# 3.1 列表是什么

用方括号[ ]来表示列表，并用逗号来分隔其中的元素。如果让Python将列表打印出来，Python将打印列表的内部表示，包括方括号

## 3.1.1访问列表元素

要访问列表元素，可指出元素的索引，并将其放在括号中。当你请求获取列表元素时，Python只返回该元素，而不包括方括号和引号。

## 3.1.2 索引从0而不是1开始

## 3.1.3 使用列表中的各个值

# 3.2 修改、添加和删除元素

## 3.2.1 修改列表元素

## 3.2.2 在列表中添加元素

方法append()将元素添加到列表末尾，而不影响列表中的其他所有程序

方法insert()可在列表的任何位置添加新元素。只需指定新元素的索引和值

## 3.2.3 从列表中删除元素

1. 使用del语句删除元素

使用del语句可以删除任何位置的列表元素，条件是知道其索引

1. 使用方法pop()删除元素

方法pop()可删除列表末尾的元素，并让你能够接着使用它

列表就像一个栈，而删除列表末尾的元素相当于弹出栈顶元素

该方法带返回值

1. 弹出列表中任何位置处的元素

可以使用pop()来删除列表中任何位置的元素，只需在括号中指定要删除的元素的索引即可。

总结：如果你要从列表中删除一个元素，且不再以任何方式使用它，就是用del语句；如果你要在删除元素后还能继续使用它，就是用方法pop()。

1. 根据值删除元素

如果你只要要删除元素的值，可使用方法remove()。

使用remove()从列表中删除元素时，也可接着使用它的值。

注意：方法remove()只删除第一个制定的值。如果要删除的值在列表中出现多次，就需要使用循环来判断是否删除了所有这样的值。

# 3.3 组织列表

## 3.3.1 使用方法sort()对列表进行永久性排列

可以按与字母相反的顺序排列列表元素，为此，只需向sort()方法传递参数reverse= True。

## 3.3.2 使用函数sorted()对列表进行临时排序

函数sorted()让你能够按特定顺序显示列表元素，同时不影响它们在列表中的原始排列顺序。如果要按与字母顺序相反的顺序显示列表，也可向函数sorted()传递参数reverse=True，作为第二个参数。

## 3.3.3 倒着打印列表

要反转列表元素的排列顺序，可使用方法reverse()。

方法reverse()永久性地修改列表元素的排列顺序，但可随时恢复到原来的排列顺序，为此只需对列表再次调用reverse()即可。

## 3.3.4 确定列表的长度

使用函数len()可快速获悉列表的长度。

# 3.4 使用列表时避免索引错误

索引-1总是返回最后一个列表元素，仅当列表为空时，这种访问最后一个元素的方式才会导致出错。