**第6章 流程控制**

程序中的语句除了一条一条按顺序执行外，还可以根据特定条件控制语句的执行方向，也 就是流程控制o Python语言中,流程控制语句包括if条件判断语句,while、for循环语句等。

Sol館语働

if语句是条件判断语句，所谓条件判断，指的是只有满足某些条件，才允许做某件事情， 否则是不允许做的。例如，现实生活中，只有通过了面试，才能到公司上班。

在程序开发中，也经常会用到条件判断。例如，用户登录的时候，只有用户名和密码全部 正确，才被允许登录。类似这种需求的功能，都可以使用条件判断语句实现。

6JJ fiW

简单的if语句由三部分组成•分别是if关键字、判断结果真假的条件表达式以及当条件表 达式结果为真时的执行语句。简单if语句格式如下：

if表达式： 执行语句

上述格式中，只有当条件表达式的结果为True时，才会执 行后面的一个或多个执行语句。如果条件表达式结果为False, 则不会执行后面的语句。简单if语句的执行流程，如图6-1所示。

看一段代码：

age *= 3*

if age *== 3 :* print （°要上幼儿园了 *"）*

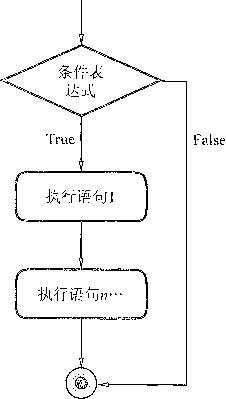
首先定义了一个变量ago,并给它赋值为3须然后使用if语 句判断表达式“age==3”是否为True,如果为True,则执行后 面的语句，输出“要上幼儿园了” o

图6』简单if语句

Python中的if语句与其他语言相比,有自己的特点:

(1 )在if后面的条件表达式不需要使用圆括号包含，而是使用冒号(：)表示条件表达式 的结束。

(2)多个执行语句不需要使用花括号包含。

6J.2暴歸褰达就 - 「

if语句根据条件表达式的值决定执行哪条语句，在Python语言中，判断一个表达式值的标 准如表6-1所示。

|  |  |
| --- | --- |
|  | 表6』判断表达式布尔值的标准 |
| 布尔值 | 表达式 |
| False | ° False  ° None  滲所有类型的数字0(包括浮点型、长整型和其他类型)  °空字符串  。空容器(包括元组,列表，字典，集合)等 . |
| True | 其他的所有值，包括:  ® True  口非0数字  ®非空字符串和容器等 |

标准的布尔值(bool)是True和False,但是在Python语言中,所有的值都可被当作 bo胡值，所以几乎不需要对它们进行显式转换，也可以说，Python语言自动会对这些值进 行转换。

下面看一个使用bool()函数判断一个表达式的值，例如：

>>> print (bool ( \* Python 是一门好用的语言 , ))

True

>>> print (bool(1^5))

True

>>> print(bool(''))

False

>>> print(bool(0))

False

6J.3參囂韻俸襄达饑 .

多个条件表达式之间可以通过布尔操作符(芻闭、。『和not)进行联合使用，实现多个条件 的判断。例如：

>>> month = 5

>>> weekday = 6

>>> day = 10

>>> if month == 5 and weekday == 6 and date >= 8 and day < 15:

…• print （"今天是5月的第二个星期日,母亲节到了 ,给妈妈打个电话吧。°） 今天是5月的第二个星期日*，*母亲节到了，给妈妈打个电话吧

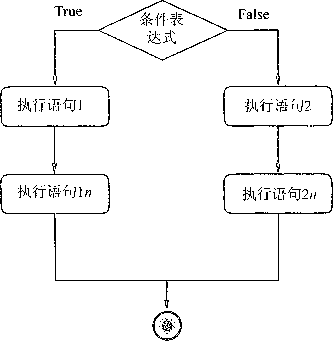
简单的if语句只规定了符合条件表达式之后执 行的语句,而if-else语句增加了不符合条件表达式的 情况下应执行的语句。

图 6-2 if-else 语句

if-else语句的语法格式如下:

if条件表达式:

执行语句1

else:

执行语句2

iFelse语句的含义是，如果条件表达式结果为 True,则执行if后的执行语句I;如果条件表达式结 果为False，则执行else后的执行语句2 °如图6-2所示。

举一个用户登录的例子，具体如下：

»> user\_name = °amdin"

>>> password = "123° >>> if user\_name == "admin" and password. == "123456":

• . • print （ n登录成功！即将进入主界面。° ）

o o \* else:

print （55您输入的用户名或者密码错误,请重新输入。°）

输出结果如下所示。

您输入的用户名或者密码错误，请重新输入。

这个例子里，使用条件表达式判断用户输入的用户名和密码是否与系统指定的用户名和密 码匹配，如果条件表达式结果为True,代表用户名和密码输入正确，可以进入系统；否则，条 件表达式结果为False,说明用户名或者密码不正确，登录失败，需要重新输入。

if能完成当条件满足时，做事情1； ifYlw能完成当条件满足时做事情1,否则做事情2； 如果有这样一种情况，当条件1满足时做事情1,当不满足条件1但是满足条件2时做事情2, 当不满足条件1、2时做事情3,那该怎么表示呢？此时可以在if语句中添加elif。

dif的语法格式如下：

if条件表达式1:

执行语句1

elif条件表达式2 :

执行语句2

elif条件表达式3 :

执行语句3

上述语句中歹首先判断条件表达式1是否为True,如果为True,则执行语句1,结束整个 if语句；否则跳过语句L继续判断Eif的条件表达式2,如果条件表达式2为True,则执行语 句2,结束整个if语句;否则继续判断条件表达式3,依此类推。

Rlif语句的执行流程如图6・3所示。

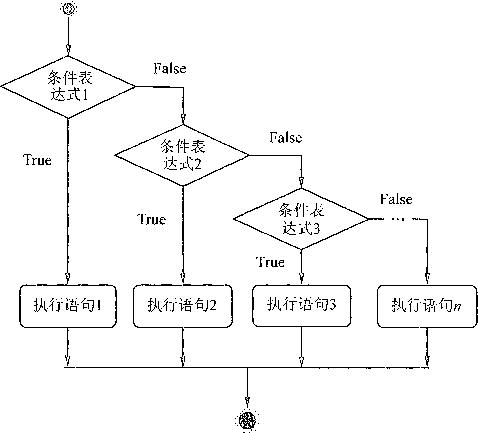


图6-3 诊elif语句

比如某网上商城根据会员积分划分会员的级别，刚注册的会员积分是0,每消费1元累计 1个积分，积分不断增加，会员等级也增加。代码如下：

>>> point = 2500

>>> if point == 0:

... print （11 注册会员"） elif 0 < point < 2000:

..print （° 铜牌会员"）

elif 2000 <= point < 10000:

"… print （"银牌会员°）

…“ elif 10000 <= point < 3Q000:

• . . print （°金牌会员°）

.o . elif point>=30000:

*…* print （°钻石会员°）

银牌会员

这段代码对会员的积分依次进行判断，假设会员积分是2500,首先进入if子句，条件表 达式point==0的值为False；接着进入第一个elif子句,条件表达式0 < point < 2000的值也为 False；于是继续进入下一个elif子句,条件表达式2000 v= point < 10000条件表达式为True, 执行这条子句下的执行语句，打印“银牌会员”的字样。剩下的分支就略过不执行了。 芸注意

©lif语句可以与攻se—起使用，此时else语句表示之前的条件之外的所有情况。格式如下: if条件表达式1:

执行语句1

elif条件表达式2 : 执行语句2

else :

执行语句*n*

氣1 応 「

if执行语句内部可以包含进一步的判断，这就是语句的嵌套。if嵌套语句的格式如下所示。 if条件表达式1:

执行语句1

if条件表达式2:

执行语句2

在if嵌套语句中，首先判断夕卜层if的条件表达式1是否为True,如果为True，则执行语句1, 并继续判断内层if的条件表达式2是否为True,如果为True,则执行语句2,否则不执行。

针对if嵌套语句，有两点需要说明：

。if语句可以多层嵌套，不仅限于两层。

韻外层和内层的if判断，都可以使用简单if语句、iFelse语句和elif语句。

比如需要根据年和月计算当月一共有多少天，代码如下所示。

>>> year = 2017

>>> month = 2

»> if month in [1,3,5,7,8,10,12]:

。… print ("%d 月有 31 天叫 % month)

…“ elif month in [4Z 6Z 9Z11]:

print (°%d 月有 30 天” % month)

«…elif month == 2:

*…。* if year % 400 == 0 or year % 4 == 0 and year % 100! = 0:

…• print ( °%d 年 %d 月有 29 天” % (year, month))

... else:

。… print ("%d 年％d 月有 28 天” % (year A month))

2017年2月有28天

我们知道，每月的总天数规律比较复杂，只能按照一定规则对月份和年份逐一判断，所以 用到了 if和elif语句。大部分月份都是30天或者31天，但是2月份特殊，闰年的2月份是29 天，其他年份的2月是28天，所以遇到2月份时，要使用嵌套语句对年份进行判断。

现在看以下两段代码：代码a和代码b,它们都有嵌套的两个if语句，唯一区别在于else 的位置不一样，那么它们的输出结果会是什么呢？

代码a：

>>> age = 4

>>> if age > 0:

*•…* if age〉= 6:

• . . print （ n超过6岁了，该上学了 "*）*

... print （"要开始学习了 °）

.。. else:

…“ print （°太小了*早*自由地玩吧"）

太小了，自由地玩吧

代码b：

>>> age = 4

>>> if age > 0:

if age〉= 6:

。… print （ °超过6岁了，该上学了 ° ）

…。 print （"要开始学习了 ° ）

..e else:

print （"太小了*号*自由地玩吧°）

从两段代码的运行结果可以看出，代码a输出了 else后的语句，而代码b没有任何输出。 代码a执行T else语句,而代码b没有,原因是代码a的else与内层if配对,而代码b的else 语句与外层if IBM o

在Python语言中，没有像别的语言那样使用花括号将代码块进行标记，而是依靠严格的 代码缩进来匹配if eke语句，这使得代码的书写更规范，可读性更高。

对于简单的i珏屁语句，可以使用判断表达式来简化代码的书写。在Python中判断表达 式的格式如下所示：

为True时的表达式if判定条件else为False时的表达式

例如，网吧管理员要根据身份证判断一个人是否成年，只有超过18岁才能进入网吧。这 种功能可以使用if语句实现，如下所示。

>>> age = *1*9

>>> is\_\_growup = False

>>> if age >= 18:

.… is growup = True

… print （"已成年n）

….else:

.o. is\_growup = False

…. print （"未成年n ）

已成年 同样的功能，也可以使用判断表达式表示，如下所示。

>>> age = 19

>>> is\_growup = True if age >= 18 else False

>>> print （"已成年"）if is\_growup else print （n 未成年 n） 已成年 —

这两段代码的效果和运行结果是一样的，但是显然判断表达式的代码更简洁，代码量更少。 在实际开发中，有经验的开发者经常使用判断表达式来简化代码的书写，提高开发速度。

［気?注意：在Pythoii语言中没有其他语言常见的三目运算符（条件表达式?a:b）,但是可以 逼1±判断表达式实现类似的功能。

So 2循环璃甸

在条件判断if语句，如果条件表达式成立，会执行一次语句。而循环语句则可以不断循 环执行语句。在Python中常用的循环包括while循环和for 循环。

虱2』whl@

while循环是一个条件循环语句，当条件满足时重复执 行语句，直到条件不满足为止。while循环的格式如下所示。

while条件表达式：

执行语句1

执行语句2

在while循环中,首先判断条件表达式是否成立,如 果成立,则依次执行语句1、执行语句2等。然后重新判 断条件表达式是否成立，如果成立则再次执行语句1、语 句2等。每次执行完都重新判断条件表达式,直到条件表 达式不成立，结束循环，不再执行循环体内的语句。

while循环流程如图6-4所示。

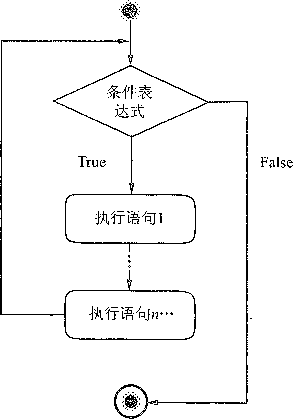
while循环常用于计数循环，比如使用循环计算10!的值，即计算1 x2x3 x x 10的结果, 代码如下所示。

图6-4 while循环

»> i = 1

»> result = 1

>>> while i <= 10:

result \*= i

... i += 1

>» print (result)

输出结果如下所7K。

3628800

在上述代码中，变量i表示要计算的乘数9变量冀suit用于存储结果。最初变量i的值为L 进入while循环，首先判断条件表达式iv=10,判断结果的值为True,执行循环体内的语句， 将［乘冀suit并赋值给result,并将i值加1, i值变成2,继续下一次循环，判断条件表达式 i<=10的值仍为True，将2乘result并赋值给result, i值加1,然后继续下一个迭代金依次进行， 直到i=ll，条件表达式i<=10的值为False,结束循环。在循环外部打印计算结果result的值。

&22 wMeWWS .

while循环里还可以嵌套while循环，它的格式如下:

while条件表达式1:

执行语句1

执行语句2

while条件表达式2 : 执行语句3 执行语句4

在while循环嵌套里，首先判断外层的条件表达式1是否成立，如果成立，则执行语句1、2, 并判断条件表达式2是否成立，如果成立，则执行语句3、4」然后再次判断条件表示式2和执 行语句3、4,直到条件表达式2不成立，内层的while循环结束。也就是说，每执行一次外层 的while语句，都要将内层的while循环重复执行到结束。

例如使用循环嵌套语句打印阶梯\*形，代码如下所示。

i=l

while i<=5:

j = 1

while j<= i:

print (f end= ' 1 )

j+= 1

print("\n")

i+= 1

输出结果如下所示。

从上述代码可以看出，while循环可以嵌套，嵌套的形式和层数都不受限制。当然，如果 嵌套的层级太多，代码会变得很复杂，不好理解，此时,最好调整一下代码结构，将嵌套的层 数控制在3层以内。

観2芻雍循舔

使用while循环时可能会出现死循环。由于while循环具有重复执行的特点，如果条件表 达式的值为True,则循环会一直执行，永远不会结束，这就是死循环。

比如下面的代码，就是一个死循环。

»> i=l

>>> while i:

".. print(i+1)

这段代码中，循环条件i的值一直都是1,所以循环的条件一直为True,也就一直执行， 永远不会结束。

在实际开发中，有些程序是不需要终止循环的，比如操作系统监听系统消息，只要操作系 统还在运行，就始终要循环监听。除了这些程序之外须我们在编程中要谨防不必要的死循环。

&2感 f@r»

在Pythoii中，版语句用于遍历序列的元素(如字符串、元组或列表)或其他可迭代对象， 它按照元素在可迭代对象中的顺序一一迭代，并在处理完所有元素后自动结束循环。for语句的 格式如下:

for临时变量in可迭代对象:

执行语句1

执行语句2

每次循环，临时变量都会被赋值为可迭代对象的当前元素，提供给执行语句使用。 比如以下代码，使用fo『语句遍历字符串的每个字符：

>>> company ="传智播客 °

>>> for c in company:

-。。 print(c)

输出结果：

传

智

播

客

(1 ) for循环遍历序列

for循环可用于遍历序列，示例如下所示。

»> words = nlove°f nItcast°]

>>> for word in words:

,• • print(word)

输岀结果：

1

love

Itcast

(2 ) for循环遍历字典

在Python中，foi•循环还可以遍历字典的所有键，使用起来非常方便，比如下例：

>» diet = {"name0:"xiaoming0 r "age":"22° "gender":"Male"}

>>> for key in diet:

.… print (key + "的值是:"+ diet [key])

输出结果：

name 的值是：xiaoming

age的值是：22 gender 的值是:Male

需要注意的是，由于字典里的元素顺序没有定义，所以使用fb『语句遍历字典内的元素时， 顺序是不确定的。

&2忐 raog® f©r flW .「 「

从前面的章节，我们已经知道，mnge()函数表示一个不变的数字序列，弟號攻)函数经常与 for循环一起使用，看一段代码：

>>> for i in range (0 7 5):

.。. print (i)

输出结果：

0

1

2

3

4

鼠2 思

1 items方法

在循环遍历字典时，关键字和对应的值可以使用血ms()方法同时得到。

例如下面的代码：

>>> demo\_dict = demo\_\_dict. items ():

>>> for kf v in { ("name"7"xiaoming")z ("age"f"22")f ("gender0 f "Male"}

。… print (k +"的值是:"+ v)

输出结果：

name 的值是：xiaoming

age的值是：22 gender 的值是：Male

2b enumerate 函数

在循环遍历序列时，索引位置和对应值可以使用enumerate()函数同时得到。

例如下面的代码：

>>> for i7 v in enumerate(["I° z"love","Itcast"]):

.•. print(i, v)

输出结果：

0 I

1. love
2. Itcast

& zip函数

在同时循环遍历两个或多个序列时，可以使用zip()函数将这些序列打包。 例如下面的代码：

»> infos = [°姓名"年龄”,°性别° ]

»> student =[”小明七，，22时/男彩]

>>> for (i7 s) in zip(infoszstudent):

…. print ('学生的｛ 0 ｝是｛1｝ L format (i, s))

输出结果：

学生的姓名是小明

学生的年龄是22

学生的性别是男

4 D reversed 函数

如果需要按照从后向前的顺序，也就是逆向的顺序循环序列的话,可以使用reversed()函数。 例如下面的代码：

>>> for i in reversed(range(1,5)):

... print(i)

输出结果:

4

3

2

1

5. sorted 函数

如果要按排序后的顺序遍历序列的话，可以使用sortedO函数。sorted。函数会生成一个新 的已排序序列，不会改动原来的序列。示例如下：

>>> colors = [1 red!f 5 green 5, 5 purple \*, \* yellow \ 5 white', 'black1]

>>> for c in sorted(colors):

..“ print(c)

输出结果：

black

green

purple

red

white

yellow

第6章流程控制Q 113 从输出结果可知，使用sorted()函数将序列内的元素按照字母顺序依次循环。

「(B虛寄

我们可以看出，大多数容器对象都可以使用for循环遍历，比如:

for a in [1,2,3,4]:

(print(a)

for a in (17 2^ 3^ 4):

(print(a)

for key in {"name":"xiaoMing"? "age°:"23°}:

(print(key)

for char in "abc":

(print(char)

使用f0『循环访问的方式简洁、易用成实际上，所有可迭代的对象都可以使用f(M循环来遍历， 这是由fOF循环的原理决定的。

f成循环使用了迭代器来访问容器中的元素，迭代器是一个对象，它有一个nextO方法，用于

返回下一个元素。当使用fi)『循环访问容器中的下一个元素时，调用迭代器的安制)方法就可以得到。

当容器内的所有元素全部访问完后，迭代器会发出一个Stopiteration异常,告诉它的调用者， 迭代已经完成。for语句只需要捕获这个异常，就知道迭代何时结束。

使用谕此函数可以得到容器的迭代器，知道了 for循环的工作原理之后，就可以使用迭代 器和谕『()函数模拟foM盾环了。例如下面的代码：

>>> numbers = [ 1f 2 f 3]

>>> for i in numbers:

... print (i)

输出结果？

1

2

3

上述代码使用fo『循环遍历了列表mimb©指中的每一个元素，并打印出来。我们可以使用 迭代器和iter()函数来实现同样的功能，如下所示。

>>> numbers = [ 12 z 3]

>>> iter\_num = iter(numbers)

>>> while True:

…. try:

e . element = next (iter\_\_num)

…。 print (element)

..。 except Stopiteration:

.。。 break

输出结果：

1

2

3

上述代码先调用谕『()函数获得一个迭代器，赋值给iter\_num,然后调用next()函数获取迭

代器的下一个元素并打印，直到迭代器抛岀Stopiteration异常，则说明迭代结束。’

从上面的两段代码可以看出for循环的工作原理了，在实际开发中，for语句已经自动调用 了迭代器的功能和itor()函数，我们在开发中，可以使用fo『循环遍历所有可迭代的对象。

6.2.8

列表推导式是利用现有列表创建一个新列表的方法，它是一个非常有用的工具，一般用于 动态地创建列表。它的语法格式如下所示。

[新兄素表达式for临时变量in可迭代对象if条件表达式]

在这个列表推导式中，for循环遍历可迭代对象的所有元素，然后验证该元素是否符合if 条件表达式，并对通过验证的元素根据新元素表达式生成新的元素。

简单的列表推导式可以省略if语句，看一个例子。

>>> demo\_list = [x\*x for x in range(0f 6)]

>>> demo\_list

[0, 1, 4,幻 16, 25]

上述代码首先遍历了 0〜5的范围内的所有元素，然后使用每个元素的二次方生成了一个新 的列表。

在列表推导式中使用if语句进行条件判断，可以对元素进行筛选，示例如下：

>>> demo\_list = [x\*x for x in range(0z 6) if x % 2 == 0]

>>> demo\_list

[0, 4, 16]

在上述代码中，if语句判断元素是否为偶数，只有偶数才计算它的二次方。

列表推导式可以叠加多个for循环，比如下例使用两个for循环生成一个矩阵:

>>> demo\_list = [(x,y) for x in range(2) for y in range(3)]

>>> demo\_list

[(0, 0) z (0, 1) , (0z 2), (1, 0) , (1, 1) , (lf *2)]*

上述代码遍历了两个列表,将两个列表的元素分别取出，组成T新的元素，最终生成T矩阵。

嵐3灘陽循环

通常情况下，循环语句会执行完所有的情况后自然结束。但是有些情况下，我们需要将当 前正在执行的循环停止，也就是跳出循环，跳出循环语句，包括跳出整个循环的break语句和 放弃本次循环的continue语句。

®-3n1 break

bweak可跳出离它最近一级的循环，能够用于while循环和for语句中，通常结合if语句判断, 当满足某个条件时，退出循环。

用于for语句：

»> name = "itcast"

»> for word in name:

… print、————一叫)

…。 if(word == °a"):

* • . break
* .。 print(word)

输出结果：

上面的for循环，遍历了 name变量的每个字符，但是有一个跳岀条件，即当遍历的字符 等于％”时，就跳出循环，不再继续遍历剩下的字符。

用于while循环：

»> i=0

>>> max=5

>>> while i<10:

… i+= 1

... print ( " n )

…。 if(i == max):

•。. break

*昨* o . print(i)

输出结果：

上述代码使用while循环依次打印0~9的值，但是定义了一个最大值max (也就是5 ) 9 如果循环变量达到最大值，则跳出循环，停止打印。

注意：break跳出的是离它最近的一级循环，如果有循环嵌套，则内层的循环跳出后不 影响外层的循环。

®d3d2 ©ooftmo® 牆匍

continue语句用于跳岀当前迭代，继续执行当前循环的下一个迭代。当遇到continiw语句时, 系统会忽略当前迭代中剩下的代码，从头开始下一个迭代。此时膏在foM盾环中，会验证是否 有下一个元素;在while循环中,会验证条件表达式是否成立o

116 Pjthoo RffiSOS: Pyfthoo

例如,以下代码从列表中找出所有的正数:

>>> for element in [0 z -2 r 5 f 7 7 -10 ]:

*。…* if element <= 0 :

o。. continue

... print(element)

输岀结果？

5

7

在上述代码中，遍历列表中的所有元素，判断该元素的值，当值小于0时跳出当前迭代， 忽略剩下的循环语句，开始遍历下一个元素，判断它的值。

(☆注意 .

break和continue只能用在循环中,除此之外不能单独使用。

6q3d3 pass 麗甸

P龄S语句表示不做任何事情，它用于语法上必须有贸但程序什么也不用做的场合。由于 Pythoii没有使用花括号来标记代码块，无法使用空花括号或者冒号表示空代码，在有些语法上 要求必须有代码的地方，如果什么都不写，系统会提示语法错误，此时就可以使用pass语句。

除此之外，还可以使用pass语句作为开发中的小技巧，用来标记留待以后开发的代码，或 者是占位符。示例如下：

while a>10:

pass #待添加代码

综上所述，使用pa%语句作为占位符的好处有如下几点：

•标记尚未完成的代码,以免忘记;

•不急于完成具体代码，先完成代码的结构，让开发者在设计的层面上进行思考。

®a3n4 躊匐

Pythoii语言与其他语言不同的一点是，它的e屁语句不仅可以在if语句中出现，还可以用 于循环语句。else语句的格式如下所示。

for临时变量in可迭代对象：

执行语句

else :

执行语句……

或者：

while条件表达式：

执行语句

else :

执行语句......

诳院语句作为循环的子句，在循环语句正常结束后执行，也就是在fb『语句迭代完所有元

素后，或者while循环的条件表达式值为False以后执行。但是在循环遇到break语句而终止的 情况下不执行。

在编程时可以将else语句与语句结合使用，当在循环中发生了某事或者找到了某元 素时使用bgk跳出循环，此时不执行else语句。如果在循环中没有执行break,则会执行else 语句。例如，打印出15以内的所有的质数膏质数就是不能被除了 1和自身之外其他的数字整 除的数。判断一个数〃是不是质数，需要用所有小于〃的数除以払 如果余数为0,则花不是 质数；如果所有小于阪大于1的数除以"的余数都不为0,则说明〃是质数。以下代码就使用 循环嵌套和else语句找岀了 15以内的所有质数。

»> i = 1

>>> while i<= 15:

… i+=l

…” j -1

“。. while j< i~l:

… j+= 1 '

。… if i% j ! = 0 :

... continue

... else:

... break

。.else:

…• print （i）

输出结果：

2

3

5

7

11

13

上述代码使用了两层while循环，第一层循环逐一判断所有15以内的数字（用变量i表示）; 第二层循环使用所有大于1小于II的数字（用变量j表示）除以i,如果余数不等于0,则将j加L 继续下一个迭代；如果i/j等于0,代表i不是质数，此时使用跳出循环。如果所有的j循 环完都没有执行切sk语句，说明i是质数，此时执行司笑里的语句，将i打印出来。