字符串

字符串定义使用单引号或双引号均可

.title() 首字母大写

.upper() 全部大写

.lower() 全部小写

.strip() 去掉两端空格

.lstrip()去掉左边空格

.rstrip()去掉右边空格

制表符 \t , 换行符 \n

整数/浮点数

+-\*/ ： 加减乘除运算符

\*\* ： 乘方运算，例如3\*\*4 ，就是3的4次方运算

ps，

1. 在字符串中使用整数/浮点数时，需要显式地调用str() 方法将整数/浮点数用作字符串，如

age = 5

print("I am "+str(age) +" years old.")

2. 在python2中，整数除法的结果只包含整数部分，小数部分会被删除。如要保留小数部分，务必确保至少有一个操作数为浮点数，这样结果也会是浮点数

如，3/2 结果是1 ， 3.0/2 或 3/2.0 结果都是1.5

注释

在python中，使用井号#标识，井号后面的内容会被python解释器忽略。

列表

定义：使用中括号[]表示列表，并使用逗号来分隔其中的元素

长度：len(xxx), 返回列表长度

索引：索引从0开始，例如xxx[2]就是访问第三个元素

索引如果为负数，就是指倒数第几个元素，例如xxx[-2]就是倒数第二个元素

bicycles=['trek','cannondale','redline','specialized']

bicycles[0] 就是第一个元素trek

bicycles[-2] 就是倒数第二个元素redline

修改元素：通过索引去指定并修改元素, eg, array[2]='abc'

添加元素：

.append(value): 在列表末尾添加元素

.insert(index,value): 在指定索引出添加元素

删除元素：

del array[index] : 删除指定索引位置的元素,eg, del array[1], 删除array中的第二个元素

.pop(): 删除末尾元素，并返回该元素的值

.pop(index): 删除指定索引位置的元素，并返回该元素的值

.remove(value): 删除第一个指定值的元素, eg array.remove('abc') , 在数组array中删除第一个值为abc的元素。

排序：xxx.sort() :对列表进行永久性排序（默认是升序排序），会改变原来列表里面元素顺序

xxx.sort(reverse=True) : 对列表进行永久性“倒序”排序

sorted(xxx): 对列表进行临时性排序，不会改变原来列表里面元素排序

sorted(xxx,reverse=True):对列表进行临时性倒序排序

xxx.reverse(): 对列表进行反转排序，会改变原来列表里面的元素排序

计算：

min(xxx): 返回最小值

max(xxx)：返回最大值

sum(xxx)：返回所有元素的总和

列表切片：使用中括号[x:y]方法截取，包含x元素，但不包含y元素

array[x:y], 从x索引元素开始，到y-1索引元素结束

array[:y], 从第一个元素开始，到y-1索引元素结束

array[x:], 从x索引元素开始，到结尾最后一个元素结束

array[:], 返回整个列表，这也是复制列表副本的方法

array[-x:-y], 从倒数第x个元素开始，到倒数第(y+1)个元素结束

array[:-y], 从第一个元素开始，到倒数第(y+1)个元素结束

array[-x:], 从倒数第x个元素开始，到结尾最后一个元素结束

复制列表: array[:], 返回一个新的与原来列表一样的列表，新旧列表不相互影响，独立的两个列表

注意：

newArray=array[:] , newArray 与array是两个独立的，互不影响的两个列表

array2=array , array2 与array 是指着同一个列表

元组 ： 就是不可改变的列表

元组使用圆括号()去定义，列表使用中括号[]去定义，

dimension=("abc","cde")

list=["abc","cde"]

元组不能通过索引去改变某个元素赋值，但可以重新给整个元组赋值

dimension[2]=xxx --> 报错，不允许

dimension=("zzz","yyy") --> 允许

循环

格式：for 循环第一行以for 开头，in一个list，冒号结果，循环体为for之后每行行头缩进了的内容， 行头没有缩进代表循环已借宿。

for xxx in xxx\_list :

....

....

数值列表

range(a,b) : 生成一系列数字，从a开始，到b-1结束，不包含b，默认步长为1

range(a,b,c) : 生成一系列数字，从a开始，以c为步长，到b-1结束，不包含b

list(range(xxx)) : 将一系列数字转换为列表，如list(range(1,20,2))

例子

squars=[]

for number in range(1,11):

squars.append(number \*\* 2)

print(squars)

//上面例子为计算1～10的平方数，并放到squars列表中，头三行代码可以用一行“列表解析”代替

squars=[number\*\*2 for number in range(1,11)]

列表解析的格式是

列表名=[expression for iter\_val in iterable]

if 条件判断

可使用 == != < <= > >= 等作判断符号，也可以使用布尔表达式True/False

多个条件判断时可以使用and 或 or 去连接

1. if 语句

if conditional\_test:

do something

2. if-else 语句

if conditional\_test:

do something

else:

do others

3. if-elif 语句 (可以有多个elif判断语句,通过了条件测试后，python会跳过余下的条件测试)

if conditional\_test\_1:

do something

elif conditional\_test\_2:

do something\_2

4. if-elif-else 语句(中间可以有多个elif判断语句)

if conditional\_test\_1:

do something

elif conditional\_test\_2:

do something\_2

else:

do others

## 字典 Dictionary

定义：字典用放在大括号｛｝中的一系列键值对表示，键和值之间使用冒号分隔，每个键值对之间使用逗号分隔

Eg: alien={‘color’:’green’, ‘point’: 5}

获取与键相关的值，可以指定字典名和放在方括号内的键

alien[‘color’]

添加键值对，字典名[‘新的键‘’] = 新的值

alien[‘speed’]=’fast’

删除键值对， del 字典名[‘键名’]

del alien[‘speed’]

遍历字典中所有键值对(遍历字典是，键值对的返回顺序与存储顺序可能不一样，如有需要可以使用sorted() 函数)

for key, value in 字典名.items() :

xxxxx

遍历字典中所有的键( 遍历字典时，会默认遍历所有的键)

for key in 字典名.keys():

xxxxx

相当于

for key in 字典名

xxxxx

遍历字典中的所有值（对于包含重复元素的列表可以通过set() 函数，返回一个元素不重复的列表）

for value in 字典名.value():

xxxx