

列表的数据操作

我们对于可变数据（例如，列表，数据库等）的操作，一般包含增、删、改、查四个方面。

一、添加元素

添加元素有以下几个方法：

- append 在末尾添加元素
- insert 在指定位置插入元素
- extend 合并两个列表

append

append会把新元素添加到列表末尾

```
#定义变量A，默认有3个元素
A = ['xiaoWang', 'xiaoZhang', 'xiaoHua']

print("-----添加之前，列表A的数据-----A=%s" % A)

#提示、并添加元素
temp = input('请输入要添加的学生姓名:')
A.append(temp)

print("-----添加之后，列表A的数据-----A=%s" % A)
```

insert

insert(index, object) 在指定位置index前插入元素object

```
strs = ['a', 'b', 'm', 's']
strs.insert(3, 'h')
print(strs) # ['a', 'b', 'm', 'h', 's']
```

extend

通过extend可以将另一个集合中的元素逐一添加到列表中

```
a = ['a','b','c']
b = ['d','e','f']
a.extend(b)
print(a) # ['a', 'b', 'c', 'd', 'e', 'f'] 将 b 添加到 a 里
print(b) # ['d','e','f'] b的内容不变
```

二、修改元素

我们是通过指定下标来访问列表元素，因此修改元素的时候，为指定的列表下标赋值即可。

```
#定义变量A，默认有3个元素
A = ['xiaoWang','xiaoZhang','xiaoHua']

print("-----修改之前，列表A的数据-----A=%s" % A)

#修改元素
A[1] = 'xiaoLu'

print("-----修改之后，列表A的数据-----A=%s" % A)
```

三、查找元素

所谓的查找，就是看看指定的元素是否存在，以及查看元素所在的位置，主要包含一下几个方法：

- in 和 not in
- index 和 count

in, not in

python中查找的常用方法为：

- in（存在），如果存在那么结果为true，否则为false
- not in（不存在），如果不存在那么结果为true，否则false

```
#待查找的列表
nameList = ['xiaoWang','xiaoZhang','xiaoHua']

#获取用户要查找的名字
findName = input('请输入要查找的姓名:')

#查找是否存在
if findName in nameList:
    print('在列表中找到了相同的名字')
else:
    print('没有找到')
```

结果1: (找到)

```
localhost:Desktop jiangwei$ python test.py
请输入要查找的姓名:x
```

结果2: (没有找到)

```
localhost:Desktop jiangwei$ python test.py
请输入要查找的姓名:x
```

说明:

in的方法只要会用了, 那么not in也是同样的用法, 只不过not in判断的是不存在

index, count

index用来查找元素所在的位置, 如果未找到则会报错; count用来计算某个元素出现的次数。它们的使用和字符串里的使用效果一致。

```
>>> a = ['a', 'b', 'c', 'a', 'b']
>>> a.index('a', 1, 3) # 注意是左闭右开区间
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ValueError: 'a' is not in list
>>> a.index('a', 1, 4)
3
>>> a.count('b')
2
>>> a.count('d')
0
```

四、删除元素

类比现实生活中，如果某位同学调班了，那么就应该把这个条走后的学生的姓名删除掉；在开发中经常会用到删除这种功能。

列表元素的常用删除方法有：

- del：根据下标进行删除
- pop：删除最后一个元素
- remove：根据元素的值进行删除

del

```
movieName = ['加勒比海盗', '骇客帝国', '第一滴血', '指环王', '霍比特人', '速度与激情']
print('-----删除之前-----movieName=%s' % movieName)
del movieName[2]
print('-----删除之后-----movieName=%s' % movieName)
```

pop

```
movieName = ['加勒比海盗', '骇客帝国', '第一滴血', '指环王', '霍比特人', '速度与激情']
print('-----删除之前-----movieName=%s' % movieName)
movieName.pop()
print('-----删除之后-----movieName=%s' % movieName)
```

remove

```
movieName = ['加勒比海盗', '骇客帝国', '第一滴血', '指环王', '霍比特人', '速度与激情']
print('-----删除之前-----movieName=%s' % movieName)
movieName.remove('指环王')
print('-----删除之后-----movieName=%s' % movieName)
```

五、排序(sort, reverse)

sort方法是将list按特定顺序重新排列，默认为由小到大，参数reverse=True可改为倒序，由大到小。

reverse方法是将list逆置。

```
>>> a = [1, 4, 2, 3]
>>> a
[1, 4, 2, 3]
>>> a.reverse() # 逆置，不排序
>>> a
[3, 2, 4, 1]
>>> a.sort() # 默认从小到大排序
>>> a
[1, 2, 3, 4]
>>> a.sort(reverse=True) # 从大到小排序
>>> a
[4, 3, 2, 1]
```

练习

请删除列表 `words = ['hello', '', ' ', 'good', 'hi', '', 'yes', ' ', 'no']` 里所有的空字符串。