

## Python核心数据类型—元组

Python元组与列表类似，但是元组属于不可变类型

特点：

任意对象的有序集合：与列表相同 元组是一个位置有序的对象集合，内容维持着从左到右的顺序，支持异构(也就是元组中可以嵌套其他的核心数据类型)

创建元组

```
a = () #创建空元组
```

```
a = (1, 2, 3) #创建一个元组
```

```
a = [1, 2, 3]
```

```
b = tuple(a) ### tuple函数将列表转换成元组
```

元组可以通过切片来访问

```
In [12]: a
```

```
Out[12]: (1, 1, 1, 2, 2, 3, 4, 5)
```

```
In [13]: a[1:]
```

```
Out[13]: (1, 1, 2, 2, 3, 4, 5)
```

```
In [14]: a[4:]
```

```
Out[14]: (2, 3, 4, 5)
```

元组是不可变的类型 所以不可以进行修改

```
In [15]: a
```

```
Out[15]: (1, 1, 1, 2, 2, 3, 4, 5)
```

```
In [16]: a[1]
```

```
Out[16]: 1
```

```
In [18]: a[1] = 'a'
```

```
TypeError
```

```
Traceback (most recent call la
```

```
st)
```

```
<ipython-input-18-4ee7d669d8b4> in <module>()
```

```
----> 1 a[1] = 'a'
```

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

### 提示元组不可以进行修改

元组方法

1.count() #返回元素在元组中的出现次数

```
In [3]: a
```

```
Out[3]: (1, 1, 1, 2, 2, 3, 4, 5)
```

```
In [4]: a.count(1)
```

```
Out[4]: 3
```

```
In [5]: a.count(2)
```

```
Out[5]: 2
```

```
In [6]: a.count(3)
```

```
Out[6]: 1
```

2.index()返回元组中输入的参数的第一次出现的位置

```
In [7]: a
```

```
Out[7]: (1, 1, 1, 2, 2, 3, 4, 5)
```

```
In [9]: a.index(1)
```

```
Out[9]: 0
```

```
In [10]: a.index(2)
```

```
Out[10]: 3
```

```
In [11]: a.index(4)
```

```
Out[11]: 6
```

元组也支持类似字符串与列表一样的序列操作 如 + \* 运算

```
In [46]: x = (1,2,3)
```

```
In [47]: y = (2,3,4)
```

```
In [48]: x + y
```

```
Out[48]: (1, 2, 3, 2, 3, 4)
```

```
In [49]: x * 3
```

```
Out[49]: (1, 2, 3, 1, 2, 3, 1, 2, 3)
```

为什么有了列表还要使用元组？

答：因为元组具有不可变性， 可以确保元组在程序中不会被另外的引用所修改，而列表就没有这样的保证了