

第5~6课 for循环与while循环

一、课程结构导图

风变科技

风变科技

风变科技

风变科技

风变科技

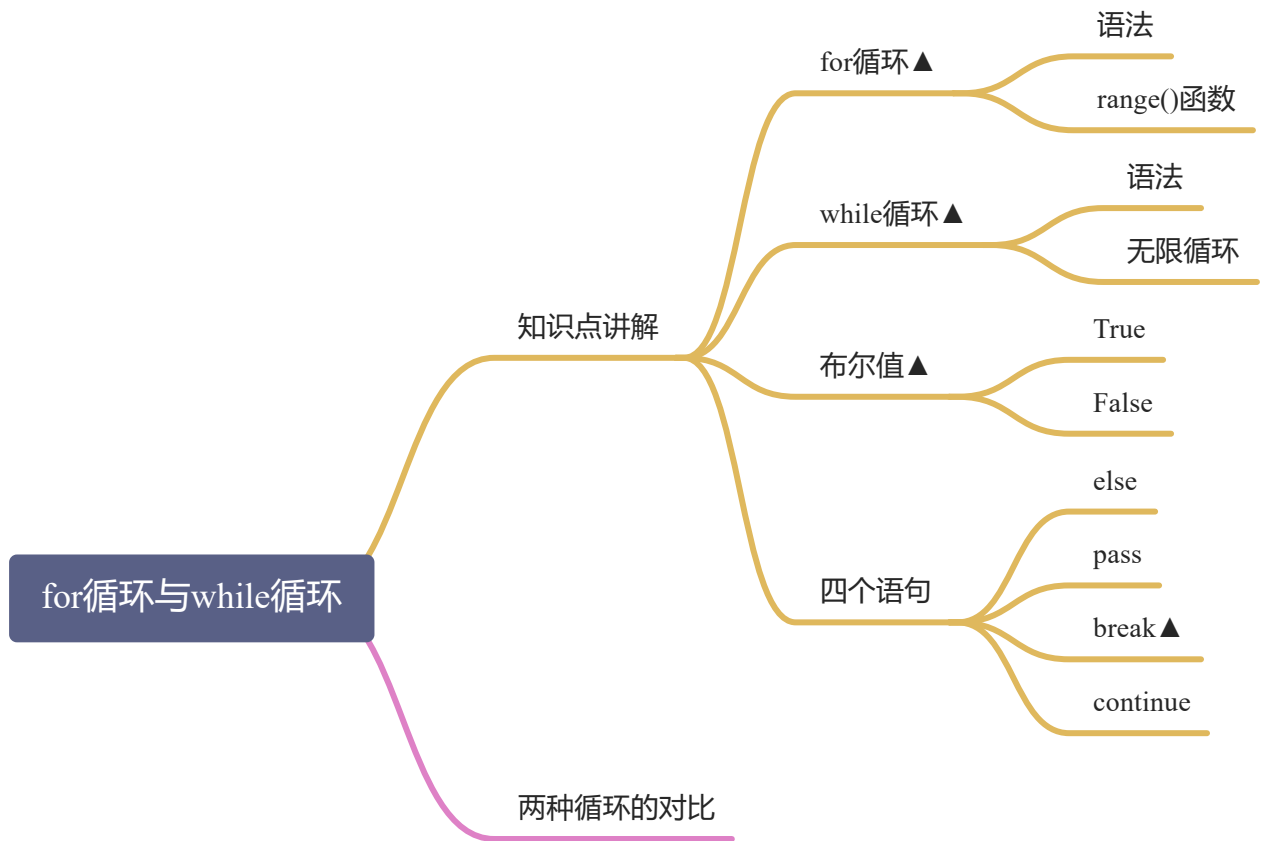
风变科技

风变科技

风变科技

风变科技

风变科技



注：▲为重点知识点。

二、知识点讲解

2.1 for循环

2.1.1 语法

概念： for..in循环是循环语句中的一种，它可以迭代使用一个序列中的每个元素。

用法1： 使用for...in...循环遍历列表或元组中的所有元素。

示例1：

```
1 a = [1,2,3,4,5,6,7,8,9]
2 for i in a:
3     print(i) # 打印列表中的每一个元素
4
5 b = (1,2,3,4,5,6,7,8,9)
6 for i in b:
7     print(i) # 打印元组中的每一个元素
```

用法2：使用for...in...循环遍历字典中所有的键。

示例2：

```
1 info = {'name':'Bob','age':18,'email':'xxx@gmail.com'}
2
3 for i in info:      #每循环一次, i获得info字典中最新的键
4     print('key:',i,',value:',info[i])
```

用法3：使用for...in...循环遍历字符串，会依次输出单个字符。

示例3：

```
1 for i in 'hello,world':
2     print(i) # 打印出字符串'hello,world'中的每个字符
```

2.1.2 range()函数

概念：range()函数可以生成一个整数序列，通常和for...in...循环搭配使用。

用法1：range()函数内含有一个参数，表示从0开始生成连续整数序列的数量。

示例1：

```
1 for i in range(10):
2     print(i) # 打印出整数0-9
```

用法2：range()函数内含有三个参数：range(a,b,c)，前两个参数分别代表生成整数序列的起止（遵循左取右不取的原则）。第三个参数代表步长（可选参数），即数字之间的间隔。

示例2：

```
1 for i in range(1,11):
2     print(i)
3     # 打印出整数1-10
4
5 for i in range(1,11,3):
6     print(i)
7     # 打印数字1, 4, 7, 10
```

2.2 while循环

2.2.1 语法

概念：while循环是循环语句的另外一种。只要在判断条件成立的情况下，while语句就会执行。

用法：与if语句相似，判断while关键字后面的条件是否成立，如果成立，就会进入循环之中，当不成立的时候就会结束循环。

示例：

```
1 number = 0
2 while number <= 100:
3     print('The number is:',number)
4     number += 20 # 和number=number+20一样，表示每循环一次数值加20
```

```
5
6 print('The loop is over!')
```

2.2.2 无限循环

概念：只要判断条件一直成立，循环语句就会一直执行。

用法：使while关键字后的判断条件保持成立，就可以让循环一直进行下去，必要的时候在循环体内使判断条件不成立，循环就会结束。

示例：

```
1 status = 1
2
3 while status == 1:
4     print('循环进行中')
5     judge = input('是否要结束循环:')
6     if judge == '是':
7         status = 0
```

代码解析：status变量表示判断是否继续循环的变量，设定初始值为1。进入循环后，第5行代码如果输入“是”的话，status变量就会变成0，循环结束，如果输入的是其他的字符串，则循环会一直进行下去。

2.3 布尔值

概念：布尔值分为True和False。在Python中有很多表示布尔值的方式，布尔常量False、0、Python零值None、空值（如空列表[]和空字符串""），都被视为False。布尔常量True和其他一切值都被视为True。¹

用法：通过比较操作符（<、<=、>、>=、!=、is、is not、in、not in）和逻辑操作符（and、not、or），可以创建返回True和False的比较表达式。

示例:

```
1 # 使用比较操作符
2 print(3 < 5)    # 打印True
3 print(3 > 5)    # 打印False
4 print('长安' == '长安')    # 打印True
5
6 # 使用逻辑操作符
7 a = 1
8 b = -1
9 print('以下是and运算')
10 if a == 1 and b == 1:    # 【b实际上是-1】
11     print('True')
12 else:
13     print('False')    # 因为and右边的布尔表达式不成立，所以11行的if条件为不成立，打印False
14
15 print('以下是or运算')
16 if a == 1 or b == 1:    # 【b实际上是-1】
17     print('True')    # or两边的条件只要有一个成立，if条件就成立，打印True
18 else:
19     print('False')
20
21 # 使用in和not in操作符
22 list = [1,2,3,4,5]
23 a = 1
24 # 做一次布尔运算，判断“a是否在列表list之中”
25 print(bool(a in list))    # 满足in的条件，打印True
26 print(bool(a not in list))    # 不满足not in的条件，打印False
```

2.4 四个语句

2.4.1 pass语句

概念：pass语句表示什么也不做。当语法上需要一个语句，但程序需要什么动作也不做时，可以使用它。

2

用法：当需要写一个条件判断语句时，如果if语句里不需要做出任何动作，就必须使用pass语句去占位。

示例：

```
1 a = int(input('请输入一个整数:'))
2 if a > 100:
3     pass
4 else:
5     print('你输入了一个小于100的数字')
```

2.4.2 break语句

概念：break语句用以中断循环语句，也就是中止循环语句的执行，即使循环条件没有变更为 False，或序列中的项目尚未完全迭代依旧如此。³

用法：在循环中使用if语句来判断是否要跳出循环，如果需要跳出循环，就在if语句里使用break语句。

示例：

```
1 while True:
2     s = input('请输入一句话: ')
3     if s == '退出':
4         break
5     print('这句话的长度为: ', len(s))
6 print('程序结束!')
```

代码解析：首先执行第2行的代码提示输入一句话并且赋值给变量s，然后执行第3行的判断条件，如果输入的不是"退出"，就会继续执行第5行的代码统计并打印输入内容的长度，len()函数可以统计字符串的长度。如果输入的是"退出"，就会执行第4行的break语句，并跳出while循环，打印"程序结束"。

2.4.3 continue语句

概念：continue语句被用来告诉程序跳过本次循环中的剩余内容，直接进行下一轮循环。

用法：在循环中使用if语句来判断是否要继续执行本次循环，如果不需要继续执行本次循环，可以使用continue语句跳过当前循环，并开始下一次循环。

示例：

```
1 while True:
2     s = input('请输入一句话：')
3     if s == '再见啦':
4         break
5     if len(s) < 3:
6         print('输入的内容至少要有三个字哟！')
7         continue
```

代码解析：第2行我们需要输入一句话并且赋值给变量s，如果这句话的内容是“再见啦”，就跳出循环，结束程序。如果不是“再见啦”就执行第5行的判断条件，判断这句话的长度是否小于3，如果小于3就提示输入的内容太短，并跳过本次循环继续进入下一次循环。

2.4.4 else语句

用法：else不但可以和if配合使用，它还能跟for循环和while循环配合使用。

示例：

```
1 # for循环与else语句配合使用
2 for i in range(5):
3     a = int(input('请输入0结束循环，你有5次机会：'))
4     if a == 0:
5         print('你触发了break语句，导致else语句不会生效。')
6         break
7 else:
8     print('5次循环你都错过了，else语句生效了。')
9
10 # while循环与else语句配合使用
```



```
11 i = 0
12 while i<5:
13     a = int(input('请输入0结束循环, 你有5次机会:'))
14     i = i+1
15     if a == 0:
16         print('你触发了break语句, 导致else语句不会生效。')
17         break
18 else:
19     print('5次循环你都错过了, else语句生效了。')
```

注意：若在执行5次循环的过程中，都没有触发到break语句，则会执行else语句，反之，只要有一次循环触发break语句，则else语句不会被执行。

三、两种循环的对比

用法：两种循环应用场景的不同主要体现在循环次数是否确定上。无论是遍历列表、元组、字典还是range()函数生成的序列，for循环通常处理循环次数确定的工作。而while循环通常处理循环次数不确定的工作。

示例：

```
1 # 循环次数确定
2 for i in '神雕侠侣':
3     print(i)
4
5 # 循环次数不确定
6 password = ''
7 while password != '816':
8     password = input('请尝试输入密码: ')
```

四、巩固练习

1. (单项选择题) 查看下列代码, 请问第三次循环的时候, 打印的结果是什么?

```
1 d = {'小十': '绿皮火车', '十一': '高铁一等座', '十二': '高铁二等座', '十三': '长途客车'}
2 for i in d:
3     print(i + '回家坐' + d[i])
```

- A、小十回家坐绿皮火车 B、十一回家坐高铁一等座
C、十二回家坐高铁二等座 D、十三回家坐长途客车

2. (单项选择题) 查看下列代码, 请问终端中最后打印的数字是多少?

```
1 count = 0
2 while count < 10:
3     if count == 8:
4         continue
5     print(count)
6     count += 1
```

- A、7 B、8 C、9 D、10

3. (多项选择题) 请问下面这段代码会打印出哪些结果?

```
1 a = [1, 2, 3]
2 b = [4, 5, 6]
3 for i in a:
4     for j in b:
5         print('%s VS %s' % (i, j))
```

- A、2 VS 4 B、5 VS 2
C、4 VS 3 D、3 VS 5

4. (实操题) 假如投资的年利率为5%, 试求从1000块增长到5000块, 需要花费多少年?

参考文献：

1. Naomi Ceder 著，戴旭译，《Python快速入门》（The Quick Python Book），第24页。
2. Python官方教程，4.5 pass语句，链接：<https://docs.python.org/zh-cn/3/tutorial/controlflow.html#<https://docs.python.org/zh-cn/3/tutorial/controlflow.html#>>
3. Swaroop C.H 著，漠伦译，《简明python编程》（A Byte of Python），第56页。