Python的列表(List)操作

|  |  |
| --- | --- |
| 列表的申明 | example\_list = [“a”,1,{“a”:”b”}] |
| 列表增加元素 | Example\_list.insert(index,obj)指定位置添加元素  Example\_list.append()在列表末尾添加元素 |
| 列表改变元素值 | Example\_list[index]=obj |
| 列表删除元素 | Example\_list.remove(obj)删除第一个符合的元素  Del example\_list[index], example.pop(index)  删除index下标的元素 |
| 列表切片 | Example\_list[starting\_index,end\_index] 取出包含starting\_index直到但不包含end\_index的子列表 |
| 列表判断元素是否存在 | Obj in example\_list |
| 列表判断有多少个元素 | Example\_list.count(obj) |
| 列表查找第一个符合指定元素下标 | Example\_list.index(obj)  如果没有该元素则报错。 |
| 列表扩充列表 | Example\_list.extend(another\_list) |
| 列表排序 | Example\_list.sort() |
| 列表反转 | Example\_list.reverse() |
| 列表的复制 | Example\_list.copy()  Copy函数复制一份完整的列表，从而复制以后列表于源列表指向不同地址，不相互影响。  1. 直接列表赋值，example\_list2 = example\_list 只会赋值地址，两个列表指向同一内存空间。  2. 列表的Copy函数默认只复制表层简单数据，复制前后的变量指向不同的内存地址。但是如果列表中含有列表则只会对内嵌列表复制地址。如果要完完全全对嵌套的列表也复制一份，则要引入原生库Copy，使用方法copy.deepcopy(obj)才会完完全全的复制。默认浅复制的原因是防止一下完全深复制，因为嵌套所占用的内存太多。 |

P.s

1.Tuple(元组) 为只读列表，只有列表中index,count方法。

2. 去除数组中所有的空格(count()与remove()方法的联动)：

Count()可以用于返回有多少个空格，和remove()每一次去除一个空格

for i in list(range(list.count(“ “))):

list.remove(“ “)