



Python

快速编程入门

第3版

黑马程序员 • 编著

黑马程序员系列教材改版升级

本书配套丰富的教学资源，包括教学 PPT、教学大纲、教学设计、源代码、课后习题及答案等

本书从初学者的角度出发，采用“理论 + 实训案例 + 阶段案例 + 综合案例”的模式详细讲解 Python 3 的相关知识



中国工信出版集团



人民邮电出版社
POSTS & TELECOM PRESS

3.3 循环语句

在某些情况下，程序可能需要重复执行某个操作。例如，当用户登录平台时，因为其长时间未登录可能会忘记密码，所以需要反复输入账号和密码，直到达到平台设定的输入次数限制或成功登录为止。为了满足这种需求，Python 提供了循环语句，使用该语句能以简洁的代码重复执行某个操作。Python 中的循环语句包括 while 语句和 for 语句。本节将针对循环语句进行详细讲解。

3.3.1 while 语句

while 语句一般用于实现条件循环，该语句由关键字 while、循环条件和英文冒号组成，while 语句和从属于该语句的代码段组成循环结构，其语法格式如下：

```
while 循环条件:  
    代码段
```

以上语法格式中的 while 关键字和英文冒号分别标识 while 语句的起始和结束，循环条件与 while 关键字以空格分隔，代码段通过缩进与 while 语句产生关联。

执行 while 语句时，若循环条件的值为 True，则执行之后的代码段，执行完代码段之后再次判断循环条件，如此往复，直至循环条件的值为 False 时循环终止，执行循环之后的代码。while 语句的执行流程如图 3-5 所示。

接下来，使用 while 语句计算 $1+2+3+\cdots+10$ 的结果，示例代码如下：

```
i = 1          # 保存要计算的数字，初始值为 1  
result = 0     # 保存累加的结果，初始值为 0  
while i <= 10: # 使用 while 语句实现 1~10 的累加  
    result += i  
    i += 1  
print(result)  # 输出累加后的结果
```

以上示例代码中变量 i 是循环因子，其初始值为 1，

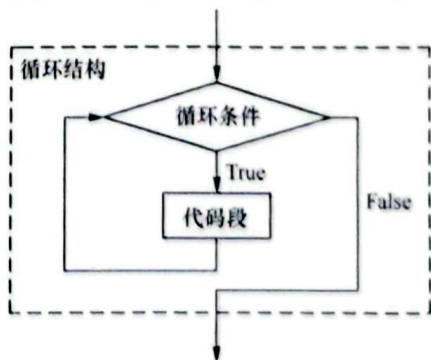


图3-5 while语句的执行流程

会随循环次数累加; 变量 `result` 用于保存累加的结果, 其初始值为 0。首次执行 `while` 语句时, 因为 `i <= 10` 的值为 `True`, 所以会执行循环中的代码段, 使得 `result` 的值由 0 变为 1、`i` 的值由 1 变为 2; 再次判断 `i <= 10`, 如此往复, 直至 `i` 的值变为 11 时, `i <= 10` 的值变为 `False`, 循环结束, 执行循环之后的输出语句, 输出 `result` 的值。

运行代码, 结果如下所示:

```
55
```

若希望程序可以一直重复执行某个操作, 则可以将循环条件的值设为 `True`, 如此便进入无限循环。下面是无限循环的示例代码:

```
while True:
    print("我是无限循环")
```

以上示例代码执行后会在控制台中一直输出“我是无限循环”。若希望程序能够停止输出, 需要通过单击终止运行按钮或其他方式手动终止程序。

需要注意的是, 虽然在实际开发中有些程序需要无限循环, 比如游戏的主程序、操作系统中的监控程序等, 但无限循环会占用大量内存, 影响程序和系统的性能, 开发人员需酌情使用。

3.3.2 for 语句

`for` 语句一般用于实现遍历循环。遍历指逐一访问目标对象中的元素, 例如逐个访问字符串中的字符; 遍历循环指在循环中完成对目标对象的遍历。`for` 语句的语法格式如下:

```
for 临时变量 in 目标对象:
    代码段
```

以上语法格式中的目标对象可以是字符串、文件或后续将会学习的其他组合类型数据; 临时变量用于保存每次循环访问的目标对象中的元素。目标对象的元素个数决定了循环的次数, 目标对象中的元素被访问完之后循环结束。

下面使用 `for` 语句遍历字符串“Python”的每个字符, 示例代码如下:

```
for word in "Python":
    print(word)
```

运行代码, 结果如下所示:

```
P
Y
t
h
o
n
```

`for` 语句常与 `range()` 函数搭配使用, 以控制循环中代码段的执行次数。`range()` 函数中若只有一个整数 `n`, 则会生成一组 `0~n-1` 的整数; 若有两个整数 `m` 和 `n`, 则会生成一组 `m~n-1` 的整数。示例代码如下:

```
for i in range(5):
    print(i)
```

运行代码, 结果如下所示:

```
0
1
2
3
4
```