

7.python列表类型

学习目标

1. 知道列表是用来干啥的
2. 如何快速创建列表
3. 列表的增删改查

列表的作用

用来存储多个数据类型元素。而且列表里面的元素是有序的，且可以重复，而且可以包含任意的数据类型。

列表的创建

1. 一维列表的创建

- 创建一个空列表的N种方法

```
# 使用list函数可以直接创建一个空列表
list1 = list()
# 使用[]创建一个空列表，推荐使用的方法，最简单方便
list2 = []
# 使用字符串的split方法创建一个空列表
list3 = "".split()
```

- 创建自定义的列表

```
# 每个列表元素之间是使用英文的逗号隔开的
list1 = [1,2,3,4]
list2 = ['a', 'b', 'c']
list3 = [1, '2', 'a', 4.5, '你好']
list4 = "abcd".split()
list5 = list('1234abcd')
```

- python的同一个列表里面可以包含任何的数据类型，也就是说，一个列表里面可以同时有数字，字符串，列表，元组，字典，集合，类.....

2. 多维列表的创建

- 多维列表其实就是列表里面嵌套列表

```
list1 = [[1,2,3,4], ['a', 'b', 'c', 'd']]
```

3. 使用列表推导式（解析式）创建列表(先知道有这么个东西，后面讲到for循环的时候会详细介绍)

```
list1 = [x for x in range(5)]
```

列表的访问

列表是序列类型，因此列表的成员是可以通过下标来访问的。

1. 一维列表的元素访问

```
list1 = [1, 2, 3, 4, 5]
list1[0]
```

2. 二维列表的元素访问

```
list1 = [[1, 2, 3, 4, 5], ['a', 'b', 'c']]
list1[0][1]
list1[1][2]
```

3. 使用for循环来遍历列表（先简单的了解for循环，知道怎么用for循环来遍历即可）

```
list1 = [1, 2, 3, 4, 5]
for i in list1:
    print i
```

4. 使用pop方法来获取列表中的元素

```
list1 = [1, 2, 3, 4, 5]
list1.pop()
```

列表的相关操作

列表是可变类型，因此列表中的元素是可以修改的，修改列表元素并不会改变列表所在的内存地址

1. 列表的修改

```
# 通过下标的方式修改列表的值
list1 = [1,2,3,4]
list1[0] = 5
# 通过切片的方式修改列表的值,注意: 使用切片的方式修改的时候, 右值必须是可迭代类型
(能用for循环遍历的)
# 目前已经学习的可迭代类型有: 字符串、列表
list1[1:3] = [666]
```

2. 往列表里面追加元素

```
list1 = []
# 使用append方法往列表里面添加元素
# append方法会把你给的数据当成是一个元素
list1.append(1)
# 使用insert方法往列表中指定的位置添加元素
list1.insert(1, 'a')
```

3. 扩展列表 (将另一个列表中的元素扩展进来)

```
list1 = [1,2,3,4]
list2 = ['a', 'b', 'c', 'd']
# 使用extend方法可以将另一个列表中的元素全部添加到列表中
list1.extend(list2)
```

4. 删除列表中的元素

```
# 使用pop方法删除列表中的元素, pop方法会返回被删除的元素
list1 = [1, 2, 3, 4, 5]
# 默认删除最后一个
list1.pop()
# 删除指定位置的元素, 这里删除的是下标为2的元素
list1.pop(2)
# 使用remove方法删除指定元素, remove方法不会返回被删除的元素
list1.remove(2) # 把2从list1中删除
```

5. 列表排序

```
# 使用列表内置的sort方法对列表进行排序，一般需要排序的列表都应该是由同一种元素组成的列表
# 因为不同类型的元素之间排序是没有什么意义的
list1 = [4, 1, 3, 6, 2, 5]
# sort方法默认是升序排列
list1.sort()
# 可以使用reverse方法或者在sort方法中指定reverse参数值为True将列表倒序
list1.reverse()
list1.sort(reverse=True)
# 当然，还可以指定sort方法中的cmp参数的返回值为-1对列表进行倒序，但是比较麻烦而已
list1.sort(cmp=lambda x, y: -1)

# 使用python的内置sorted方法对列表进行排序，参数的意义跟列表的sort一样的
list2 = sorted(list1)
```

列表操作符

1. 两个列表相加 (+)

```
list1 = [1, 2, 3, 4]
list2 = [5, 6, 7]
list3 = list1 + list2
```

2. 列表重复 (*)

```
# 使用*号可以将列表中的元素重复n次，返回一个重复之后新的列表
list1 = [1, 2, 3]
list2 = list1 * 3
```

3. 成员操作符 (in, not in)

```
>>> list1 = [1,2,3]
>>> 1 in list1
>>> 5 not in list1
```

4. 切片操作

- [:] 得到整个列表中的所有元素
- [:end] 得到从开始到end-1的所有元素
- [start:] 得到从start开始到结束的所有元素
- [start: end] 得到从start到end-1的所有元素
- [start: end: step] 从start到end中每隔step取一个元素

列表的内建函数

函数	说明
<code>list.append(obj)</code>	往列表里面追加一个元素obj
<code>list.count(obj)</code>	统计obj在列表中出现的次数
<code>list.extrend(iterable)</code>	将另一个可迭代对象（列表、字符串、元组…）中的所有元素添加到列表里面
<code>list.index(obj)</code>	返回元素obj在列表中的位置（下标）
<code>list.insert(index, obj)</code>	将obj添加到列表index的位置中
<code>list.pop([index])</code>	从列表中删除一个元素，并返回删除的元素，默认删除最后一个，可通过索引指定删除哪个元素
<code>list.remove(value)</code>	从列表中删除第一个匹配到的元素value，不返回删除的元素
<code>list.reverse()</code>	翻转列表，将元素前后翻转
<code>list.sort(cmp=None, key=None, reverse=False)</code>	将列表进行排序，默认是从小到大进行排序的，可以通过设置reverse值将结果倒序。key是指以什么进行比较，cmp是比较两个值的大小的函数

列表的其他操作（对于零基础的现阶段不要求掌握）

函数	说明
<code>sum(iterable)</code>	将迭代对象中的每个数进行相加，迭代对象的元素必须是数字组成的，不能是字符串
<code>zip(seq1 [, seq2 [...]])</code>	返回一个由元组组成的列表，元组中元素的数量取决于队列的个数
<code>enumerate(iterable[, start])</code>	返回一个枚举对象（可迭代），对象中的每个值由元素的索引以及元素值组成的元组
<code>map(function, seq[,...])</code>	将seq中的每个元素都作为function的参数，最后返回一个列表。如果队列不止一个，那么每个队列就代表一个参数（比如有两个队列，那么函数必须要有两个参数接收才行），如果每个队列的元素个数不一样，那么缺少元素就会以None赋值。

作业

1. 掌握python列表的所有相关操作，及对应的知识点。因为列表在python中用得是非常多的。
2. 用sum函数结合前面学到的列表相关知识，一句话计算出从1加到100的和。
3. 将题目2改成计算100内所有奇数的和又该怎么做？
4. 熟练掌握列表的切片操作，切片操作对于所有的序列类型都是通用的，掌握了一种就能掌握其他的。