

目录:

- 创建元组
- 访问元组
- 遍历元组
- 附加程序

附加程序:

创建元组

元组是一种不可变的序列，一旦创建就不能修改。创建元组可以使用tuple([iterable])函数或直接用逗号“,”将元素分隔。

Python Shell 实例代码

```
1  >>>21,32,43,45                                ①
2  (21,32,43,45)
3  >>>(21,32,43,45)                                ②
4  (21,32,43,45)
5  >>>a = (21,32,43,45)
6  >>>print(a)
7  (21,32,43,45)
8  >>>('Hello', 'world')                          ③
9  ('Hello', 'world')
10 >>>('Hello', 'world', 1, 2, 3)                  ④
11 ('Hello', 'world', 1, 2, 3)
12 >>>tuple([21,32,43,45])                        ⑤
13 (21,32,43,45)
```

代码第①行创建了一个有4个元素的元组，创建元组时使用小括号把元素括起来不是必须的；

代码第②行使用小括号将元素括起来，这只是为了提高程序的可读性

代码第③行创建了一个字符型元组

代码第④行创建了一个字符串和整数混合的元组。Python中没有强制声明数据类型，因此元组中的元素可以是任何数据类型

另外，元组还可以使用 tuple([iterable]) 函数创建，参数iterable可以是任何可迭代对象。代码第⑤行使用了 tuple 函数创建元组对象，实参[21, 32, 43, 45]是一个列表，列表是可迭代对象，可以作为 tuple()函数参数创建元组对象。

创建元组还需注意极端情况：⚡

```
1  >>> a = (21)
2  >>> type(a)
3  <class 'int'>
4  >>> a = (21, )
5  >>> type(a)
6  <class 'tuple'>
7  >>> a = ()
8  >>> type(a)
9  <class 'tuple'>
```

访问元组

元组作为序列可以通过下标索引访问其中的元素，也可以对其进行分片

PythonShell实例：

```
1  >>>a = ('Hello', 'world', 1, 2, 3)      ①
2  >>>a[1]
3  'world'
4  >>>a[1:3]
5  ('world', 1)
6  >>>a[2:]
7  (1, 2, 3)
8  >>>a[:2]
9  ('Hello', 'world')
```

上述代码第①行是元组a，a[1]是访问元组的第二个元素，表达式a[1:3]、a[2:]和a[:2]都是分片操作

元组还可以进行拆包(Unpack)操作，就是将元组的元素取出给不同变量

PythonShell实例：

```
1  >>> a = ('Hello', 'world', 1, 2, 3)
2  >>> str1, str2, n1, n2, n3 = a
3  >>> str1
4  'Hello'
5  >>> str2
6  'world'
7  >>> n1
8  1
9  >>> n2
10 2
11 >>> n3
12 3
13 >>> str1, str2, *n = a
14 >>> str1
15 'Hello'
16 >>> str2
17 'world'
18 >>> n
19 [1, 2, 3]
```

**n可以直接获取剩余的元素*

遍历元组

一般用for语句遍历元组，实例代码

```

1  #coding=utf-8
2  #!/usr/bin/python3
3
4
5  a = {21, 32, 43, 45}
6
7  for item in a:
8      print(item)
9
10 print('-----')
11
12 for i, item in enumerate(a):
13     print('{0} - {1}'.format(i, item))

```

运行结果:

```

1  21
2  32
3  43
4  45
5  -----
6  0 - 21
7  1 - 32
8  2 - 43
9  3 - 45

```

其中enumerate(a)函数可以获取元组对象

附加程序

```

1  #coding = utf-8
2
3  a = (20)
4  print(type(a))
5
6  a = (20,)
7  print(type(a))
8  #tuple
9
10 a = (20, 30, 40, 50, 60)
11 print(a)
12 print(a[1])
13 print(a[1:3])
14
15 a = ('Hello', 'world', 1, 2, 3)
16 str1, str2, n1, n2, n3 = a
17 print(str1)
18 print(str2)
19 print(n1, "", n2, "", n3)
20
21 str1, str2, *n = a
22 print(n)

```

