

五、循环语法

目标：

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

5.1 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。

5.1.2 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1
4     ...
```

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

```
1 i = 1
2 while i<=5:
3     print('媳妇我错了')
4     i += 1
5 print('任务结束')
```

计数器习惯书写

```
1 i = 0
2 while i<5:
3     print('媳妇我错了')
4     i += 1
5 print('任务结束')
```

解释器如何执行：

```
1 # 采用Debug每步每步执行
2 i = 0
3 while i<5:
4     print('媳妇我错了')
5     i += 1
6 print('原谅你了')
```

while应用一:

```
1  """
2  分析: 1-100的累加和, 即1+2+3+4+即前两个数字的相加结果+下一个数字 (前一个数字+1)
3  """
4  i = 0
5  sum = 0
6  while i<100:
7      sum +=i
8      i += 1
9
10 print(f'sum={sum}')
```

while应用二:

```
1  """
2  分析: 1-100的偶数和, 即2+4+6+8得到偶数的方法如下:
3  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
4  初始值为0/2, 计数器每次累加2
5
6  """
```

```
1  # 方法一: 条件判断和2取余数则累加
2  i = 1
3  result = 0
4  while i <= 100:
5      if i % 2 == 0:
6          result += i
7      i += 1
8  print('result=%d' % result)
```

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

5.1.2 while循环嵌套

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

```

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

```

快速体现：

```

1 j = 0
2 while j<3:
3     i =0
4     while i<3:
5         print('媳妇我错了')
6         i+=1
7     print('刷碗')

```

5.1.3 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。

break：当某些条件成立，退出当前循环

```

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

```

continue：当某些条件成立时候，退出当前循环，继而执行下一次循环

```

1 i = 1
2 while i<=5:
3     if == 3:
4         print('有虫子,这个苹果不说了')
5         # 如果使用continue, 在continue之前一定要修改计数器, 否则进入死循环
6         i += 1
7         continue
8     print(f'吃了第{i}个苹果')
9     i += 1

```

5.1.4 while循环嵌套应用

应用一：

```

1 j = 0
2 while j<5:
3     # 一行星星开始
4     i = 0
5     while i<5:
6         print('*', end='')
7         i += 1
8     # 一行星星结束
9     print() # 默认换行
10    j+=1

```

应用二：

```

1 # 三角形：每行星星的个数和行号数相等
2 j = 0
3 while j<5:
4     # 一行星星开始
5     i = 0
6     # 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
7     while i<=j:
8         print('*', end='')
9         i += 1
10    # 一行星星结束
11    print() # 默认换行
12    j+=1

```

应用三：

```

1 #重复打印9行乘法表达式
2 """
3 1、打印一个乘法表达式：x * x = x*x
4 2、一行打印多个表达式--一行表达式的个数和行数相等 -- 循环：一个表达式，不换行
5
6 """
7 j = 1
8 while j<=9:
9     #一行的表达式开始
10    i = 1
11    while i<=j:
12        print(f'{i} * {j} = {i*j}',end='\t')
13        i += 1
14    #一行表达时的结束
15    print()
16    j += 1
17

```

5.1.5 for循环

```

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     .....

```

```

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

```

1、break退出for循环

```

1 str1 = 'itheima'
2 for i in str1:
3     if i == 'e':
4         break
5     print(i) # i t h

```

2、continue退出for循环

```

1 str1 = 'itheima'
2 for i in str1:
3     if i == 'e':
4         continue
5     print(i) # i t h i m a

```

5.2 while...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。

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('媳妇我错了')
4     i += 1
5 print('媳妇原谅我了')

```

while...else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

```

```

1 i = 1
2 while i <= 5:
3     print('媳妇我错了')
4     i += 1
5 else:
6     print('原谅我了')
7

```

5.2.1 while...else之break和continue

break:

```

1 i = 1
2 while i<5:
3     if i==3:
4         print('这边说的不真诚')
5         break
6     print('媳妇我错了')
7     i += 1
8 else:
9     print('媳妇原谅我了')

```

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。

continue:

```

1 i = 1
2 while i<5:
3     if i==3:
4         i +=1
5         print('这边说的不真诚')
6         continue
7     print('媳妇我错了')
8     i += 1
9 else:
10    print('媳妇原谅我了')

```

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。

5.3 for...else

```

1 for 临时变量 i 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

```

```

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")

```

5.3.1 for...else之break和continue

break:

```
1 str1 = 'itheima'
2 for i in str1:
3     if i == 'e':
4         break
5         #continue
6     print(i)
7 else:
8     print('循环正常结束执行的else代码')
```

continue:

```
1 str1 = 'itheima'
2 for i in str1:
3     if i == 'e':
4         #break
5         continue
6     print(i)
7 else:
8     print('循环正常结束执行的else代码')
```