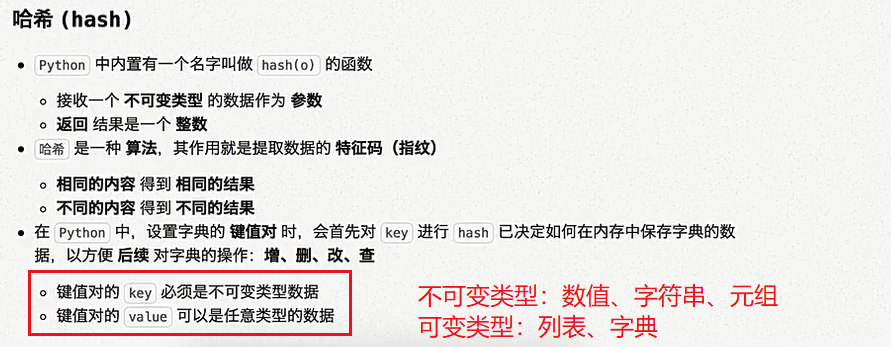


**9、不使用解释器，直接用终端运行.py文件（Shebang）**

10、引用的概念



11、hash函数



12、全局变量

（1）：在函数内部,无法直接修改全局变量的值。解释器：会在函数内部生成一个局部变量（名字与全局变量一致），而不是直接对全局变量进行修改

（2）：若要在函数内部修改全局变量的值，必须使用global



（3）全局变量命名规则



（4）Python基本程序结构图



13、在函数中，实现返回多个值：

使用元组的方式即可：如下

**#返回多个数据**

**def measure():**

**temp = 10**

**wet = 30**

**return temp,wet #result为元组**

**result = measure()**

**print(result)**

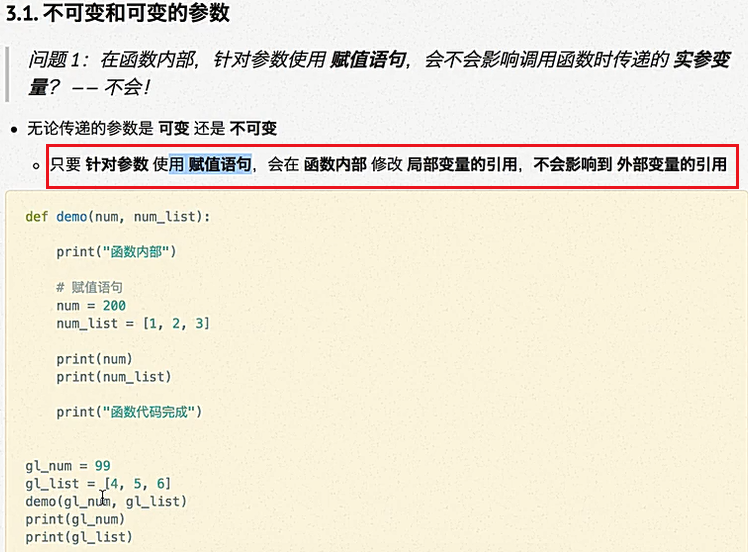
**#若想，用多个变量一次性接收函数的返回值，方法如下**

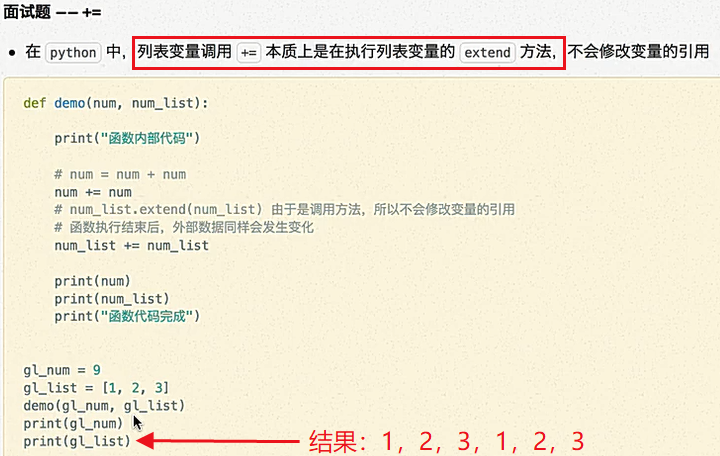
**#注意:变量的个数必须和返回元组的元素个数相同**

**gl\_temp,gl\_wet = measure()**

**print("T = %d , W = %d" % (gl\_temp,gl\_wet))**

14、函数：对形参进行赋值后，实参的变化





思考：

**# 分析：列表在函数之中赋值变化**

def test\_list(num):

# num += num

# num = num + num

num.remove(1)

print(num)

num\_list = [1,2,3]

test\_list(num\_list)

print(num\_list)

**# 分析：字典在函数之中赋值变化**

def test\_dict(num):

num\_1 = num

num\_1.pop("name")

print(num\_1)

print(num)

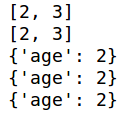
num\_dict = {"name":1,

"age":2

}

test\_dict(num\_dict)

print(num\_dict)

答案：

15、缺省参数的使用：  
示例代码：

**#对于缺省参数的使用**

def gender\_choice(name,age=20,gender=True):

if not gender:

gender = "felman"

else:

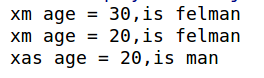
gender = "man"

print("%s age = %d,is %s" % (name,age,gender))

gender\_choice("xm",30,**gender=False**)

gender\_choice("xm",**gender=False**)

gender\_choice("xas")

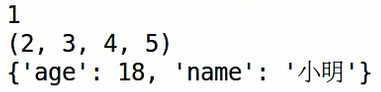
答案：

缺省参数使用的注意事项：

（1）：带有默认值的缺省参数一定要放在参数列表的末位置

（2）：调用有多个缺省参数函数时，必须制定参数名，使得解释器可以正确找到对应的关系

16、多值参数：

答案：

17、递归的应用：

**#定义一个递归函数，用于求解数字和**

def recursion(num):

#1：定义出口

if num == 1:

return 1

#2：求解1+2+3......+num的和

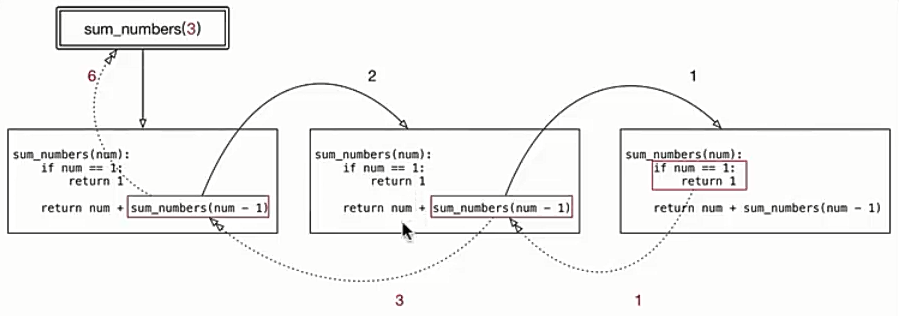
temp = recursion(num-1)

#返回 两个数字的和

return num + temp

result = recursion(100)

print(result)

递归调用的示意图：