

知识点

1. 元组和列表的区别

2. 元组的常用操作

2.元组

元组（**tuple**）是值的一个序列。其中的值可以是任何类型，并且按照整数下标索引，与列表类似。但元组中的元素不能修改，列表中的元素可以修改。

2.元组

2.1元组与列表的区别

元组基本上都是不可改变的列表。元组几乎具有列表所有的特性，除开那些违反不变性的特征。也就是说，没有函数和方法可以改变元组。

不可变的tuple有什么意义？ 因为tuple不可变，所以代码更安全。如果可能，能用tuple代替list就尽量用tuple。当后续介绍字典类型时，会发现字典的键必须是不可变的，因此元组可以用作字典的键，但列表不能。

2.元组

2.1元组与列表的区别

语法上，元组就是用逗号分隔的一系列值，使用“=”将元组赋给变量。

```
>>> tuple1='a','1','boy'
```

虽然并不必需，但元组常常用括号括起来。

```
>>> tuple1=('a','1','boy')  
>>> tuple1  
( 'a', '1', 'boy')
```

2.元组

2.1元组与列表的区别

新建元组的另一种形式是使用内置函数**tuple**。不带参数时，它会新建一个空元组。

```
>>> t=tuple()  
>>> t  
()
```

2.元组

2.1元组与列表的区别

但是，要定义一个只有一个元素的tuple，例如：

```
>>> t = (1)
>>> t
1
```

则定义的不是tuple，是1这个数！这是因为括号()既可以表示tuple，又可以表示数学公式中的小括号，这就产生了歧义。因此，Python规定，这种情况下，按小括号进行计算，计算结果自然是1。所以，只有一个元素的tuple定义时必须加一个逗号“,”来消除歧义，即：

```
>>> t = (1,)
>>> t
(1,)
```

2.元组

2.1元组与列表的区别

其他序列（列表和字符串）的操作都可用于元组，除了那些会违反不变性的列表运算符。

- “+”和“*”运算符同样适用于元组。
- 成员操作（in）和for循环同样适用于元组
- 长度（len）、最大（max）和最小（min）同样适用于元组。

没有任何的操作能更改元组。例如append、extend、insert、remove、pop、reverse和sort不能用于元组。

2.元组

2.2元组的常用操作

由于元组和列表比较相似，使得对列表的很多操作如“+”、“*”等运算符以及长度（len）、最大（max）、最小（min）等运算都适用于元组。而那些如append、extend、insert、remove、pop、reverse和sort等能改变元素的操作都不能直接操作元组。

然而，当元组中存在列表元素项时，就可以使用上述如append等方法修改list中的元素，达到间接修改元组元素的目的。

2.元组

2.3元组的常用函数

元组的常用函数与列表类似，除了那些会违反不变性的函数。

表3-1 元组的常用函数

语 法	描 述
<code>cmp(tuple1, tuple2)</code>	比较两个元组元素
<code>len(tuple)</code>	计算元组元素个数
<code>max(tuple)</code>	返回元组中元素最大值
<code>min(tuple)</code>	返回元组中元素最小值
<code>tuple(seq)</code>	将列表转换为元组

代码练习题

已知元组info_tuple = ("zhangsan", 18, 1.75, "zhangsan")

请打印出如下信息：

- 1、“zhangsan”的索引；
- 2、统计“zhangsan”出现的次数；
- 3、统计元组元素的个数。

答案：

```
print(info_tuple.index("zhangsan"))  
print(info_tuple.count("zhangsan"))  
print(len(info_tuple))
```

代码练习题

打印元组`info_tuple = ("zhangsan", 18, 1.75)`中所有元素。

答案:

```
for my_info in info_tuple:  
    print(my_info)
```
