第三章 第二节 Python基础篇—循环结构

Python While 循环

Python 循环

Python 有两个原始的循环命令：

- while 循环
- for 循环

while 循环

如果使用 while 循环，只要条件为真，我们就可以执行一组语句。

实例

只要 i 小于 7，打印 i：

```
i = 1
while i < 7:
    print(i) i += 1
```

```
1
2
3
4
5
6
```

第三章 第二节 Python基础篇—循环结构

Python While 循环

注释：请记得递增 i，否则循环会永远继续。

while 循环需要准备好相关的变量。在这个实例中，我们需要定义一个索引变量 i，我们将其设置为 1。

break 语句

如果使用 break 语句，即使 while 条件为真，我们也可以停止循环：

实例

在 i 等于 3 时退出循环：

```
i = 1
while i < 7:
    print(i)
    if i == 3:
        break
    i += 1
```

```
1
2
3
```

第三章 第二节 Python基础篇—循环结构

Python While 循环

continue 语句

如果使用 continue 语句，我们可以停止当前的迭代，并继续下一个：

实例

如果 i 等于 3，则继续下一个迭代：

```
i = 0
while i < 7:
    i += 1
    if i == 3:
        continue
    print(i)
```

```
1
2
4
5
6
7
```

第三章 第二节 Python基础篇—循环结构

Python While 循环

else 语句

通过使用 else 语句，当条件不再成立时，我们可以运行一次代码块：

实例

条件为假时打印一条消息：

```
i = 1
while i < 6:
    print(i)
    i += 1
else:
    print("i is no longer less than 6")
```

```
1
2
3
4
5
i is no longer less than 6
```

第三章 第二节 Python基础篇—循环结构

Python While 循环

Python For 循环

for 循环用于迭代序列（即列表，元组，字典，集合或字符串）。

这与其他编程语言中的 for 关键字不太相似，而是更像其他面向对象编程语言中的迭代器方法。

通过使用 for 循环，我们可以为列表、元组、集合中的每个项目等执行一组语句。

实例

打印 fruits 列表中的每种水果：

```
fruits = ["apple", "banana", "cherry"]
for x in fruits:
    print(x)
```

```
apple
banana
cherry
```

提示： for 循环不需要预先设置索引变量

第三章 第二节 Python基础篇—循环结构

Python While 循环

Python For 循环

循环遍历字符串

甚至连字符串都是可迭代的对象，它们包含一系列的字符：

实例

循环遍历单词 "banana" 中的字母：

```
for x in "banana":
    print(x)
```

```
b
a
n
a
n
a
```

第三章 第二节 Python基础篇—循环结构

Python While 循环

Python For 循环

break 语句

通过使用 break 语句，我们可以在循环遍历所有项目之前停止循环：

实例

如果 x 是 "banana"，则退出循环：

```
fruits = ["apple", "banana", "cherry"]
for x in fruits:
    print(x)
    if x == "banana":
    break
```

```
apple
banana
```

第三章 第二节 Python基础篇—循环结构

Python While 循环

Python For 循环

continue 语句

通过使用 continue 语句，我们可以停止循环的当前迭代，并继续下一个：

实例

不打印香蕉：

```
fruits = ["apple", "banana", "cherry"]
for x in fruits:
if x == "banana":
    continue
print(x)
```

```
apple
cherry
```

第三章 第二节 Python基础篇—循环结构

Python While 循环

Python For 循环

range() 函数

如需循环一组代码指定的次数，我们可以使用 range() 函数，
range() 函数返回一个数字序列，默认情况下从 0 开始，并递增 1（默认地），并以指定的数字结束。

实例

使用 range() 函数：

```
for x in range(10):
        print(x)
```

```
0
1
2
3
4
5
6
7
8
9
```

第三章 第二节 Python基础篇—循环结构

Python While 循环

Python For 循环

range() 函数

注意： `range(10)` 不是 0 到 10 的值，而是值 0 到 9。

`range()` 函数默认 0 为起始值，不过可以通过添加参数来指定起始值：`range(3, 10)`，这意味着值为 3 到 10（但不包括 10）：

实例

使用起始参数：

```
for x in range(3, 10):
          print(x)
```

```
3
4
5
6
7
8
9
```

第三章 第二节 Python基础篇—循环结构

Python While 循环

Python For 循环

range() 函数

range() 函数默认将序列递增 1，但是可以通过添加第三个参数来指定增量值：range(2, 30, 3)：

实例

使用 3 递增序列（默认值为 1）：

```
for x in range(3, 50, 6):
          print(x)
```

```
3
9
15
21
27
33
39
45
```

第三章 第二节 Python基础篇—循环结构

Python While 循环

Python For 循环

For 循环中的 Else

for 循环中的 else 关键字指定循环结束时要执行的代码块：

实例

打印 0 到 9 的所有数字，并在循环结束时打印一条消息：

```
for x in range(10):
        print(x)
else:
        print("Finally finished!")
```

```
0
1
2
3
4
5
6
7
8
9
Finally finished!
```

第三章 第二节 Python基础篇—循环结构

Python While 循环

Python For 循环

嵌套循环

嵌套循环是循环内的循环。

“外循环”每迭代一次，“内循环”将执行一次：

实例

打印每个水果的每个形容词：

```
adj = ["red", "big", "tasty"]
fruits = ["apple", "banana", "cherry"]
for x in adj:
          for y in fruits: print(x, y)
```

```
red apple
red banana
red cherry
big apple
big banana
big cherry
tasty apple
tasty banana
tasty cherry
```

第三章 第二节 Python基础篇—循环结构

Python While 循环

Python For 循环

pass 语句

for 语句不能为空，但是如果您处于某种原因写了无内容的 for 语句，请使用 pass 语句来避免错误。

实例

```
for x in [0, 1, 2]:
        pass
```