

列表和元组

1

列表和元组 —— 列表

创建及访问列表

- 列表是有序、可变的数据类型
- 列表中可以包含不同类型的对象
- 列表可以由[]或工厂函数创建
- 支持下标及切片操作

更新列表

- 通过下标只能更新值，不能使用标添加新值

```
>>> alist = [10, 35, 20, 80]  
>>> alist[-1] = 100  
>>> alist[1:3] = [30, 50, 80]
```

列表内建函数

列表方法	操 作
<code>list.append(obj)</code>	向列表中添加一个对象obj
<code>list.count(obj)</code>	返回一个对象obj 在列表中出现的次数
<code>list.extend(seq)</code>	把序列seq的内容添加到列表中
<code>list.index(obj)</code>	返回obj对象的下标
<code>list.insert(index, obj)</code>	在索引量为index 的位置插入对象obj
<code>list.reverse()</code>	原地翻转列表
<code>list.sort()</code>	排序

2

列表和元组

—— 元组

创建元组

- 通过()或工厂函数tuple()创建元组
- 元组是有序的、不可变类型
- 与列表类似，作用于列表的操作，绝大多数也可以作用于元组

单元素元组

- 如果一个元组中只有一个元素，那么创建该元组的时候，需要加上一个逗号

```
>>> atuple = ('hello')
>>> print(atuple)
hello
>>> type(atuple)
<class 'str'>
>>> atuple = ('hello',)
>>> print(atuple)
('hello',)
>>> type(atuple)
<class 'tuple'>
```


“更新” 元组

- 虽然元组本身是不可变的，但是因为它同时属于容器类型，也就意味着元组的某一个元素是可变的容器类型，那么这个元素中的项目仍然可变

```
>>> atuple = ('bob', ['boy', 23])
>>> atuple[-1] = ['boy', 22]
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: 'tuple' object does not support item assignment
>>> atuple[-1][-1] = 22
>>> atuple[-1].append(175)
>>> atuple
('bob', ['boy', 22, 175])
```

- 用列表构建栈结构
 1. 栈是一个后进先出的结构
 2. 编写一个程序，用列表实现栈结构
 3. 需要支持压栈、出栈、查询功能



更多精彩...



<http://bj.linux.tedu.cn/>
企业QQ：86198501