



Python 程序设计基础

Python Programming



元组

- **和列表一样，元组也是序列类型。**
- **元组其实是受限的列表。**
- **元组和列表最大的区别：元组是不可变对象，不可以直接修改元组中的元素值，也没有增加和删除元素操作。**
- **元组的优点：占用的内存空间较小；不会意外修改元组的值；比列表操作速度快；可以作为字典的键以及集合的成员，而列表不行。**
- **有些内置函数和方法返回元组。**



元组

- **元组中的元素用逗号分隔并且由一对圆括号括住。**

```
>>> t1 = ()
>>> t1
()
>>> t2 = (1, 2, 3)
>>> t2
(1, 2, 3)
>>> t3 = ("red", "green", "blue")
>>> t3
('red', 'green', 'blue')
>>> t4 = (2, "three", 4.5)
>>> t4
(2, 'three', 4.5)
>>> t5 = ("one", 2.0, 5, (100, 200))
>>> t5
('one', 2.0, 5, (100, 200))
```

一个不包含任何元素的元组被称为空元组；可以用空的圆括号创建一个空元组。

一个元组在另一个元组中，称为嵌套元组。



元组

- **创建只有一个元素的元组，要在元素后面加上逗号，否则创建的不是元组。**

```
>>> t1 = (1)      # 整数
>>> t1
1
>>> type(t1)
<class 'int'>
>>> t2 = (1,)     # 元组
>>> t2
(1,)
>>> type(t2)
<class 'tuple'>
```



元组

➤ 另一个创建元组的方法是使用 tuple 内置函数。

```
>>> t1 = tuple()
>>> t1
()
>>> t2 = tuple(range(1, 10))
>>> t2
(1, 2, 3, 4, 5, 6, 7, 8, 9)
>>> t3 = tuple("abcd")
>>> t3
('a', 'b', 'c', 'd')
>>> t4 = tuple([value for value in range(1, 10, 2)])
>>> t4
(1, 3, 5, 7, 9)
```

tuple 函数将字符串分割成由单独的字符组成的字符元组。



元组的基本操作

➤ 除了增加和删除操作，元组的基本操作和列表雷同

```
t1 = (55, -22, 3, 45, 11, 62, 38, 985, 211)
t2 = ("C++", "Java", "Python")
t3 = ("C++", "Java", "Python")
print(t1)
print(tuple(reversed(t1)))
print(tuple(sorted(t1, reverse=True)))
print(t1.count(11))
print(t1.index(11))
print(t1[0])
print(t1[-1])
print(t1[3:6])
print(t1 + t2)
print(2 * t2)
print("C#" in t2)
print(t2 is t3)
print(t2 == t3)
for value in t1:
    print(value, end=' ')
```

(55, -22, 3, 45, 11, 62, 38, 985, 211)
(211, 985, 38, 62, 11, 45, 3, -22, 55)
(985, 211, 62, 55, 45, 38, 11, 3, -22)
1
4
55
211
(45, 11, 62)
(55, -22, 3, 45, 11, 62, 38, 985, 211, 'C++', 'Java', 'Python')
('C++', 'Java', 'Python', 'C++', 'Java', 'Python')
False
False
True
55 -22 3 45 11 62 38 985 211



元组的基本操作

- `zip(*iterables)` 是内置函数，接受一个或多个序列作为参数，将序列中对应的元素打包成元组，然后返回由这些元组组成的可迭代对象。使用 `tuple` 函数将可迭代对象转换为一个元组并返回该元组，也可以使用 `list` 函数将可迭代对象转换为一个列表并返回该列表。
- 若作为参数传入的各序列长度不相同，返回的可迭代对象的长度与参数中长度最短的序列相同。

```
>>> x = [1, 2, 3]
>>> z = zip(x)
>>> tuple(z)
((1,), (2,), (3,))
```

```
>>> x = [1, 2, 3]
>>> y = "abcdef"
>>> z = zip(x, y)
>>> for i in z:
>>>     print(i)
```

```
(1, 'a')
(2, 'b')
(3, 'c')
```



元组的基本操作

- 尽管一个元组可以包含另一个元组，嵌套的元组本身还是被看作单个元素。下面这个元组的长度是 4。

```
>>> t = ("Hello", 1, ("Java", "Python", "C++"), (3, 4, 5))
>>> len(t)
4
```

- 如果元组中的元素是可变对象，这样的可变对象是可以修改的。从某种意义上讲，这也改变了元组。

```
>>> t = ("Python", 4.5, [1, 2, 3, 4])
>>> t
('Python', 4.5, [1, 2, 3, 4])
>>> t[2][0] = '111'
>>> t
('Python', 4.5, ['111', 2, 3, 4])
```