

Python程序设计

元组

刘安

苏州大学，计算机科学与技术学院

<http://web.suda.edu.cn/anliu/>

本节涉及到的知识点

- 元组的基本概念
- 创建元组的方法
- 元组和列表的相同点和不同点
- 元组的顶层不变性



元组的基本概念

- 和列表相似，元组也是一组任意类型的对象，对象在元组中具有固定的位置，位置用非负整数 $0, 1, \dots$ 表示
- 因为元素具有固定的位置，所以元组也支持索引、分片、连接、重复等操作（和列表类似）
- 和可变类型的列表不同，元组属于不可变类型，元组一旦创建之后，不可修改

元组的表现形式

- 列表的元素放在[]中，用逗号分开，[]不可少
- 元组的元素可以放在()中，用逗号分开
 - 一般情况下，()可以省略
 - 有些情况()不可少，比如空元组，函数调用等

```
>>> t = 1, 2, 3
>>> t
(1, 2, 3)
>>> t = 1,
>>> t
(1,)
>>> t = (1) # An integer
>>> t
1
```

```
>>> def f(*args):
        print(len(args))

>>> f(1, 2, 3) # 3 integers
3
>>> f((1, 2, 3)) # 1 tuple
1
```

创建元组

- 用()创建空元组
- 用逗号分开的对象序列
 - 如果只有一个对象，其后面必须要有逗号，否则最后一个对象后面可以没有逗号。圆括号是可选的
- 使用函数tuple([iterable])：使用iterable中的所有元素创建一个元组。如果没有指定iterable，创建空元组

```
>>> tuple(range(5))  
(0, 1, 2, 3, 4)  
>>> tuple('abcde')  
( 'a', 'b', 'c', 'd', 'e')  
>>> tuple([1, 2, 3, 4])  
(1, 2, 3, 4)
```

元组常见的操作

- 元组也是一种序列类型，支持序列对象上的大多数操作，比如索引、分片、连接、重复等，但元组属于不可变类型

```
>>> t = (1, 2) + (3, 4)
>>> t
(1, 2, 3, 4)
>>> t = t * 2
>>> t
(1, 2, 3, 4, 1, 2, 3, 4)
>>> t[3]
4
>>> t[1:5]
(2, 3, 4, 1)
>>> sorted(t)
[1, 1, 2, 2, 3, 3, 4, 4]
```

```
>>> t[1] = 10
Traceback (most recent call last):
  File "<pyshell#423>", line 1, in
<module>
    t[1] = 10
TypeError: 'tuple' object does not
support item assignment
```

元组的顶层不变性

- 元组的不变性是指顶层元素不能改变
- 如果顶层元素本身是可变的，比如列表，其内容是可变的

```
>>> t = (1, [2, 3], '4')
```

```
>>> t[0] = 1
```

```
Traceback (most recent call last):
```

```
File "<pyshell#430>", line 1, in <module>
```

```
t[0] = 1
```

```
TypeError: 'tuple' object does not support item assignment
```

```
>>> t[1][0] = '2'
```

```
>>> t
```

```
(1, ['2', 3], '4')
```