**Day06笔记**

day06(上午)：

1.**全局变量&局部变量**：

作用范围：作用域

全局变量：

理解：

1).作用域比较大，定义在.py文件中，函数外

2).可以被文件中任何一个函数所使用

局部变量：

理解：

2).作用域比较小，函数的形参或者函数体中

3).只能被当前的这个函数所使用，出了此函数，外部够不着它

**在局部位置修改全局变量的值：**

**先使用global**关键字定位到需要修改的全局变量，然后在使用全局变量 = 值操作进行修改

代码如下：

**global a**

**a = 30**

**在内部函数中修改外部函数中局部变量的值：**

**先使用nonlocal**关键字定位到需要修改的局变变量，然后使用局部变量 = 值操作进行修改

代码如下：

**nonlocal b**

**b = 100**

2.外部函数&内部函数：

内部函数：定义在外部函数的内部，成为其一部分(认为是外部函数的成员)

a=10  *#a为全局变量*  
**def** func1():  
 **global** a  
 a=20  
 print(a)  
**def** func2():  
 b=100 *#b为局部变量*  
 print(a,b)  
func2()  *#得到 10 100*func1() *#得到20*

func2() *#得到20 100*  
print(a) *#得到10***以下代码有问题：因为b是属于func2()函数中的局部变量，出了func2()函数  
 外界都不能使用***# print(b)*

**【注意事项】：**

1).内部函数可以随意使用外部函数的内容，外部函数不能使用内部函数的内容 注意：和作用域有关

2).内部函数不能单独执行，需要依赖于其外部函数才能执行

**def** outer():  
 print(**"我是外部函数..."**)  
 b=20  
 **def** inner():  
 print(**"我是内部函数..."**)  
 **nonlocal** b  
 b=30  
 print(b)  
 inner()  
 print(b)

outer()

**以下代码有问题：对于内部函数的执行只能依赖于它的外部函数，不能出了外部函数在外界单独执行；***# inner()*

3.ascii码

字符编码(字符集)

字符数据 <--> ascii码表 <--> 二进制数据

只需要记住3个分界点：

**1).'0' --> 48**

**2).'A' --> 65**

**3).'a' --> 97**

内置函数：

**ord(str)：传入字符数据，返回整数值(ascii码值)**

**chr(num)：传入整数数据(ascii码值)，返回字符数据**

**演示ord()、chr()函数的使用**s1 = **'a'**print(ord(s1),type(ord(s1))) *#得到97 <class 'int'>*num = 65  
print(chr(num),type(chr(num))) *#得到A <class 'str'>*

**需求：从键盘读入一个英文字母，如果是小写输入其大写，如果是大写输出其小写**s = input(**'请输入一个英文字母：'**)

*#得到s字符数据所对应的ascii码值*num = ord(s)  
**if** num >= 65 **and** num <= 90:

*#如果代码能够执行到此处，说明键盘输入的是大写英文字母* print(**'大写字母：%s，转换为小写为：%s'** %(s,chr(num + 32)))  
**else**:  
 *# 如果代码能够执行到此处，说明键盘输入的是小写英文字母* print(**'小写字母：%s，转换为大写为：%s'** % (s, chr(num - 32)))

----------------------------------------------------------------------------------------

day06(下午)：

1.**列表生成式：**

作用：执行完毕之后会返回给程序一个**列表对象**，在不破坏原本复杂需求的同时；

更加便捷的得到一个列表对象(代码更加简短)；

它属于python语法糖的一种

什么是语法糖？

在编码的过程中，可能会去到业务逻辑相对比较复杂的情况；

此时代码的量会比较大，如果我们在适当的时机选择使用python的语法糖来实现；

那么此时程序的结构会变的更加的简洁、优雅(代码变少)，但是不会破坏原本的业务逻辑；

给我们的编程带来不一样的快感...

举例：**三元运算符**就是python语法糖之一

格式：

**[expr for ver1,[ver2],... in 序列对象(range、str、list、tuple、dict...) if ...]**

**#需求1：得到[1,2,3,...,10]列表对象** *#方式一：不使用列表生成式*print(list(range(1,11)))  
*#方式二：使用列表生成式*print([x **for** x **in** range(1,11)])  
**#需求2：得到 1\*1，2\*2，3\*3,...,10\*10的一个列表对象** *#方式一：不使用列表生成式*lt1 = []  
**for** i **in** range(1,11):  
 lt1.append(i \*\* 2)  
print(lt1)  
*#方式二：使用列表生成式*print([x \*\* 2 **for** x **in** range(1,11)])  
**#需求3：得到 1\*1，2\*2，3\*3,...,10\*10的一个列表对象，元素只接受偶数的乘积值***#方式一：不使用列表生成式*lt2 = []  
**for** i **in** range(1,11):  
 **if** i % 2 == 0:  
 lt2.append(i \*\* 2)  
print(lt2)  
*#方式二：使用列表生成式*print([x \*\* 2 **for** x **in** range(1,11) **if** x % 2 == 0])  
**#需求4：实现全排列  
str1 = 'ABC'  
str2 = 'XYZ'  
效果如下：['AX','AY','AZ','BX','BY',...,'CZ']**str1 = **'ABC'**str2 = **'XYZ'***#方式一：不使用列表生成式*lt3 = []  
**for** i **in** str1:  
 **for** j **in** str2:  
 lt3.append(i + j)  
print(lt3)  
*#方式二：使用列表生成式*print([x + y **for** x **in** str1 **for** y **in** str2])

**需求5：  
字典对象：{'aa':'AA','bb':'BB','cc':'CC'}  
  
得到字典对象的键和值，然后组合成为如下数据存入到列表中返回：  
['aa=AA','bb=BB','cc=CC']  
'''**dic1 = {**'aa'**:**'AA'**,**'bb'**:**'BB'**,**'cc'**:**'CC'**}  
*#方式一：不使用列表生成式*lt4 = []  
  
**for** k,v **in** dic1.items():  
 lt4.append(k + **'='** + v)  
  
print(lt4)  
*#方式二：使用列表生成式*print([k + **'='** + v **for** k,v **in** dic1.items()]) **需求6：将列表中的每个字符串元素都转换为全小写返回  
lt = ['PYTHON','Java','PHP','IOS']  
得到：['python','java','php','ios']**lt = [**'PYTHON'**,**'Java'**,**'PHP'**,**'IOS'**]  
*#方式一：不使用列表生成式*lt5 = []  
**for** i **in** lt:  
 lt5.append(i.lower())  
print(lt5)  
*#方式二：使用列表生成式*print([x.lower() **for** x **in** lt]) **需求7：将列表中的每个字符串元素都转换为全小写返回  
lt = ['PYTHON','Java','PHP','IOS',200]  
得到：['python','java','php','ios']**lt = [**'PYTHON'**,**'Java'**,**'PHP'**,**'IOS'**,200] **以下代码报错：类型 AttributeError**

*# print([x.lower() for x in lt])* **lower()函数是属于字符串类型独有的函数，而lt中200是int类型的数据，  
一旦尝试调用lower()就会报错(非法的，不允许的)；我们怎么样优化解决此问题？  
内置函数：isinstance(obj,type)  
作用：判断obj对象是否属于type类型，如果类型匹配，则返回True，反之，返回False**print([x.lower() **for** x **in** lt **if** isinstance(x,str)])

2.**常用的内置函数：**

**round()：abs()：max()：min()：sum()：hex()：oct()：bin()：pow(x,y)：divmod(x,y)：**

ord()：chr()：type()：id()：len()：

**演示常用内置函数的使用：****round(num1,num2)：四舍五入**print(round(3498.1415926))  
print(round(3498.1415926,3),round(3498.1415926,-2))  
**abs(num)：绝对值 特点：不管正数还是负数的绝对值都是其正数的形式，0还是0**print(abs(-1),abs(-2),abs(0),abs(1),abs(2))  
**max(seq)、min(seq)：得到seq对象的最大、最小值**lt = [1,2,3,4,5]  
print(max(lt)) *#得到5*  
print(max(tuple(lt))) *#得到5*  
print(max(**"abcdefgABCDE1234567"**)) *#得到g*  
print(min(range(10))) *#得到0*  
print(min({10,30,-20,-90,100})) *#得到-90*  
**sum(seq)：得到seq中所有元素累加以后的和值返回**print(sum({10,20,30,40,50,10,20})) *#得到150* **【注意事项】：  
1).对于str类型的数据不能使用sum()求和，没有意义  
2).对于sum()函数只能接受容器类型的数据，不能接受多个单值数据**

**例如以下两个函数就是非法的***# print(sum('12345'))  
#print(sum(1,2,3,4,5))* **hex(num)：将十进制数据num转换为十六进制返回(数据类型：str)  
oct(num)：将十进制数据num转换为八进制返回(数据类型：str)  
bin(num)：将十进制数据num转换为二进制返回(数据类型：str)**print(hex(100)) *#得到0x64*  
print(oct(100)) *#得到0o144*  
print(bin(100)) *#得到0b1100100* **pow(x,y)：计算x的y次方返回**

print(pow(3,3)) *#得到27*  
print(pow(2,5)) *#得到32*  
print(pow(5,-1)) *#得到0.2*  
print(pow(5,-2)) *#得到0.04* **divmod(x,y)：返回一个元祖对象，有两个元素；  
 第一个元素记录了x除以y后的商  
 第二个元素记录了x除以y后的余数**  
print(divmod(9,2)) *#得到(4, 1)*   
print(divmod(10,2)) *#得到(5, 0)*

1. random模块中常用的函数： **演示random模块中常用的函数：**

**import** random **# random()：返回一个[0,1)之间的随机浮点值(双精度)**print(random.random())  
**# uniform(a,b)：返回[a,b]之间的一个随机浮点值 a,b两个变量大小随意 --> 1).(5,10) 2).(10,5)**print(random.uniform(10,20))  
print(random.uniform(30,20))  
**# randrange([start],end,[step])：返回[start,end)的范围之间的一个随机整数，考虑步长**print(random.randrange(0,10,2))**# randint(a,b)：返回[a,b]之间的一个随机整数**print(random.randint(1,3))  
**# choice(seq)：随机获取seq中的某个元素返回**print(random.choice(**'abcdefg'**))  
print(random.choice(**'我是一个中国人'**))  
print(random.choice(range(10)))  
**# sample(seq,num)：从seq对象中随机获取num个元素，以列表的形式返回**print(random.sample((**'a'**,**'b'**,**'c'**,**'d'**,**'e'**,**'f'**),3))  
**# shuffle(lt)：将list对象lt中的元素打乱 【注意】只能接受list对象**lt = [1,2,3,4,5,6,7]  
random.shuffle(lt)  
print(lt)

4.math模块中常用的属性和函数：

属性：

pi：圆周率

e： 自然数

函数：

ceil(num)：向上取整

floor(num)：向下取整

sqrt(num)：开平方根

radians(obj)：将角度转为弧度

degrees(obj)：将弧度转为角度

扩展：

sys模块中的argv属性的使用：

将当前正在执行的.py文件的绝对路径存入到列表中返回

os模块中的system()函数的使用：

在system函数的形参位置可以接受一些dos指令(str类型)来执行；

**演示math模块中常用的属性和函数：**

**import** math **#ceil()、floor()：**print(math.ceil(3.14),math.floor(4.99)) *#得到4 4*  
print(math.ceil(-3.14))  *#得到-3*  
print(math.floor(-3.14)) *#得到-4***#sqrt()：**print(math.sqrt(4)) *#得到2*  
print(math.sqrt(9)) *#得到3*  
print(math.sqrt(25)) *#得到5*  
print(math.sqrt(2)) *#得到1.41421356237*  
**#radians()、degrees()：**print(math.radians(180)) *#3.141592653589793*print(math.radians(360)) *#6.283185307179586*print(math.degrees(3.141592653589793)) *#得到180.*0 **扩展：  
sys模块中的argv属性的使用：  
将当前正在执行的.py文件的绝对路径存入到列表中返回  
os模块中的system()函数的使用：  
在system函数的形参位置可以接受一些dos指令(str类型)来执行；  
import** sys,os  
print(sys.argv)  
os.system(**'ipconfig'**)  
**os.system('dir')**os.system(**'cls'**)

5字符串中常用的函数：

join(seq,format)：将seq容器对象中的元素以format格式进行连接，返回一个字符串数据

lt = [**'i'**,**'love'**,**'you'**,**'very'**,**'much'**]  
print(**'-'**.join(lt))  
print(str.join(**'#'**,lt))  
print(**' '**.join(lt),type(**' '**.join(lt)))  
print(lt)

ljust(width,format)：将str数据显示在左侧，如果长度/尺寸不满width，那么右侧补format

rjust(width,format)：将str数据显示在右侧，如果长度/尺寸不满width，那么左侧补format

center(width,format)：将str数据显示在中间，如果长度/尺寸不满width，那么左右侧补format

zfill(width)：将str数据显示在右侧，如果长度/尺寸不满width，那么左侧补0

str1 = **'i love you very much'**print(str1.ljust(50,**'\*'**))

*#i love you very much\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**print(str1.rjust(50,**'-'**)) *#------------------------------i love you very much*print(str.rjust(str1,50,**'$'**)) *#$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$i love you very much*print(str1.center(50,**'^'**))

*#^^^^^^^^^^^^^^^i love you very much^^^^^^^^^^^^^^^*print(str1.zfill(50)) *#000000000000000000000000000000i love you very much*

**strip()： 去除字符串左、右两侧匹配到的字符数据，返回一个新串**

**lstrip()：去除字符串左侧匹配到的字符数据，返回一个新串**

**rstrip()：去除字符串右侧匹配到的字符数据，返回一个新串**

str1 = **' ,,,..\ i hate you very much ...,,;;\ '**print(str1.lstrip(**'\ ,.'**))

*#i hate you very much ...,,;;\*print(str1.rstrip(**'\;., '**))

*#,,,..\ i hate you very much*print(str1.strip(**',\;. '**))  
*#i hate you very much*

replace(old,new,[number])：将str中的old内容替换为new，默认全部替换；也可以选择性的替换number次  
str1 = **"千锋啊千锋，你是我的骄傲！千锋啊千锋，我为你自豪！"**print(str1.replace(**'千锋'**,**'老郭'**))

*#老郭啊老郭，你是我的骄傲！老郭啊老郭，我为你自豪！*print(str1.replace(**'千锋'**,**'老郭'**,2))

*#老郭啊老郭，你是我的骄傲！千锋啊千锋，我为你自豪！*  
print(str1)

*#千锋啊千锋，你是我的骄傲！千锋啊千锋，我为你自豪！*

split(format)：将str以format格式进行切割，返回一个列表对象；默认空参数的split()可以切割的内容：' '、\n、\t

splitlines()： 将str以\n进行切割，返回一个列表对象；

str1 = **'i love\nabc you\t very\n much'**print(str1.split(**' '**))

*#['i', 'love\nabc', 'you\t', 'very\n', 'much']*  
print(str1.split())

*#['i', 'love', 'abc', 'you', 'very', 'much']*  
print(str1.split(**'\n'**))

*#['i love', 'abc you\t very', ' much']*  
print(str1.splitlines())

*#['i love', 'abc you\t very', ' much']*

index(s)：返回字符串s在原串中对应的下标/索引，如果存在多个相同的数据,返回第一个匹配成功的，如果都没有匹配成功，会报错

count(s)：返回字符串s在原串中出现的次数，如果一次都没有出现，不会报错，返回0

str1 = **'abcdefgd1234567'**print(str1.index(**'d'**)) *#3*  
print(str1.index(**'efg'**))  *#4*  
*print(str1.index('xyz')) #报错*str1 = **'dsaalkfaaasdfaaaa2141aaaaaaxyz'**print(str1.count(**'a'**)) *#15*  
print(str1.count(**'aa'**)) *#7*  
*#如果一次都没有，返回0，不报错*print(str1.count(**'www'**)) *#0*

find(format)：查找字符串中首次出现的format的位置

rfind(format)：查找字符串中最后一次出现的format的位置

str1 = **'山不在在高，有仙则灵；水不在在深，有龙则灵；斯是在在陋室，惟吾德馨'**print(str1.find(**'在在'**)) *#2*  
print(str1.rfind(**'在在'**)) *#24*

upper()：将字符串数据变为全大写

lower()：将字符串数据变为全小写

capitalize()：将字符串数据首字母大写，其余字母小写

title()：将字符串数据首字母大写，其余字母小写 备注：如果存在多个单词，每个单词首字母都要考虑

swapcase()：将字符串中大写变为小写，小写变为大写

str1 = **'today is thursday PM'**print(str1.upper()) *#TODAY IS THURSDAY PM*  
print(str1.lower()) *#today is thursday pm*  
print(str1.capitalize()) *#Today is thursday pm*  
print(str1.title()) *#Today Is Thursday Pm*  
print(str1.swapcase()) *#TODAY IS THURSDAY pm*

**以下两个函数结果都是布尔值：**

startswith(format)：判断字符串是否以format开头

endswith()： 判断字符串是否以format结尾

str1 = **'hahaha abcdefghijk 32141234 hehehe'**print(str1.th(**'h'**)) *#True*print(str1.startswith(**'ha'**)) *#True*print(str1.startswith(**'haha'**)) *#True*print(str1.startswith(**'Haha'**)) *#False*print(str1.startswith(**'hahha'**)) *#False*print(str1.endswith(**'e'**)) *#True*print(str1.endswith(**'he'**)) *#True*print(str1.endswith(**'ehe'**)) *#True*print(str1.endswith(**'Hehe'**)) *#False*

**以下函数的执行结果都是布尔值**

isalnum()：判断字符串的内容是否都是英文、数字字符；如果是，返回True；反之，返回False

isalpha(): 判断字符串的内容是否都是英文字符；如果是，返回True；反之，返回False

isdecimal()：判断字符串的内容是否都是数字字符；如果是，返回True；反之，返回False

isdigit()： 判断字符串的内容是否都是数字字符；如果是，返回True；反之，返回False

islower()： 判断字符串中英文字符是否全小写；如果是，返回True；反之，返回False 【注意】数字、其它字符不影响

isupper()： 判断字符串中英文字符是否全大写；如果是，返回True；反之，返回False 【注意】数字、其它字符不影响

istitle()： 判断字符串内容中的英文子串(多组)的首字母是否大写

str1 = **'3214Love&Tdsajflks\*^'**print(str1.isalnum()) *#False*  
print(str1.isalpha()) *#False*  
print(str1.isdigit()) *#False*  
print(str1.isdecimal()) *#False*  
print(str1.islower()) *#False*  
print(str1.isupper()) *#False*  
print(str1.istitle()) *#Ture*