

量化金融 python 基础 3-Loop

2023 年 5 月 15 日

0.1 Loop 循环

python 主要提供了两种循环结构：for 循环和 while 循环

For 循环是一种用于迭代遍历数据结构（如列表、元组、字符串等）的循环结构。

For 循环可以按顺序遍历一个序列中的每一个元素，并且在遍历时可以进行一些操作。

for 语句经常和列表 List 搭配使用。

```
[1]: # 打印列表中的每个元素。
fruits = ["apple", "banana", "cherry", "grape", "pear"]
for fruit in fruits:
    print(fruit)
```

```
apple
banana
cherry
grape
pear
```

```
[2]: # for loop with range for 循环和函数 range() 搭配使用
for i in range(5):
    print(i)
```

```
0
1
2
3
4
```

```
[3]: # range() 加入 step 步长
# range(start, end, step)
# step 代表每隔 X 步
for i in range(1, 6, 2):
    print(i)
```

1
3
5

```
[4]: # 按照从 0 开始的顺序打印索引和 fruits
for i in range(len(fruits)):
    print(i, fruits[i])
```

0 apple
1 banana
2 cherry
3 grape
4 pear

```
[5]: # 以上功能也可以通过 python 内建函数 enumerate 实现
# enumerate 返回迭代过程中索引和对应的值
for i, j in enumerate(fruits):
    print(i, j)
# 通常, 如果同时要用到索引和列表值, 用 enumerate 会比 range(len()) 方便更快一些。
```

0 apple
1 banana
2 cherry
3 grape
4 pear

```
[6]: # zip() 函数可以实现列表、元组或者其他序列的元素配对。
# 生成从 0 到 5 的列表
one2five_list = list(range(1,6))
zipped = zip(one2five_list, fruits)
```

```
[7]: # zipped 是无法直接查看的, zipped 是一个迭代器,
# 需要先转化成 list
```

```
list(zipped)
```

```
[7]: [(1, 'apple'), (2, 'banana'), (3, 'cherry'), (4, 'grape'), (5, 'pear')]
```

```
[8]: # 而且 zipped 迭代器只能使用一次，其内容在用完一次之后就会被清空  
list(zipped)
```

```
[8]: []
```

```
[9]: zipped = zip(one2five_list, fruits)  
zipped_list = list(zipped)  
for i in zipped_list:  
    print("解压之前",i,"， 解压之后",*i)  
# * 运算符可以 " 解压 " 元组并将其内容打印为单独的参数。
```

解压之前 (1, 'apple') , 解压之后 1 apple
解压之前 (2, 'banana') , 解压之后 2 banana
解压之前 (3, 'cherry') , 解压之后 3 cherry
解压之前 (4, 'grape') , 解压之后 4 grape
解压之前 (5, 'pear') , 解压之后 5 pear

```
[10]: # zipped_list 是一个已配对的序列，  
# zip 函数还可以拆分该序列，转化成各自的列表。  
index, fruits2 = zip(*zipped_list)  
print(index, fruits2)
```

```
(1, 2, 3, 4, 5) ('apple', 'banana', 'cherry', 'grape', 'pear')
```

```
[11]: # List Comprehension 列表解析，也称列表推导式  
# 它可以通过对现有可迭代对象（如列表、元组、字符串或范围）的每个元素应用表达式或操作  
# 来创建一个新列表。  
  
one2ten_list = list(range(11))  
# 筛选过滤偶数，获得奇数列表，  
# %2 代表除以 2 取余，如果有余数则留下  
get_odd = [i for i in one2ten_list if i % 2]  
get_odd
```

```
[11]: [1, 3, 5, 7, 9]
```

```
[12]: one2five_list = [1, 2, 3, 4, 5]
      # 对 one2five 列表每个元素算平方
      get_square = [i**2 for i in one2five_list]
      get_square
```

```
[12]: [1, 4, 9, 16, 25]
```

```
[13]: # python 除了 for 循环, 还可以用 while 循环
      i = 0
      while i < 5:
          print(fruits[i])
          i += 1
```

```
apple
banana
cherry
grape
pear
```

```
[14]: # while True if break 经常被搭配在一起使用,
      # 当出现特定情况时, 用 break 语句中断结束循环, 否则就一直循环下去
      # 此处用到 if, 只打印输出前五个元素。
      i = 0
      while True:
          if (i < 5):
              print(fruits[i])
              i+=1
          else:
              break
```

```
apple
banana
cherry
grape
pear
```

```
[15]: i = 0
      while i < 5:
```

```
if (fruits[i] == "grape"):
    i += 1
    continue # 如果是 grape, 那就不执行后面的代码, 直接进行下一轮循环。
print(fruits[i])
i += 1
```

apple
banana
cherry
pear

0.2 Summary

- for
- range
- list comprehensions 列表解析式
- enumerate
- zip
- while
- break
- continue