

Python语言基础与应用

计算和控制流 / 控制流：条件循环 (while)

陈斌 北京大学 gischen@pku.edu.cn



控制流：条件循环 (while)

- › 循环结构
- › 循环结构的要素
- › 条件循环
- › 嵌套循环

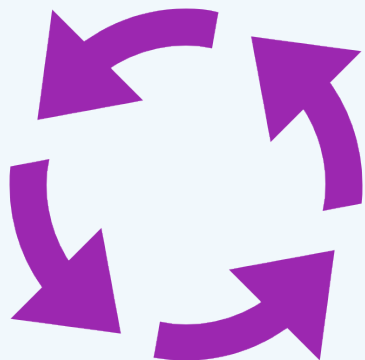
循环结构

› 让计算机执行冗长单调的重复性任务

- 根据需要对一系列操作进行设定次数或者设定条件的重复，这样的控制流程，就称作循环结构

› 作用

- 能持续对大量数据进行处理
- 在长时间里对一些未知状况进行连续监测循环结构

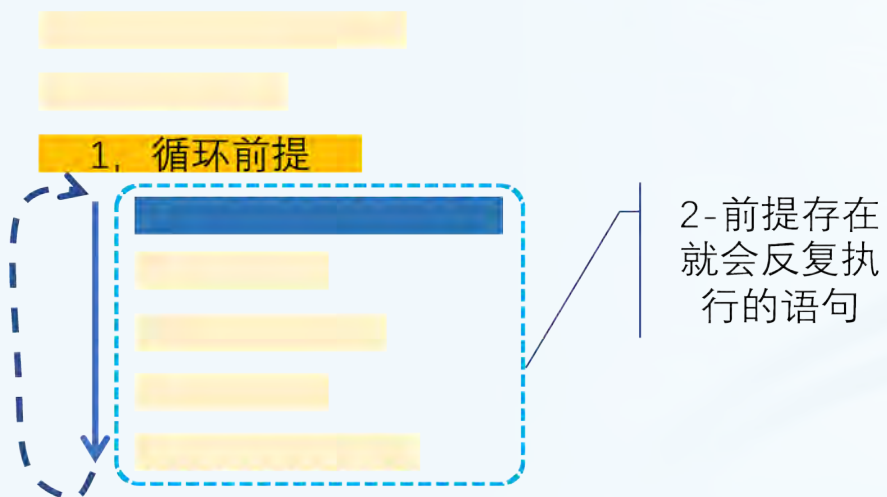


循环结构的要素

基本要素

循环前提和执行语句

- 在执行这组语句之前，计算机检查循环前提是否存在，只要存在，就会反复执行这组语句，直到循环前提消失



循环结构的两个基本要素

循环结构的要素

› 循环前提的类型

- 从某个容器或者潜在的数据集中逐一获取数据项，什么时候取不到数据项了，循环的前提就消失
- 只要逻辑表达式计算结果为真(True)，循环的前提就存在，什么时候逻辑表达式计算结果为假(False)，循环的前提就消失了

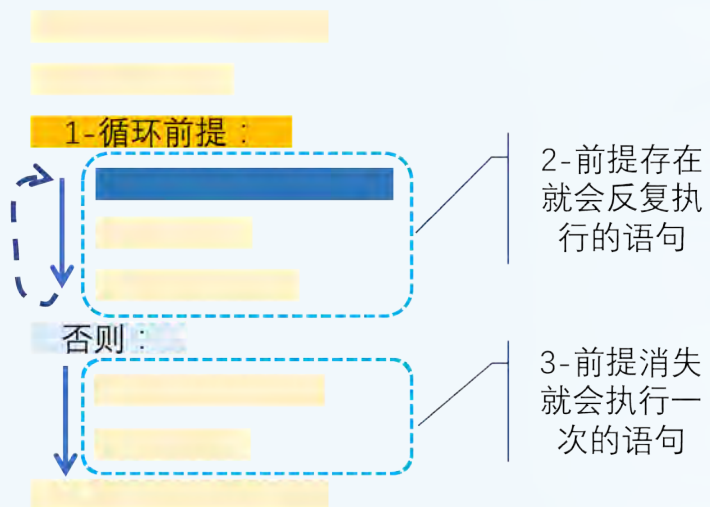
› 与条件分支结构的区别

- 循环结构会多次检查循环前提

循环结构的要素

› 扩展要素

- 当循环前提消失，停止执行这组语句的时候，执行一次另一组语句



循环结构扩展要素

条件循环

› while循环语法

while <逻辑表达式>:

 <语句块>

break #跳出循环

continue #略过余下循环语句

 <语句块>

else: #条件不满足退出循环，则执行

 <语句块>

- **while**、**else:** 保留字
- 逻辑表达式：指所有运算的结果为逻辑类型（True或False）的表达式
- 语句块：一组语句

条件循环

```
>>> n = 5
>>> while n > 0:
    n = n - 1
    if n < 2:
        break
    print (n)
```

4
3
2

条件循环

```
>>> n = 5
>>> while n > 0:
    n = n - 1
    if n < 2:
        continue
    print (n)
else:
    print ('END!')
```

```
4
3
2
END!
```

嵌套循环

› 双重甚至多重的循环

```
for i in range(1, 10000):  
    n = i  
    step = 0  
    while n != 1:  
        if n % 2 == 0:  
            n = n // 2  
        else:  
            n = n * 3 + 1  
            step += 1  
    else:  
        print(i, "Traped!", step, "steps")  
1167 Traped! 31 steps  
1168 Traped! 119 steps  
1169 Traped! 57 steps  
1170 Traped! 57 steps  
1171 Traped! 57 steps  
1172 Traped! 119 steps  
1173 Traped! 119 steps  
1174 Traped! 119 steps  
1175 Traped! 119 steps  
1176 Traped! 119 steps  
1177 Traped! 31 steps  
1178 Traped! 119 steps
```

- 中断程序运行：CTRL+C
- 通过修改程序中range函数的参数，还可以验证其它范围的连续整数