**一、python变量和数据类型**

**01 python中的数据类型**

①**整数**：十六进制表示整数，用**0x**前缀和0-9，a-f表示，

例如：0xff00，0xa5b4c3d2

②**浮点数**

③**字符串**：用**‘’**/**“”**括起来的任意文本

④**布尔值**：**True/False**（注意大小写）

⑤**空值**：**None**

**02 print**

print 'hello,','python'

print("hello, python")

**03 pyhton的注释**

Python的注释以 **#** 开头，后面的文字直到行尾都算注释

# print 'hello'

**04 变量**

在Python程序中，变量是用一个变量名表示，变量名必须是大小写英文、数字和下划线（\_）的组合，且不能用数字开头

**05 定义字符串**

①如果字符串中包含 ‘ ’，及用“ ”括起来。

例子："I'm OK"

②如果字符串中包含“ ”，及用‘ ’括起来。

例子：'Learn "Python" in imooc'

③如果字符串中同时包含‘ ’和“ ”，使用转义字符**\**。

例子：'Bob said \"I\'m OK\".'

**06 raw字符串与多行字符串**

在字符串前面加 r ，字符串的字符就不需要转义了

多行字符串''' ... '''

**07 Unicode字符串**

ASCII编码：A 65 z122

**08 整数和浮点数**

整数和浮点数直接进行四则混合运算

整数+整数=整数

浮点数+浮点数=浮点数

整数+浮点数=浮点数

例子：

11/4=2

11%4=3

11.0/4=2.75

**09 布尔类型**

0、空字符串、None看成是False，其他数值和非空字符串堪称True

a and b , a: true结果:false 返回:a

a:false结果取决与b 返回:b

a or b , a: true结果:true 返回:a

a:false结果取决与b 返回:b

**二、List和Tuple类型**

**01 python创建list**

例子：

L = ['Adam', 95.5, 'Lisa', 85, 'Bart', 59]

print L

//['Adam', 95.5, 'Lisa', 85, 'Bart', 59]

**02 python按照索引访问list**

例子：

L = [95.5,85,59]

print L[0] //95.5

print L[1] //85

print L[2] //59

print L[3] //不存在会报错

**02 python倒序访问list**

例子：

L = [95.5, 85, 59]

print L[-1]

print L[-2]

print L[-3]

**02 python添加新元素**

①append() 方法，把新同学追加到 list 的末尾；L.append('Paul')

②insert()方法，它接受两个参数，第一个参数是索引号，第二个参数是待添加的新元素；L.insert(0, 'Paul')

**02 python从list删除元素**

pop()方法，删除最后一个元素并且返回该元素/ pop(2)删除对应索引的元素

**02 python替换元素**

例子：

L = ['Adam', 'Lisa', 'Bart']

L[0]='Bart'

L[2]='Adam'

print L

//['Bart', 'Lisa', 'Adam']

**02 python创建tuple**

①创建tuple和创建list唯一不同之处是用( )替代了[ ]

②tuple没有 append()方法，也没有insert()和pop()方法。所以，新同学没法直接往 tuple 中添加，老同学想退出 tuple 也不行。

③获取 tuple 元素的方式和 list 是一模一样的，我们可以正常使用 t[0]，t[-1]等索引方式访问元素，但是不能赋值成别的元素

**02 python创建单元素tuple**

0元素的tuple： t=() //()

t=(1) // 1

t=(1,) //(1,)

**02 python“可变”的tuple**

例子：

t = ('a', 'b', ['A', 'B'])

L=t(2) //['A', 'B']

L[0]= 'X'

L[1]= 'Y'

print(t)// ('a', 'b', ['X, 'Y'])

修改后不可变：

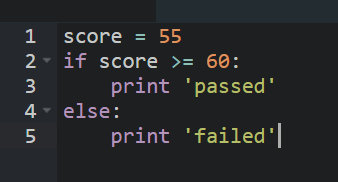
t = ('a', 'b', ('A', 'B'))

**三、条件判断和循环**

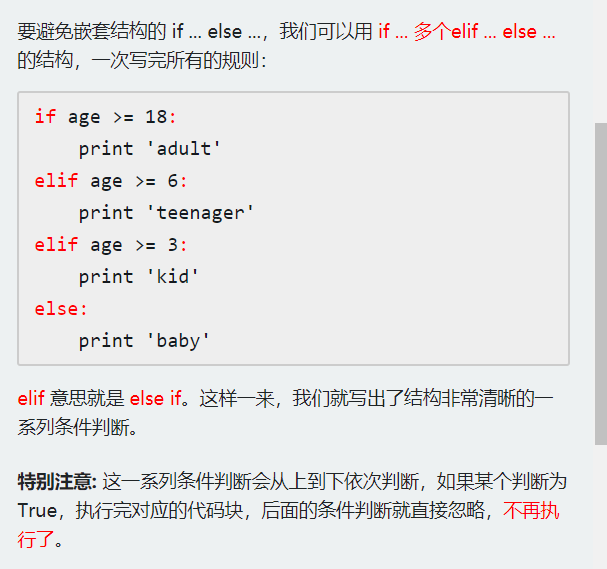
**01 python之if语句**



**02 python之if-else**

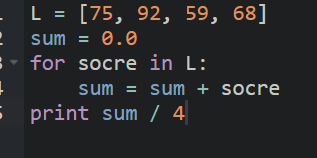


**03 python 之if-elif-else**



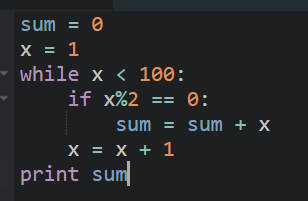
**04 python 之for循环**

例子：计算平均值



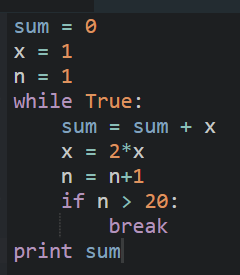
**05 python之while循环**

例子：求100以内奇数的和



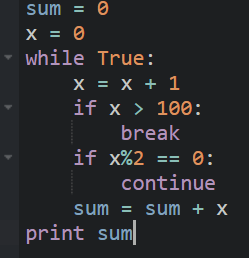
**06 python之break退出循环**

例子：计算计算 1 + 2 + 4 + 8 + 16 + ... 的前20项的和。



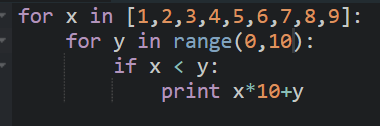
**07 python之continue继续循环**

例子：对0-100的while循环进行改造，增加continue计算奇数的和



**08 python 之多重循环**

例子：对100以内的两位数，请使用一个两重循环打印出所有十位数数字比个位数数字小的数，例如，23（2 < 3）

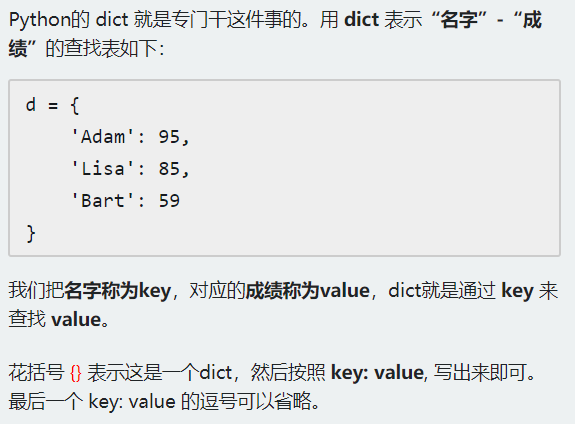


**四、Dict和Set类型**

**01 python什么是dict**

list和tuple可以表示顺序集合

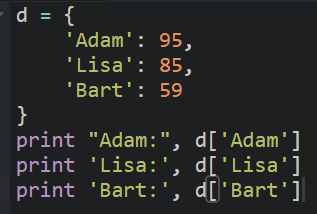
dict ：映射/类似json



由于dict也是集合，**len()** 函数可以计算任意集合的大小

**02 python访问dict**

例子：使用d[key]访问dict



Adam: 95

Lisa: 85

Bart: 59

**03 python 中dict的特点**

①**dict查到速度快**/list的查找速度随着元素的增加而下降

占用内存大，浪费很多内容/list占用内存小，但查找速度慢

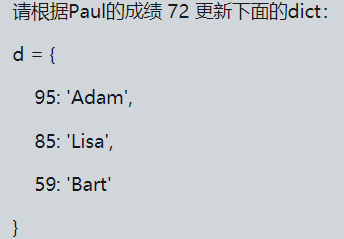
dict是按key查找，所以key唯一不重复

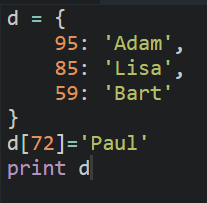
②**dict存储key-value序对没有顺序** 和list不一样

③**dict作为key的元素必须不可变**

**04 python更新dict**

例子：

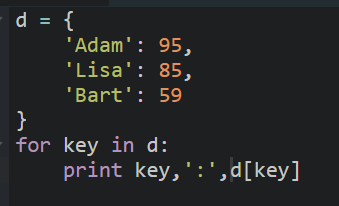




结果：{72: 'Paul', 59: 'Bart', 85: 'Lisa', 95: 'Adam'}

**05 python遍历dict**

例子：请用 for 循环遍历如下的dict，打印出 name: score



// Lisa : 85

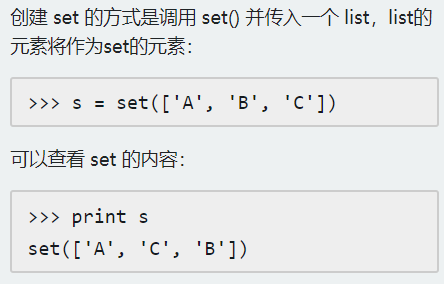
Adam : 95

Bart : 59

**06 python什么是set**

①dict的作用是建立一组 key 和一组 value 的映射关系，dict的key是不能重复的。

②set 持有一系列元素，这一点和 list 很像，但是set的元素没有重复，而且是无序的，这点和 dict 的 key很像。



③set是无序的、不能包含重复元素，会自动去掉重复元素

例子：

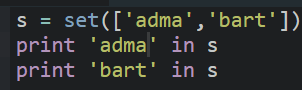


**07 python访问set**

set存储时无序集合，不能通过索引来访问

可用 in 操作符来判断

例子：



//True

True

**08 python set的特点**

①内部结构和dict很像，唯一区别是不存储value

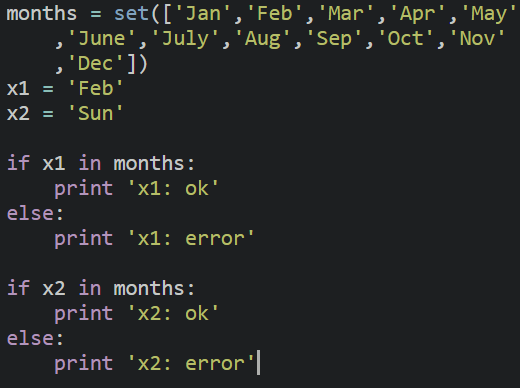
②set存储的元素和dict的key类似，是不可变对象

③set存储的元素也是没有顺序的

例子：

月份也可以用set表示，请设计一个set并判断用户输入的月份是否有效。

月份可以用字符串'Jan', 'Feb', ...表示。



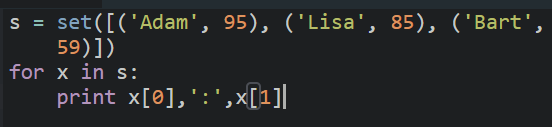
// x1: ok

x2: error

**09 python遍历set**

for循环实现

例子：请用 for 循环遍历如下的set，打印出 name: score 来



// Lisa : 85

Adam : 95

Bart : 59

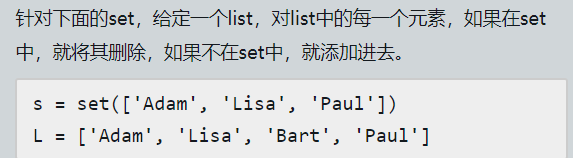
说明：输出的x是一个tuple

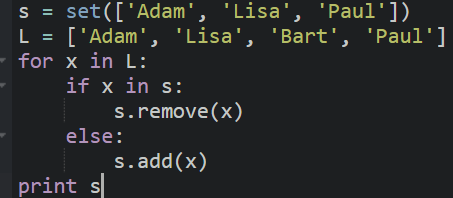
**010 python更新set**

一是把新的元素添加到set中，然后把已有的元素从set删除

添加元素：add（） 删除元素：remove（）

例子：



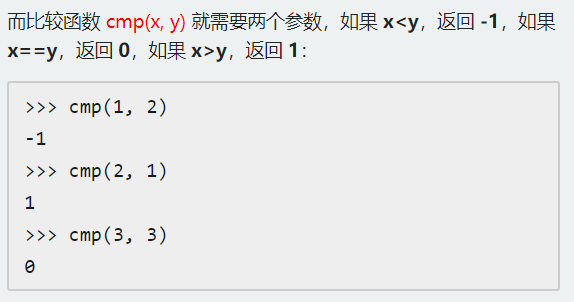


// set(['Bart'])

**五、函数**

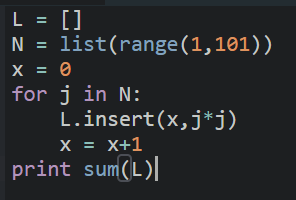
**01 python之什么是函数**

**02 python调用函数**





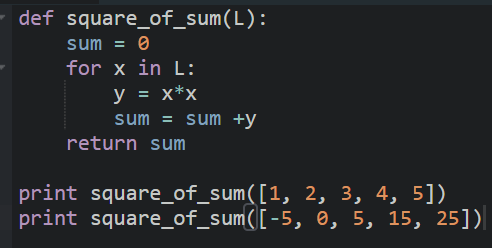
例子：sum()函数接受一个list作为参数，并返回list所有元素之和。请计算 1\*1 + 2\*2 + 3\*3 + ... + 100\*100。



//338350

**03 python编写函数**

例子：请定义一个 square\_of\_sum 函数，它接受一个list，返回list中每个元素平方的和。



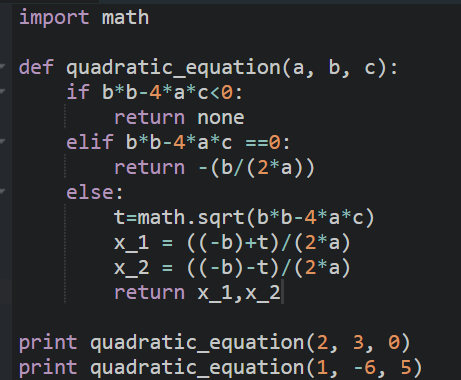
//55 900

**04 python函数返回多值**

例子：一元二次方程的定义是：ax² + bx + c = 0

请编写一个函数，返回一元二次方程的两个解。

注意：Python的math包提供了sqrt()函数用于计算平方根。

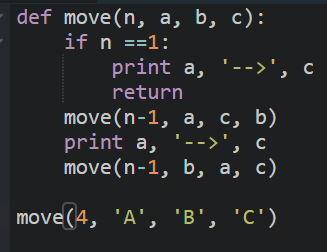


//(0.0, -1.5) (5.0, 1.0)

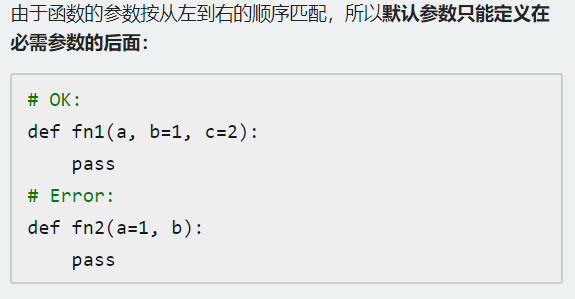
**05 python递归函数**

使用递归函数需要注意防止栈溢出。

例子：



**06 python定义默认参数**



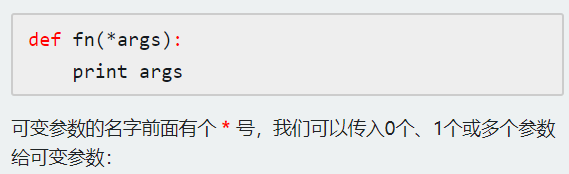
例子: 请定义一个 greet() 函数，它包含一个默认参数，如果没有传入，打印 'Hello, world.'，如果传入，打印 'Hello, xxx.'

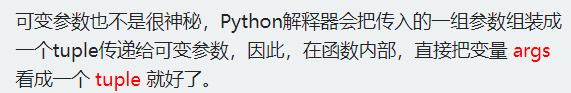


//hello,world.

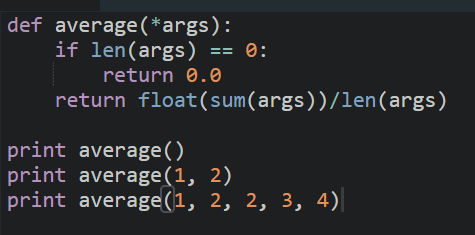
hello,Bart.

**07 python定义可变参数**





例子:接受可变参数求平均值



//0.0 1.5 2.4

**六、切片**

**01 对list进行切片**

①取前三个元素： L[0:3]

②如果第一个索引是0，则可省略： L[:3]

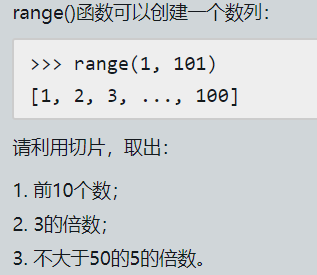
③从头到尾：L[:]

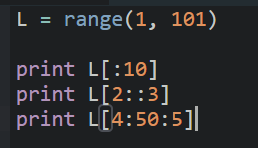
// 实际上是复制了一个新的list

④切片操作还可以指定第三个参数： L[::2]

//第三个参数表示每N个取一个，上面的 L[::2] 会每两个元素取出一个来，也就是隔一个取一个。

例子：





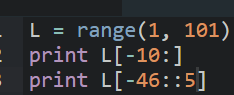
说明：print L[2::3] 从索引为2开始取到完，每隔三个去一次

print L[4:50:5] 从索引为4开始到50结束，每隔五个去一次

**02 倒序切片**

记住倒数第一个元素的索引是-1。倒序切片包含起始索引，不包含结束索引。

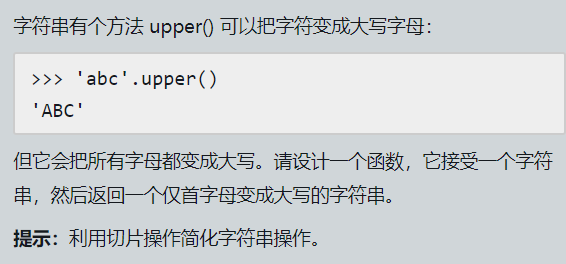
例子：利用倒序切片对1-100的数列取出：\*最后10个数\*最后10个5的倍数

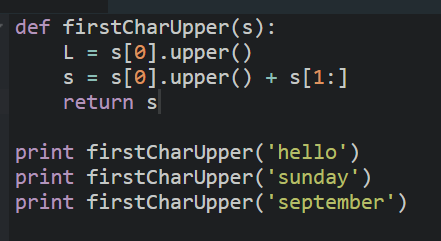


//55整数时54反着过来就是46

**03 对字符串切片**

例子：





//Hello Sunday September

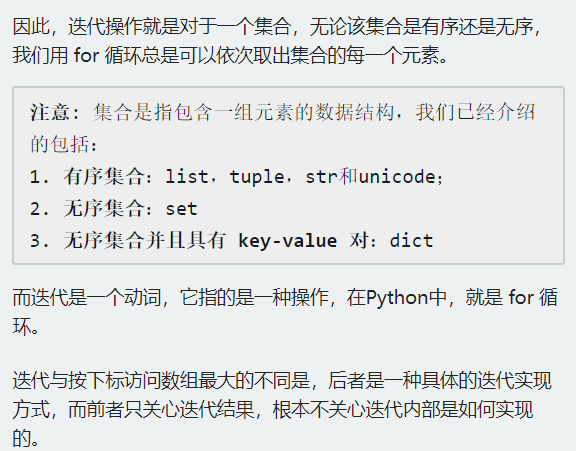
**七、迭代**

**01 迭代**

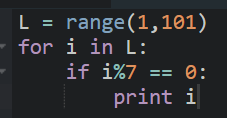
①在Python中，如果给定一个list或tuple，我们可以通过for循环来遍历这个list或tuple，这种遍历我们成为迭代（Iteration）。

②在Python中，迭代是通过 **for ... in** 来完成的

③因为 Python 的 for循环不仅可以用在list或tuple上，还可以作用在其他任何可迭代对象上。



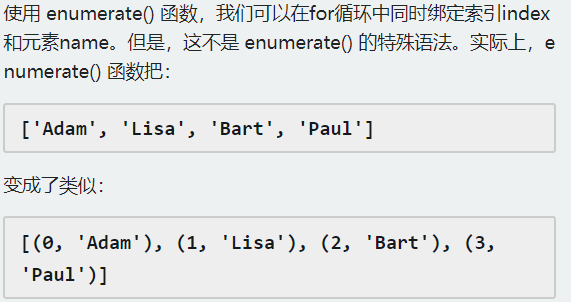
例子：请用for循环迭代数列 1-100 并打印出7的倍数。



**02 索引迭代**

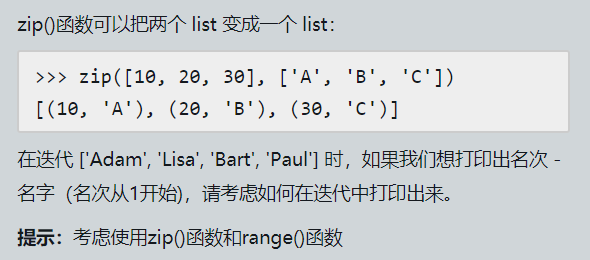
①Python中，迭代永远是取出元素本身，而非元素的索引。

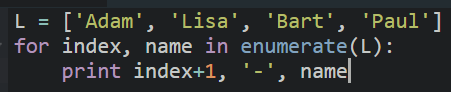
②方法是使用 **enumerate()** 函数，拿到索引值



③可见，索引迭代也不是真的按索引访问，而是由 enumerate() 函数自动把每个元素变成 (index, element) 这样的tuple，再迭代，就同时获得了索引和元素本身。

例子：





1 - Adam

2 - Lisa

3 - Bart

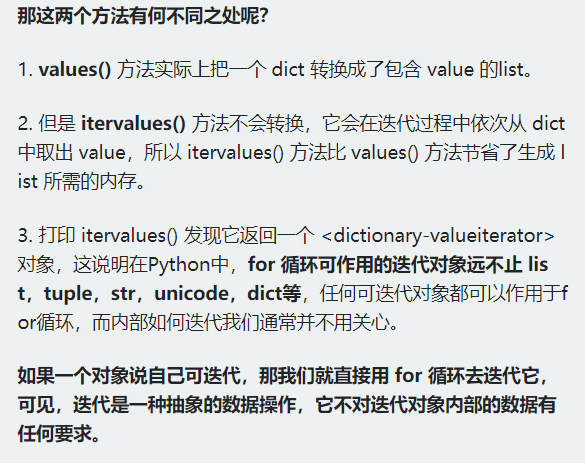
4 - Paul

**03 迭代dict的value**

①dict对象本身就是可迭代对象，用 for 循环直接迭代 dict，可以每次拿到dict的一个key。

②dict 对象有一个 **values()** 方法，这个方法把dict转换成一个包含所有value的list，这样，我们迭代的就是 dict的每一个 value

③ict除了values()方法外，还有一个 **itervalues()** 方法，用 itervalues() 方法替代 values() 方法，迭代效果完全一样

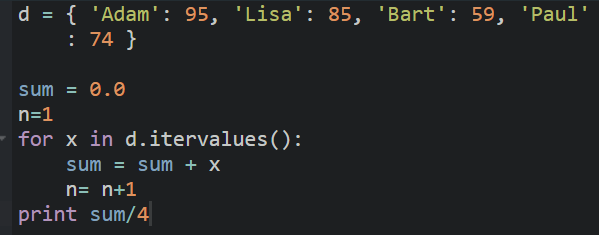


例子：

给定一个dict：

d = { 'Adam': 95, 'Lisa': 85, 'Bart': 59, 'Paul': 74 }

请计算所有同学的平均分

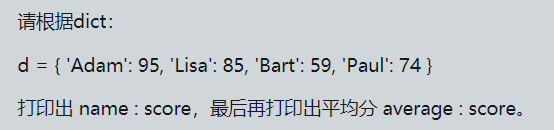


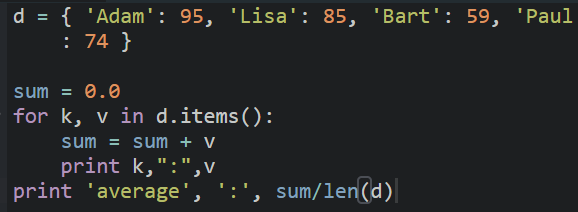
//78.25

**04 迭代dict的key和value**



例子：





//Lisa : 85

Paul : 74

Adam : 95

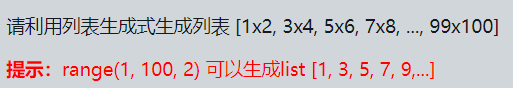
Bart : 59

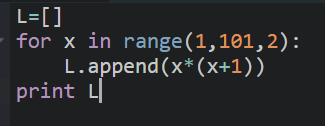
average : 78.25

**八、列表生成器**

**01 生成列表**

例子：

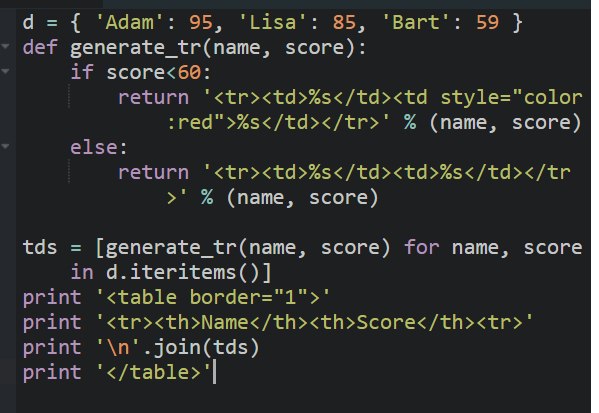




**02 复杂表达式**

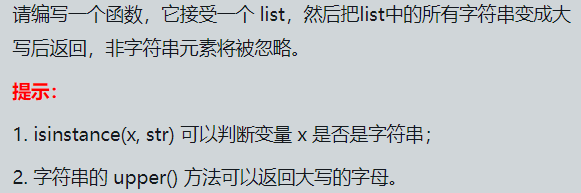
例子：

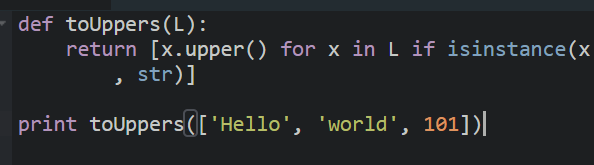




**03 条件过滤**

例子：





//['HELLO', 'WORLD']

**04 多层表达式**

例子：利用 3 层for循环的列表生成式，找出对称的 3 位数。例如，121 就是对称数，因为从右到左倒过来还是 121。

