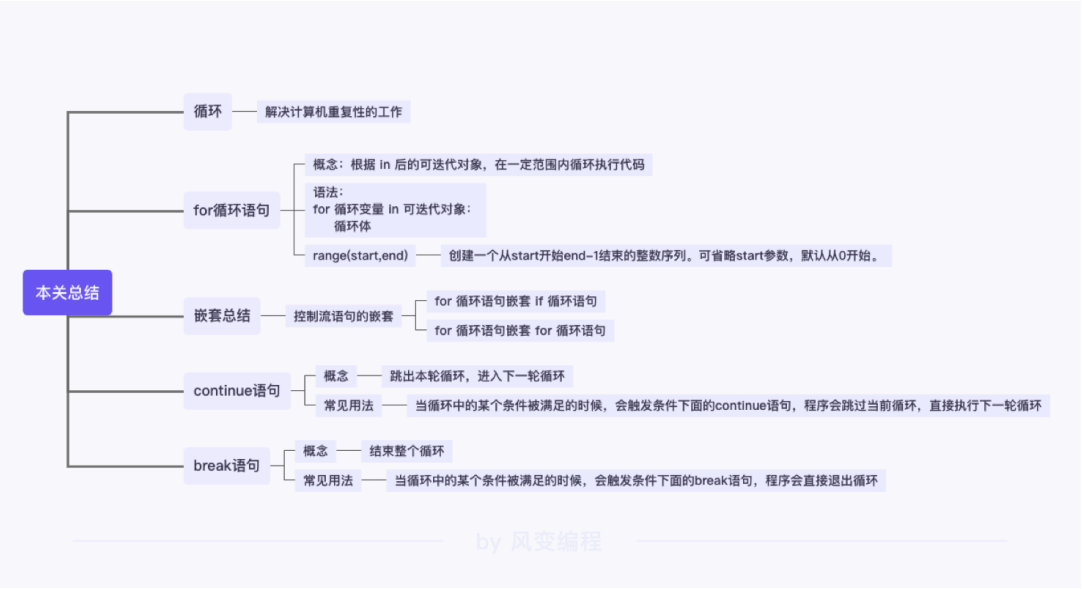


基础语法第5关笔记

主线课程：



一、for循环语句

1.1 for循环的概念

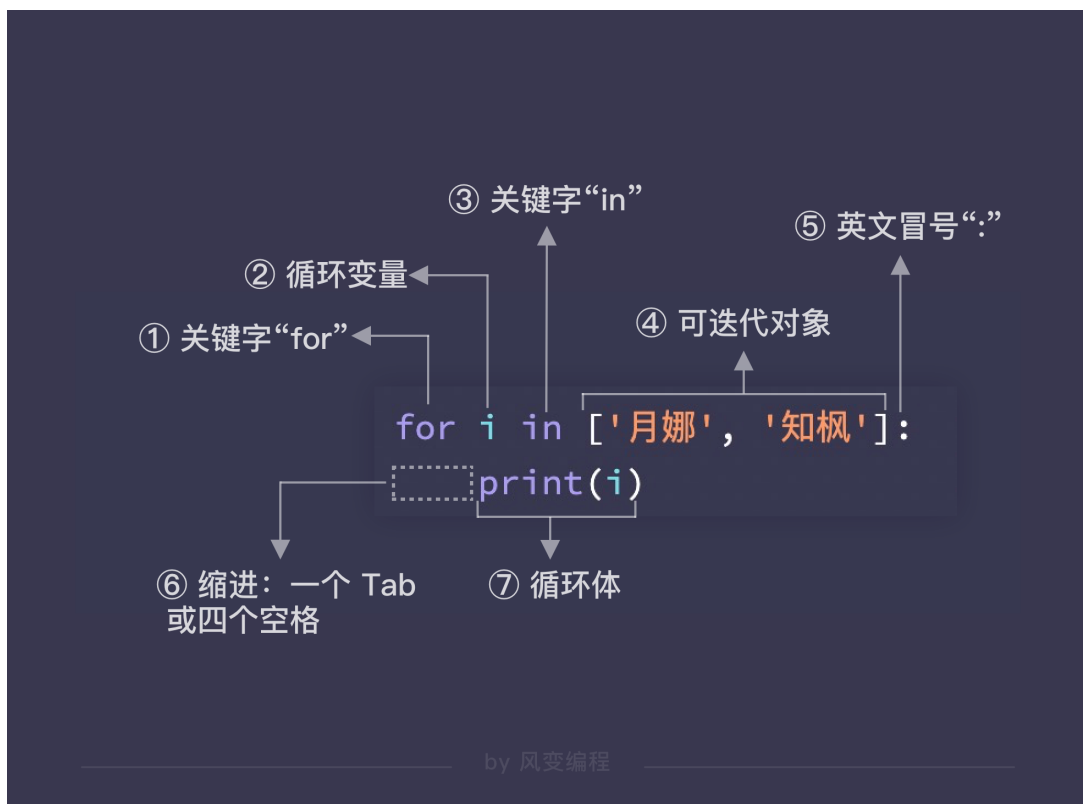
循环是反复执行某些代码的一种处理过程，可以完成具有规律性的重复操作。

for 循环语句就是通过重复执行循环体来实现循环，并能简省代码量。

for 循环语句根据 in 后的可迭代对象，在一定范围内循环执行代码

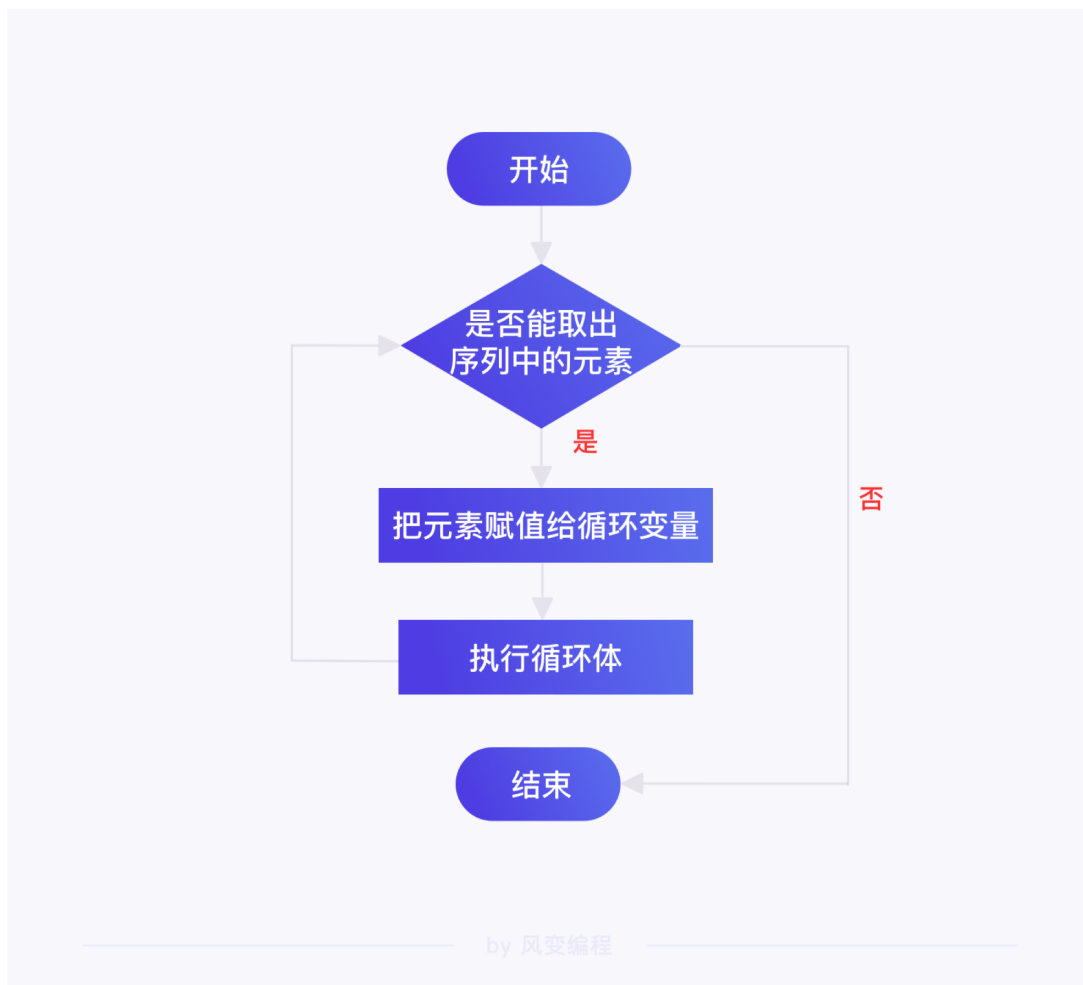
1.2 组成要素

for 循环语句包含了七个要素：① 关键字“for”；② 循环变量；③ 关键字“in”；④ 可迭代对象；⑤ 英文冒号“:”；⑥ 缩进；⑦ 循环体。



其中要素① 和要素③是固定写法。其次要素④可迭代对象可以理解为一个“盒子”，可以被程序一次一次地取出数据。就像我们学过的序列，即字符串、元组和列表都可以叫做可迭代对象。

1.3 执行流程



二、 嵌套总结

嵌套的概念及类型

根据我们学到的几个知识点，我们可以把嵌套归纳为以下几点：

- 1 一个简单个体和另外一个简单个体组合在一起，变成了一个复合体。
- 2 复合体又可以和别的个体组合，形成新的复合体。
- 3 最重要的是，无论简单个体还是复合体，我们都可以把他看成是一个个体。
- 4 我们常说的嵌套，就是复合体里有复合体。

by 风变编程

一般常见的的嵌套类型

- (1) 数据类型嵌套（列表、元组）
- (2) 条件判断嵌套
- (3) 条件判断语句的多层嵌套。
- (4) 循环语句嵌套

三、 for 循环语句的嵌套

3.1 for循环语句嵌套条件判断语句

for 循环语句下的代码块，即循环体，也可以是一个条件判断语句。
举个例子：

```
1 number_list = [12, 62, 33, 47, 59]
2 num = int(input('请输入一个数值: '))
3 for n in number_list:
4     if n > num:
5         print('{}大于{}'.format(n, num))
6 # 12大于9
7 # 62大于9
8 # 33大于9
9 # 47大于9
10 # 59大于9
```

3.2 for 循环语句嵌套 for 循环语句

进入大循环体后，程序执行循环体内的 **小for** 循环语句，当循环体内的小for语句循环结束后才会再次回到大循环体进行下一步的循环。

```
pen_bag = ['笔', '尺子', '涂改液']
computer_bag = ['电脑', '鼠标']
bag_list = [pen_bag, computer_bag]

for bag in bag_list:
    for item in bag:
        print(item)
```

循环体

概括

```
pen_bag = ['笔', '尺子', '涂改液']
computer_bag = ['电脑', '鼠标']
bag_list = [pen_bag, computer_bag]

for bag in bag_list:
    (循环体)
```

by 风变编程

举个例子：

```
1 pen_bag = ['笔', '尺子', '涂改液']
2 computer_bag = ['电脑', '鼠标']
3 bag_list = [pen_bag, computer_bag]
4 for bag in bag_list:
5     for item in bag:
6         print(item)
7
8 # 笔
9
10 # 尺子
11
12 # 涂改液
13
14 # 电脑
15
16 # 鼠标
```

四、循环控制

4.1 continue语句

continue单词的汉语翻译是“继续”，用在循环里，可以理解为跳出本轮循环，进入下一轮循环的意思。

常见的使用方法是：当循环体中的某个条件被满足后，触发该条件下面的continue语句。这时，程序会跳过本轮循环的剩余语句，执行下一轮循环。

<pre>main.py 1 to_do_list = ['酒吧驻唱', '学Python', '看剧'] 2 for i in to_do_list: 3 if i == '看剧': 4 continue 5 print(i)</pre>	<pre>终端 bash:root\$ python /home/python-class/root/main.py 酒吧驻唱 学Python 写论文 睡觉 █</pre>
--	--

4.2 break语句

break可译为“打破，终止”，用在循环里就是打破循环，终止循环的意思。

与continue语句一样，break语句也必须在循环体中使用。

常见的使用方法是：当循环体中的某个条件被满足后，触发该条件下面的break语句。这时，程序会提前结束整个循环。

<pre>main.py 1 to_do_list = ['酒吧驻唱', '学Python', '看剧'] 2 for i in to_do_list: 3 if i == '看剧': 4 break 5 print(i)</pre>	<pre>终端 bash:root\$ python /home/python-class/main.py 酒吧驻唱 学Python </pre>
---	--

支线课程：



一、while循环

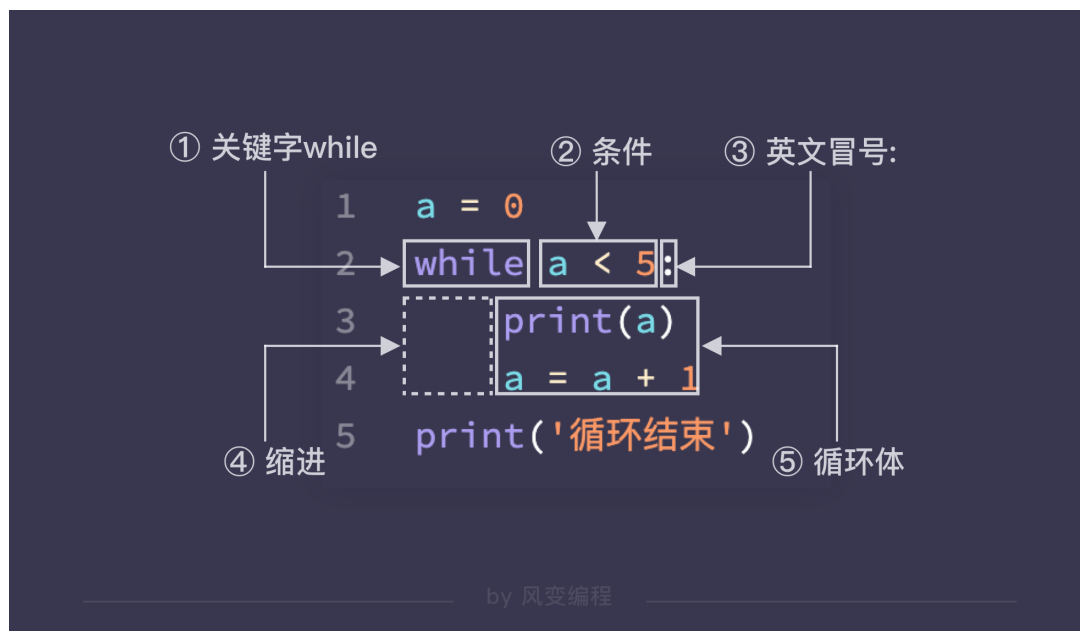
while循环是“当...时候，执行某段程序”形式的循环。

当给定的条件为真时，执行循环体；条件为假时，结束循环。

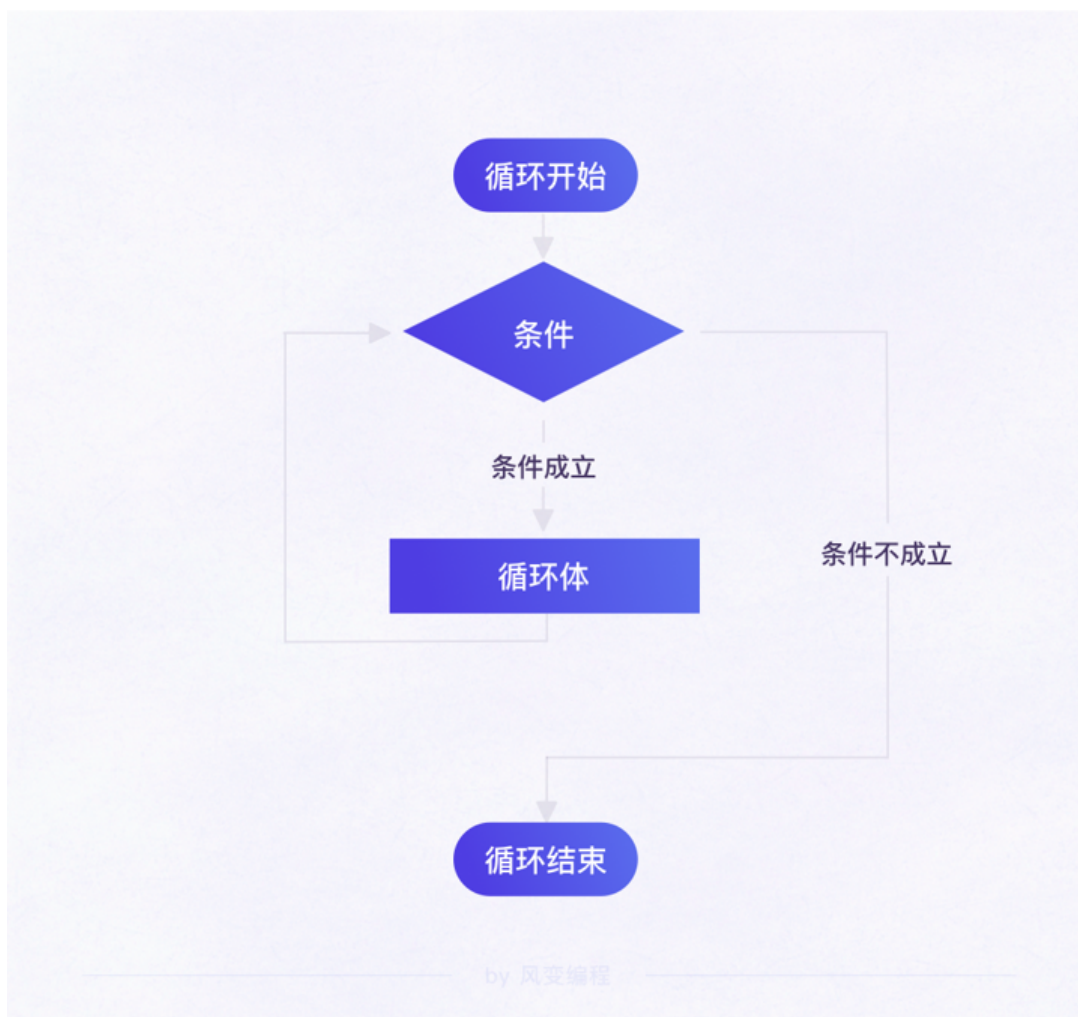
• while循环语句的五要素

while循环语句包含了五个要素：

1. 关键字while；
2. 条件；
3. 英文冒号；
4. 缩进；
5. 循环体（要被重复执行的语句）



• while循环的执行过程



- 无限循环

当while循环中的条件永远成立时，理论上，程序会一直执行循环体中的代码。

选中终端后，按键盘上的ctrl+c可以终止无限循环

二、控制循环

1.通过变量控制循环

过改变变量的值，使条件不成立，这样我们就能退出循环。

- 通过计数的方式改变变量

```
1 a = 0
2 while a < 5:
3     print(a)
4     a = a + 1 #每次循环a的值都会+1，直到a=5就不满足条件了
5 print('循环结束')
```

- 增强赋值符号

普通形式	增强赋值形式
<code>a = a + b</code>	<code>a += b</code>
<code>a = a - b</code>	<code>a -= b</code>
<code>a = a * b</code>	<code>a *= b</code>
<code>a = a / b</code>	<code>a /= b</code>

by 风变编程

• 通过输入的方式改变变量

先设置初始变量，然后在循环体中，借助input()函数来改变变量的值，进而使条件不成立，最后退出循环。

```

1 #定义初始变量doing为空字符串
2 doing = ''
3 #当变量doing不等于'写简历'时，while循环会一直执行
5 while doing != '写简历':
6     #获得用户输入来改变doing的值
8     doing = input("请输入知枫在做什么？")
9     print("知枫在" + doing)
10
11 #当我们输入'写简历'，条件不成立，循环结束，程序执行while循环后的语句，打印
12 print("知枫终于去写简历了!!!")

```

2.continue语句

执行到continue语句，程序会跳过本轮循环的剩余语句，直接回到循环的开头，执行下一轮循环。

<pre> main.py 1 to_do_list = ['酒吧驻唱', '学Python', '看剧'] 2 for i in to_do_list: 3 if i == '看剧': 4 continue 5 print(i) </pre>	<pre> 终端 bash:root\$ python /home/python-class/root/main.py 酒吧驻唱 学Python 写论文 睡觉 [] </pre>
--	---

3.break语句

执行到break语句，程序会提前结束整个循环（即跳出循环），执行while循环后的语句，无需再理会一开始设置的条件。

main.py

```
1 a = 0
2 while a < 5:
3     a += 1
4     if a == 3:
5         break
6     print(a)
7 print('循环结束')
```

终端

```
bash:root$ python /home/python-class/root/
main.py
1
2
循环结束
□
```