

# 目标

---

- 了解循环
- while语法【重点】
- while应用
- break和continue
- while循环嵌套【重点】
- while循环嵌套应用【难点】
- for循环

## 一. 循环简介

---

### 1.1 循环的作用

---

思考：假如我有个女朋友，有一天我们闹矛盾生气了，女朋友说：道歉，说100遍“媳妇儿，我错了”。这个时候程序员会怎么做？

答：100遍 `print('媳妇儿，我错了')`

思考：复制粘贴100次吗？

答：重复执行100次一样的代码，程序中循环即可

循环的作用：让代码更高效的重复执行。

### 1.2 循环的分类

---

在Python中，循环分为 `while` 和 `for` 两种，最终实现效果相同。

## 二. while的语法

---

```
1 while 条件:
2     条件成立重复执行的代码1
3     条件成立重复执行的代码2
4     .....
```

### 2.1 快速体验

---

需求：复现重复执行100次 `print('媳妇儿，我错了')`（输出更简洁一些，我们这里设置5次）。

分析：初始值是0次，终点是5次，重复做的事情输出“媳妇儿，我错了”。

```
1 # 循环的计数器
2 i = 0
3 while i < 5:
4     print('媳妇儿，我错了')
5     i += 1
6
7 print('任务结束')
```

## 三. while的应用

### 3.1 应用一：计算1-100累加和

分析：1-100的累加和，即 $1 + 2 + 3 + 4 + \dots$ ，即前两个数字的相加结果 + 下一个数字(前一个数字 + 1)。

```
1 i = 1
2 result = 0
3 while i <= 100:
4     result += i
5     i += 1
6
7 # 输出5050
8 print(result)
```

注意：为了验证程序的准确性，可以先改小数值，验证结果正确后，再改成1-100做累加。

### 3.2 应用二：计算1-100偶数累加和

分析：1-100的偶数和，即 $2 + 4 + 6 + 8 + \dots$ ，得到偶数的方法如下：

- 偶数即是和2取余结果为0的数字，可以加入条件语句判断是否为偶数，为偶数则累加
- 初始值为0 / 2，计数器每次累加2

#### 3.2.1 方法一：条件判断和2取余数则累加

```
1 # 方法一：条件判断和2取余数为0则累加计算
2 i = 1
3 result = 0
4 while i <= 100:
5     if i % 2 == 0:
6         result += i
7         i += 1
8
9 # 输出2550
10 print(result)
```

### 3.2.2 方法二：计数器控制

```
1 # 方法二：计数器控制增量为2
2 i = 0
3 result = 0
4 while i <= 100:
5     result += i
6     i += 2
7
8 # 输出2550
9 print(result)
```

## 四、break和continue

break和continue是循环中满足一定条件退出循环的两种不同方式。

### 4.1 理解

举例：一共吃5个苹果，吃完第一个，吃第二个...，这里"吃苹果"的动作是不是重复执行？

情况一：如果吃的过程中，吃完第三个吃饱了，则不需要再吃第4个和第五个苹果，即是吃苹果的动作停止，这里就是break控制循环流程，即**终止此循环**。

情况二：如果吃的过程中，吃到第三个吃出一个大虫子...是不是这个苹果就不吃了，开始吃第四个苹果，这里就是continue控制循环流程，即**退出当前一次循环继而执行下一次循环代码**。

#### 4.1.1 情况一：break

```
1 i = 1
2 while i <= 5:
3     if i == 4:
4         print(f'吃饱了不吃了')
5         break
6     print(f'吃了第{i}个苹果')
7     i += 1
```

执行结果：

```
C:\Users\黑马程序员\AppData\Local\Programs\Python\Python37\python3.exe
吃了第1个苹果
吃了第2个苹果
吃了第3个苹果
吃饱了不吃了
Process finished with exit code 0
```

## 4.1.2 情况二：continue

```
1 i = 1
2 while i <= 5:
3     if i == 3:
4         print(f'大虫子，第{i}个不吃了')
5         # 在continue之前一定要修改计数器，否则会陷入死循环
6         i += 1
7         continue
8     print(f'吃了第{i}个苹果')
9     i += 1
```

执行结果：

```
C:\Users\黑马程序员\AppData\Local\Programs\Python\Python37\python3.exe
吃了第1个苹果
吃了第2个苹果
大虫子，第3个不吃了
吃了第4个苹果
吃了第5个苹果
Process finished with exit code 0
```

# 五. while循环嵌套

## 5.1 应用场景

故事梗概：有天女朋友又生气了，惩罚：说3遍“媳妇儿，我错了”，这个程序是不是循环即可？但如果女朋友说：还要刷今天晚饭的碗，这个程序怎么书写？

```
1 while 条件:
2     print('媳妇儿，我错了')
3     print('刷晚饭的碗')
```

但如果女朋友还是生气，把这套惩罚要连续3天都执行，有如何书写程序？

```
1 while 条件:
2     while 条件:
3         print('媳妇儿, 我错了')
4     print('刷晚饭的碗')
```

## 5.2 语法

```
1 while 条件1:
2     条件1成立执行的代码
3     .....
4     while 条件2:
5         条件2成立执行的代码
6         .....
```

总结：所谓while循环嵌套，就是一个while里面嵌套一个while的写法，每个while和之前的基础语法是相同的。

## 5.3 快速体验：复现场景

### 5.3.1 代码

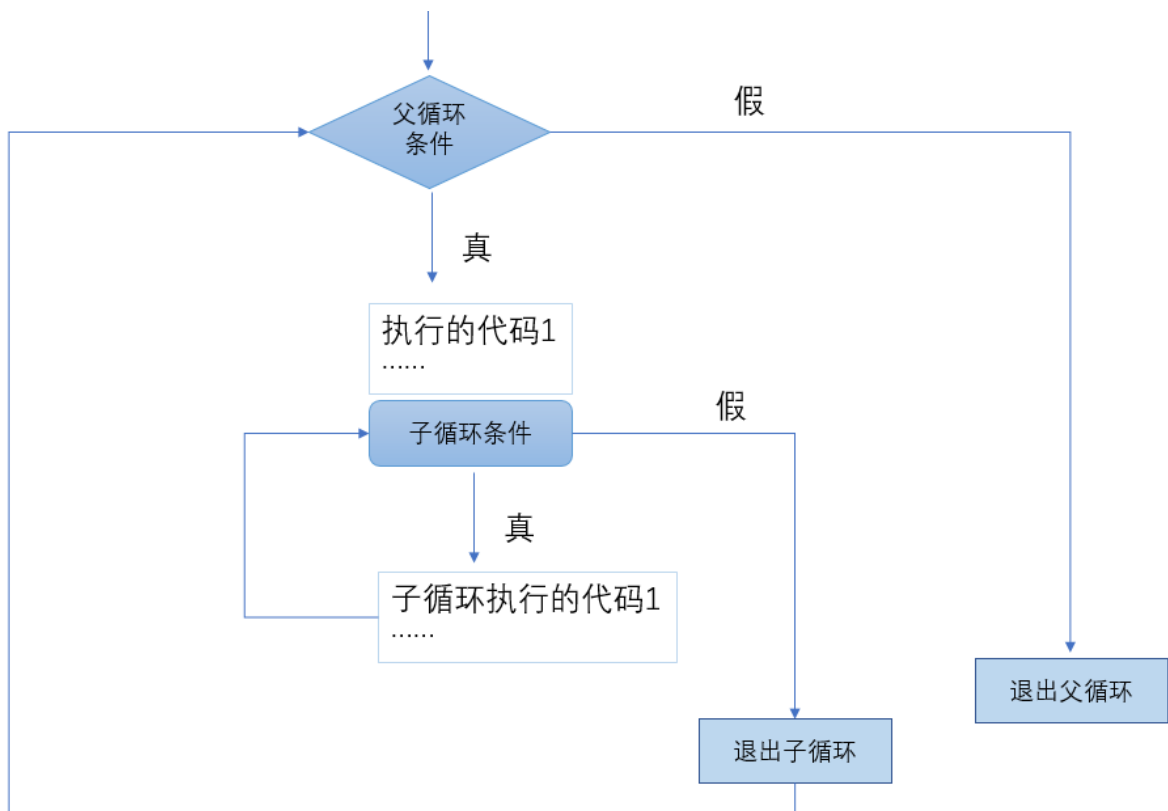
```
1 j = 0
2 while j < 3:
3     i = 0
4     while i < 3:
5         print('媳妇儿, 我错了')
6         i += 1
7     print('刷晚饭的碗')
8     print('一套惩罚结束-----')
9     j += 1
```

### 5.3.2 执行结果

```
Run: 循环嵌套快速体验 x
C:\Users\黑马程序员\AppData\Local\Programs\Python\Python37\python3.exe
媳妇儿，我错了
媳妇儿，我错了
媳妇儿，我错了
刷晚饭的碗
一套惩罚结束-----
媳妇儿，我错了
媳妇儿，我错了
媳妇儿，我错了
刷晚饭的碗
一套惩罚结束-----
媳妇儿，我错了
媳妇儿，我错了
媳妇儿，我错了
刷晚饭的碗
一套惩罚结束-----
Process finished with exit code 0
```

### 5.3.3 理解执行流程

当内部循环执行完成之后，再执行下一次外部循环的条件判断。



## 六. while循环嵌套应用

### 6.1 应用一：打印星号(正方形)

## 6.1.1 需求

```
1 *****
2 *****
3 *****
4 *****
5 *****
```

## 6.1.2 代码

分析：一行输出5个星号，重复打印5行

```
1 # 重复打印5行星星
2 j = 0
3 while j <= 4:
4     # 一行星星的打印
5     i = 0
6     while i <= 4:
7         # 一行内的星星不能换行，取消print默认结束符\n
8         print('*', end='')
9         i += 1
10    # 每行结束要换行，这里借助一个空的print，利用print默认结束符换行
11    print()
12    j += 1
```

## 6.2 应用二：打印星号(三角形)

### 6.2.1 需求

```
1 *
2 **
3 ***
4 ****
5 *****
```

### 6.2.2 代码

分析：一行输出星星的个数和行号是相等的，每行：重复打印行号数字个星号，将打印行星号的命令重复执行5次实现打印5行。

```

1  # 重复打印5行星星
2  # j表示行号
3  j = 0
4  while j <= 4:
5      # 一行星星的打印
6      i = 0
7      # i表示每行里面星星的个数，这个数字要和行号相等所以i要和j联动
8      while i <= j:
9          print('*', end='')
10         i += 1
11     print()
12     j += 1

```

## 6.3 九九乘法表

### 6.3.1 执行结果

```

1*1=1
1*2=2   2*2=4
1*3=3   2*3=6   3*3=9
1*4=4   2*4=8   3*4=12  4*4=16
1*5=5   2*5=10  3*5=15  4*5=20  5*5=25
1*6=6   2*6=12  3*6=18  4*6=24  5*6=30  6*6=36
1*7=7   2*7=14  3*7=21  4*7=28  5*7=35  6*7=42  7*7=49
1*8=8   2*8=16  3*8=24  4*8=32  5*8=40  6*8=48  7*8=56  8*8=64
1*9=9   2*9=18  3*9=27  4*9=36  5*9=45  6*9=54  7*9=63  8*9=72  9*9=81

```

### 6.3.2 代码

```

1  # 重复打印9行表达式
2  j = 1
3  while j <= 9:
4      # 打印一行里面的表达式 a * b = a*b
5      i = 1
6      while i <= j:
7          print(f'{i}*{j}={j*i}', end='\t')
8          i += 1
9      print()
10     j += 1

```

## 七、for循环

### 7.1 语法

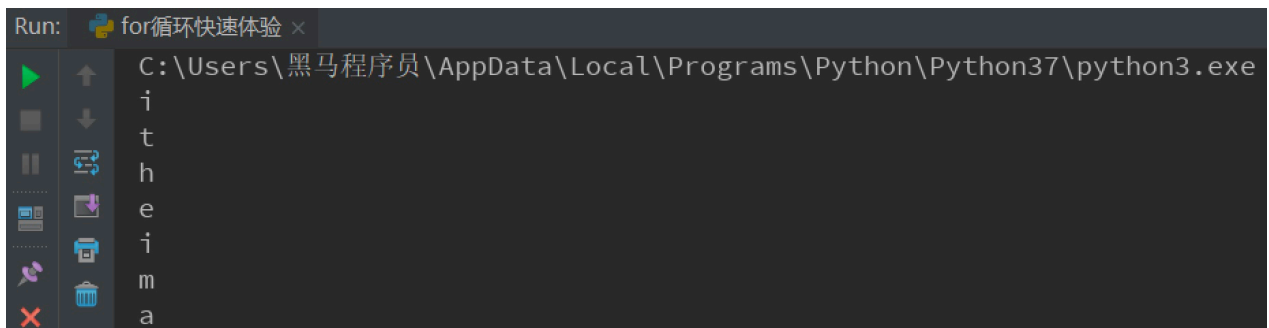


```
1 for 临时变量 in 序列:
2     重复执行的代码1
3     重复执行的代码2
4     .....
```

## 7.2 快速体验

```
1 str1 = 'itheima'
2 for i in str1:
3     print(i)
```

执行结果：

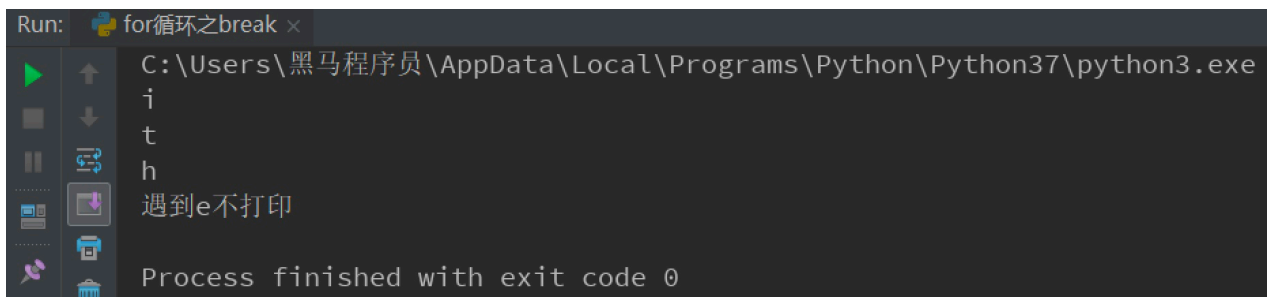


```
Run: for循环快速体验 x
C:\Users\黑马程序员\AppData\Local\Programs\Python\Python37\python3.exe
i
t
h
e
i
m
a
```

## 7.3 break

```
1 str1 = 'itheima'
2 for i in str1:
3     if i == 'e':
4         print('遇到e不打印')
5         break
6     print(i)
```

执行结果：



```
Run: for循环之break x
C:\Users\黑马程序员\AppData\Local\Programs\Python\Python37\python3.exe
i
t
h
遇到e不打印
Process finished with exit code 0
```

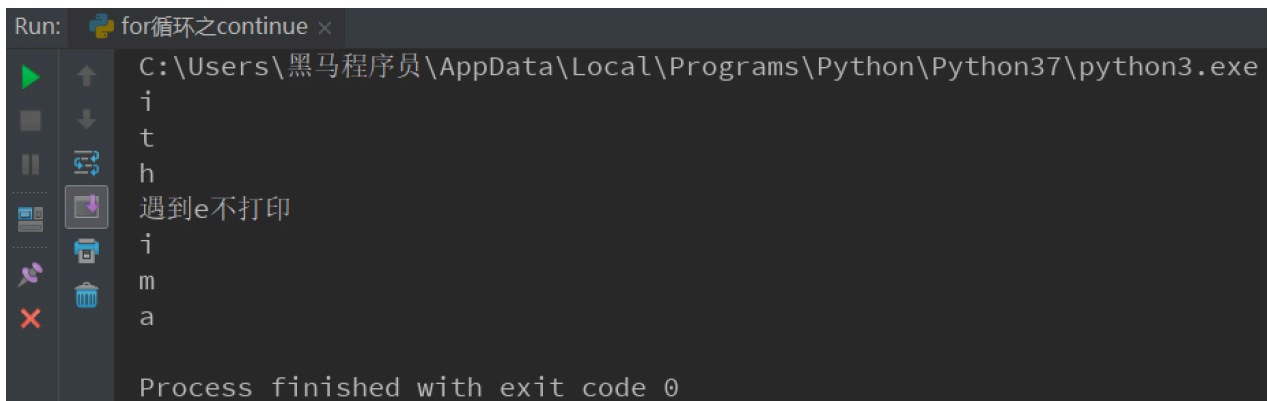
## 7.4 continue

```

1  str1 = 'itheima'
2  for i in str1:
3      if i == 'e':
4          print('遇到e不打印')
5          continue
6      print(i)

```

执行结果：



```

Run: for循环之continue x
C:\Users\黑马程序员\AppData\Local\Programs\Python\Python37\python3.exe
i
t
h
遇到e不打印
i
m
a
Process finished with exit code 0

```

## 八. else

循环可以和else配合使用，else下方缩进的代码指的是当循环正常结束之后要执行的代码。

### 8.1 while...else

需求：女朋友生气了，要惩罚：连续说5遍“媳妇儿，我错了”，如果道歉正常完毕女朋友就原谅我了，这个程序怎么写？

```

1  i = 1
2  while i <= 5:
3      print('媳妇儿，我错了')
4      i += 1
5  print('媳妇儿原谅我了...')

```

思考：这个print是不是没有循环也能执行？

#### 8.1.1 语法

```

1  while 条件:
2      条件成立重复执行的代码
3  else:
4      循环正常结束之后要执行的代码

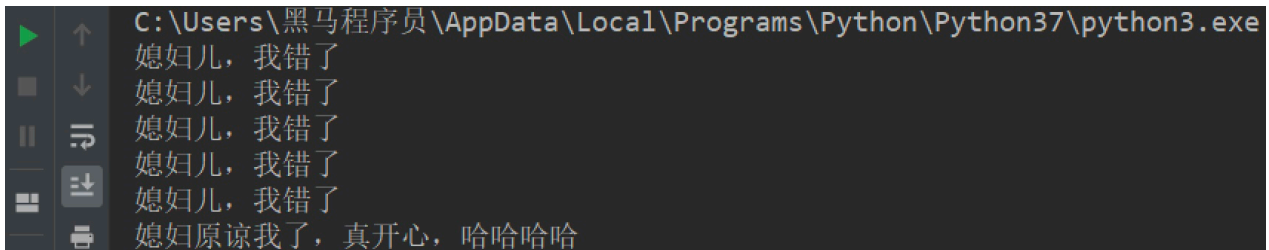
```

#### 8.1.2 示例

```

1 i = 1
2 while i <= 5:
3     print('媳妇儿，我错了')
4     i += 1
5 else:
6     print('媳妇原谅我了，真开心，哈哈哈哈')

```



```

C:\Users\黑马程序员\AppData\Local\Programs\Python\Python37\python3.exe
媳妇儿，我错了
媳妇儿，我错了
媳妇儿，我错了
媳妇儿，我错了
媳妇儿，我错了
媳妇原谅我了，真开心，哈哈哈哈

```

### 8.1.3 退出循环的方式

需求：女朋友生气，要求道歉5遍：媳妇儿，我错了。道歉到第三遍的时候，媳妇埋怨这一遍说的不真诚，是不是就是要退出循环了？这个退出有两种可能性：

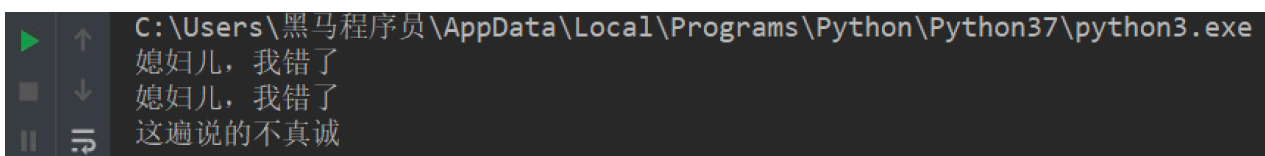
- 更生气，不打算原谅，也不需要道歉了，程序如何书写？
- 只一遍不真诚，可以忍受，继续下一遍道歉，程序如何书写？

#### 1. break

```

1 i = 1
2 while i <= 5:
3     if i == 3:
4         print('这遍说的不真诚')
5         break
6     print('媳妇儿，我错了')
7     i += 1
8 else:
9     print('媳妇原谅我了，真开心，哈哈哈哈')

```



```

C:\Users\黑马程序员\AppData\Local\Programs\Python\Python37\python3.exe
媳妇儿，我错了
媳妇儿，我错了
这遍说的不真诚

```

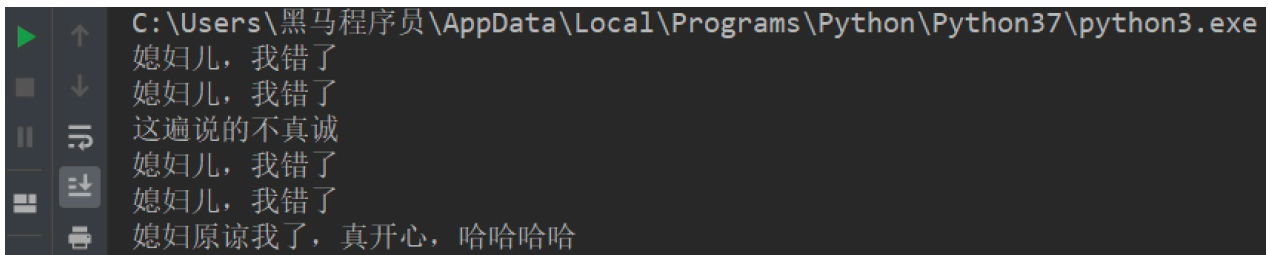
所谓else指的是循环正常结束之后要执行的代码，即如果是break终止循环的情况，else下方缩进的代码将不执行。

#### 2. continue

```

1  i = 1
2  while i <= 5:
3      if i == 3:
4          print('这遍说的不真诚')
5          i += 1
6          continue
7      print('媳妇儿，我错了')
8      i += 1
9  else:
10     print('媳妇原谅我了，真开心，哈哈哈哈')

```



```

C:\Users\黑马程序员\AppData\Local\Programs\Python\Python37\python3.exe
媳妇儿，我错了
媳妇儿，我错了
这遍说的不真诚
媳妇儿，我错了
媳妇儿，我错了
媳妇原谅我了，真开心，哈哈哈哈

```

因为continue是退出当前一次循环，继续下一次循环，所以该循环在continue控制下是可以正常结束的，当循环结束后，则执行了else缩进的代码。

## 8.2 for...else

### 8.2.1 语法

```

1  for 临时变量 in 序列:
2      重复执行的代码
3      ...
4  else:
5      循环正常结束之后要执行的代码

```

所谓else指的是循环正常结束之后要执行的代码，即如果是break终止循环的情况，else下方缩进的代码将不执行。

### 8.2.2 示例

```

1  str1 = 'itheima'
2  for i in str1:
3      print(i)
4  else:
5      print('循环正常结束之后执行的代码')

```

### 8.2.3 退出循环的方式

1. break终止循环

```

1  str1 = 'itheima'
2  for i in str1:
3      if i == 'e':
4          print('遇到e不打印')
5          break
6      print(i)
7  else:
8      print('循环正常结束之后执行的代码')

```

执行结果：

```

Run: for循环之break x
C:\Users\黑马程序员\AppData\Local\Programs\Python\Python37\python3.exe
i
t
h
遇到e不打印
Process finished with exit code 0

```

没有执行else缩进的代码。

## 2. continue控制循环

```

1  str1 = 'itheima'
2  for i in str1:
3      if i == 'e':
4          print('遇到e不打印')
5          continue
6      print(i)
7  else:
8      print('循环正常结束之后执行的代码')

```

执行结果：

```

Run: for循环之continue x
C:\Users\黑马程序员\AppData\Local\Programs\Python\Python37\python3.exe
i
t
h
遇到e不打印
i
m
a
循环正常结束之后执行的代码
Process finished with exit code 0

```

因为continue是退出当前一次循环，继续下一次循环，所以该循环在continue控制下是可以正常结束的，当循环结束后，则执行了else缩进的代码。

# 总结

---

- 循环的作用：控制代码重复执行
- while语法

```
1 while 条件:
2     条件成立重复执行的代码1
3     条件成立重复执行的代码2
4     .....
```

- while循环嵌套语法

```
1 while 条件1:
2     条件1成立执行的代码
3     .....
4     while 条件2:
5         条件2成立执行的代码
6         .....
```

- for循环语法

```
1 for 临时变量 in 序列:
2     重复执行的代码1
3     重复执行的代码2
4     .....
```

- break退出整个循环
- continue退出本次循环，继续执行下一次重复执行的代码
- else
  - while和for都可以配合else使用
  - else下方缩进的代码含义：当循环正常结束后执行的代码
  - break终止循环不会执行else下方缩进的代码
  - continue退出循环的方式执行else下方缩进的代码