### 变量和数据类型

* 1. 布尔类型

and 与运算

or 或运算

not 非运算

0 ，空字符串‘’ ，None 为False

其他数值、非空字符都是True

|  |
| --- |
| a=’t’  b=’e’  #若a为True，与运算结果取决于b,则结果将返回b的值；若a为False，与运算结果一定为False，短路计算，不考虑b了，所以返回a的值  print(a and b) # 输出e  print(a or b) # 输出t  a=’’  print a and b #输出 e |

* 1. print

print 可以打印多个变量，用逗号分开，一个逗号显示一个空格

### List和Tuple类型

list为有序集合。类似于php数组

list=[‘a’,’b’]

* 1. list中添加新元素

list.append(‘ele’) 在list末尾添加ele元素

list.insert(index,’ele’) 把新元素ele添加到list中索引时index的位置。

insert为前插，即插入到原来index那个元素的前面，所以list.insert(-1,’ele’),会插入到倒数第二个位置。

* 1. list中删除元素

list.pop() 删除list最后一个元素，并返回这个元素

list.pop(index) 删除index索引位置的元素，并返回这个元素

* 1. 创建tuple

tuple也是有序列表，中文翻译为‘元祖’，一旦创建不能修改。

t = ('Adam', 'Lisa', 'Bart')

tuple也可以用t[2]这样访问，就是不能修改。

创建单元素tuple时，元素后要多加个“，”

t=(1,) 跟运算符（）以示区分

tuple的不可修改，是指向不变，但是指向的内容可以更改。

|  |
| --- |
| t = ('a', 'b', ['A', 'B'])  L[0] = 'X'  L[1] = 'Y'  print t  ('a', 'b', ['X', 'Y'])  #这里第三个元素还是指向list，只是list里的元素变了，但指向并没改变 |

1. 条件判断和循环

关于缩进，python中，相同缩进的代码视为代码块，缩进使用四个空格，不要使用Tab，更不要混合使用。

* 1. if

|  |
| --- |
| if a>b:  print(…) |

* 1. if else

|  |
| --- |
| if a>b:  …  else:  …. |

* 1. for循环

|  |
| --- |
| L = ['a','b','c','d']  for name in L: #name为每个元素的变量  print(name) |

* 1. while

|  |
| --- |
| while |

### dict和set类型

* 1. dict

带索引的集合，形式：

|  |
| --- |
| d = {  'Adam': 95,  'Lisa': 85,  'Bart': 59  } |

访问时可以用d[‘Adan’] ,但是若这个key不存在，直接报错。

所以可以使用：

if ‘Adam’in d:

…

或d.get(‘Adam’) //存在这个key，返回value，不存在，返回None

* 1. dict特点

1. 查找速度快
2. key不能重复
3. key-value序对是没有顺序的
4. 作为key的元素必须不可变，list不能作为key
   1. set

无序不重复的集合

s = set([‘A’,’B’,’C’]) //set()里传入一个list

‘A’ in s 返回True或False ,判断一个元素是否在set中

特点：

1. 元素必须是不可变对象
2. 无序的

向set添加元素：

s.add(‘D’) //若添加的元素已存在在set中，不会报错，但也不会再添加进去一次了

从set删除元素：

s.remove(‘D’) //若要删除的元素不存在，会报错

所以，add（）可以直接添加，remove()前需要先判断

### 函数

* 1. 定义函数

def 函数名(形参):

…

return …

* 1. 可变参数

def fn(\*args):

print args

Python解释器会把传入的一组参数组装成一个tuple传递给可变参数，因此，在函数内部，直接把变量 args 看成一个 tuple 就好了。

### 切片

* 1. 对list切片

前包含后不包含

L[0:3] 从索引0开始，到索引3，但不包括3

L[:3] 若第一个索引是0，也可以省略

L[:] 表示从头到尾

L[::3]表示每三个元素取出一个，即隔两个取一个（三个参数）

* 1. 对字符串切片

字符串也可以看成list进行切片

'ABCDEFG'[:3] 返回结果还是字符串

### 迭代

* 1. 集合

集合是指包含一组元素的数据结构，我们已经介绍的包括：

1. 有序集合：list，tuple，str和unicode；

2. 无序集合：set

3. 无序集合并且具有 key-value 对：dict

他们都可以用for in 来迭代

* 1. for value in L

迭代取出来的永远是元素本身，不是索引

要想取索引：使用enumerate(L)

|  |
| --- |
| L = ['Adam', 'Lisa', 'Bart', 'Paul']  for t in enumerate(L):  print t  输出：  (0, 'Adam')  (1, 'Lisa')  (2, 'Bart')  (3, 'Paul') |

enumerate将list有序集合每个元素都变成了tuple （index,value）

可以进一步简化：

|  |
| --- |
| L = ['Adam', 'Lisa', 'Bart', 'Paul']  for index,value in enumerate(L):  print index,'-',value |

index代表索引，value代表元素

* 1. zip（）

将两个list合并为一个list

|  |
| --- |
| zip([10, 20, 30], ['A', 'B', 'C'])  结果：[(10, 'A'), (20, 'B'), (30, 'C')] |

* 1. dict迭代

|  |
| --- |
| d = { 'Adam': 95, 'Lisa': 85, 'Bart': 59, 'Paul': 74 }  for i in d:  print i  结果：Lisa  Paul  Adam  Bart |

dict 迭代里的i，是key，并不是value

要想获取value，使用dict的values()方法，实际上是将所有的value值组合成一个list；

items()

将dict转化成包含tuple的list

|  |
| --- |
| d = { 'Adam': 95, 'Lisa': 85, 'Bart': 59 }  print d.items()  [('Lisa', 85), ('Adam', 95), ('Bart', 59)]  然后通过  for key,value in d.items():  …  循环 |

* 1. 列表生成式

range(1,101,2) 生成一个list集合，从1到100（不含101），且间隔是1

第一个参数是从几开始，

第二个参数是到几结束（不包含），

第三个参数是每几个数取一个出来

列表生成式格式：[x\*x for x in L]

前面是列表中的元素，后面跟for循环

后面还可以跟if判断语句

[x\*x for x in L if x%2==0] 只有满足if条件才将x返回到列表

还可以循环嵌套，只要在后面继续跟for in 即可

[x+y+z for x in L for y in M for z in N]

等同于

|  |
| --- |
| A=[]  for x in L:  for y in M:  for z in N:  A.append(x+y+z) |

### 函数式编程

* 1. map()

map(fn,list)

fn：函数

list：有序集合

map将list里的每个元素都经过fn函数处理，返回一个新的list，并不改变原来的list