# 1 条件语句

1. if 判断条件：

执行语句……else：

执行语句……

2. if 判断条件1:

执行语句1……elif 判断条件2:

执行语句2……elif 判断条件3:

执行语句3……else:

执行语句4……

例：

num = 5 if num == 3: # 判断num的值

print 'boss' elif num == 2:

print 'user'elif num == 1:

print 'worker'elif num < 0: # 值小于零时输出

print 'error'else:

print 'roadman' # 条件均不成立时输出

# 2 While循环语句

while 语句时还有另外两个重要的命令 continue，break 来跳过循环，continue 用于跳过该次循环，break 则是用于退出循环，此外"判断条件"还可以是个常值，表示循环必定成立，具体用法如下：

# continue 和 break 用法

i = 1while i < 10:

i += 1

if i%2 > 0: # 非双数时跳过输出

continue

print i # 输出双数2、4、6、8、10

i = 1while 1: # 循环条件为1必定成立

print i # 输出1~10

i += 1

if i > 10: # 当i大于10时跳出循环

break

！！ 以上的无限循环你可以使用 CTRL+C 来中断循环。

## 循环使用 else 语句

在 python 中，for … else 表示这样的意思，for 中的语句和普通的没有区别，else 中的语句会在循环正常执行完（即 for 不是通过 break 跳出而中断的）的情况下执行，while … else 也是一样。

#!/usr/bin/python

count = 0

while count < 5:

print count, " is less than 5"

count = count + 1

else:

print count, " is not less than 5

！！ 类似if语句的语法，如果你的while循环体中只有一条语句，你可以将该语句与while写在同一行中

# 3 for 循环语句

for循环的语法格式如下：

for iterating\_var in sequence:

statements(s)

**实例：**

#!/usr/bin/python# -\*- coding: UTF-8 -\*-

for letter in 'Python': # 第一个实例

print '当前字母 :', letter

fruits = ['banana', 'apple', 'mango']

for fruit in fruits: # 第二个实例

print '当前字母 :', fruit

print "Good bye!"

## 3.1通过序列索引迭代

另外一种执行循环的遍历方式是通过索引，如下实例：

#!/usr/bin/python# -\*- coding: UTF-8 -\*-

fruits = ['banana', 'apple', 'mango']

for index in range(len(fruits)):

print '当前水果 :', fruits[index]

print "Good bye!"

以上实例输出结果：

当前水果 : banana当前水果 : apple当前水果 : mangoGood bye!

## 3.2 循环使用 else 语句

在 python 中，for … else 表示这样的意思，for 中的语句和普通的没有区别，else 中的语句会在循环正常执行完（即 for 不是通过 break 跳出而中断的）的情况下执行，while … else 也是一样。

如下实例：

#!/usr/bin/python# -\*- coding: UTF-8 -\*-

for num in range(10,20): # 迭代 10 到 20 之间的数字

for i in range(2,num): # 根据因子迭代

if num%i == 0: # 确定第一个因子

j=num/i # 计算第二个因子

print '%d 等于 %d \* %d' % (num,i,j)

break # 跳出当前循环

else: # 循环的 else 部分

print num, '是一个质数'

Break

用来打破最小封闭的for 或者while循环：

就是如果有两层for只会打断最后一层

## continue 语句

Python continue 语句跳出本次循环，而break跳出整个循环。

continue 语句用来告诉Python跳过当前循环的剩余语句，然后继续进行下一轮循环。

continue语句用在while和for循环中。

## pass 语句

Python pass是空语句，是为了保持程序结构的完整性。

pass 不做任何事情，一般用做占位语句。

for letter in 'Python':

if letter == 'h':

pass #占位，不会报错

print '当前字母 :', letter