



4.1.2 列表

• 列表操作

- `list.append(x)`

在列表的末尾添加元素x，等价于 $a[\text{len}(a):] = [x]$

- `list.extend(L)`

在列表末尾加入指定列表L中的所有元素，等价于 $a[\text{len}(a):] = L$ 或 $a + L$

- `list.insert(i, x)`

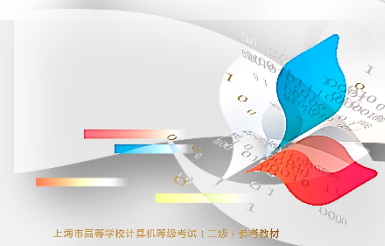
在给定位置插入元素，即在位置i处插入x，其余元素依次向后退。

`a.insert(0, x)`

`a.insert(len(a), x)`

- `list.remove(x)`

删除列表中第一个值为x的元素，若该元素不存在则出错。



4.1.2 列表

- `list.pop([i])`

删除列表中给定位置*i*处的元素，并返回该元素。若不指定索引值 (`list.pop()`)，则移除并返回列表中的最后一个元素。

- `list.clear()`

删除列表中的所有元素，等价于：`del a[:]`。

- `list.index(x)`

返回列表中值为*x*的元素的位置索引，若不存在则出错。

- `list.count(x)`

返回*x*在列表中出现的次数。

- `list.sort(key=None, reverse=False)`

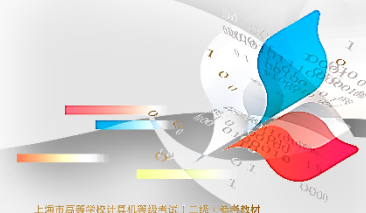
对列表中的元素排序，默认为升序。

- `list.reverse()`

将列表中元素的顺序反转。

- `list.copy()`

返回列表的浅复制，等价于`a[:]`。



4.1.2 列表

• 【例4-7】 列表操作举例。

```
>>> a = [66.25, 333, 333, 1, 1234.5]
>>> print(a.count(333), a.count(66.25), a.count('x'))
2 1 0
>>> a.insert(2, -1)
>>> a.append(333)
>>> a
[66.25, 333, -1, 333, 1, 1234.5, 333]
>>> a.index(333)
1
>>> a.remove(333)
>>> a
[66.25, -1, 333, 1, 1234.5, 333]
>>> a.reverse()
>>> a
[333, 1234.5, 1, 333, -1, 66.25]
>>> a.sort()
>>> a
[-1, 1, 66.25, 333, 333, 1234.5]
>>> a.pop()
1234.5
>>> a
[-1, 1, 66.25, 333, 333]
```

4.1.2 列表

- 【例4-7】 列表操作举例。

```
>>> a = [-1, 1, 66.25, 333, 333, 1234.5]
```

```
>>> del a[0]
```

```
>>> a
```

```
[1, 66.25, 333, 333, 1234.5]
```

```
>>> del a[2:4]
```

```
>>> a
```

```
[1, 66.25, 1234.5]
```

```
>>> del a[:]
```

```
>>> a
```

```
[]
```

del操作可以通过元素的位置索引来将其从列表中删除。*注意：del不是一个列表对象的操作方法，而是可用于列表元素的Python内置操作命令。*

del与pop()不同，pop()会返回被删除的元素。此外，del还可以用来删除子列表或整个列表变量。



4.1.2 列表



【例】求一句英文句子的单词数。单词是字母数字串，中间没有空格。

```
sentence="This is a pen "  
words=sentence.split()  
print(len(words))
```



4.1.2 列表

- 列表解析 (列表推导式)

常用于从集合对象中有选择地获取并计算元素。

[<表达式> for x1 in <序列1> [..... for xN in <序列N> if <条件表达式>]]

【例4-8】 创建平方数列表的方法比较。

```
squares = []
for x in range(10):
    squares.append(x**2)
print (squares)
```

列表解析语句

```
squares = [x**2 for x in range(10)]
```



4.1.2 列表

【例】在一行中输入若干个整数，至少输入一个整数，整数之间用空格分割，要求将数据按从小到大排序输出。

```
a=input().split()
list_a=[int(n) for n in a]
list_a.sort()
print (list_a)
```

```
from math import pi
[str(round(pi,i)) for i in range(1,6)]
[str(round(pi,i)) for i in range(1,6) if i%2 !=0]
```

4.1.2 列表

- 模块的引用

- import方式:

- import <模块1>

- import <模块2>

- ...

- ...

- import <模块n>

- 引入模块中的所有函数, 调用的时候需要加模块名

- from方式:

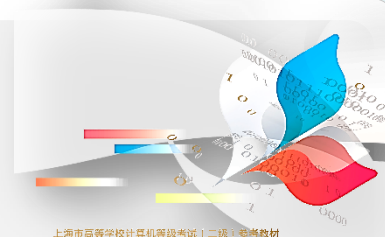
- from <模块名> import <函数名>

- 引入模块中的单个函数, 调用的时候也不需要再加模块名

- from <模块名> import *

- 引入模块中的所有函数, 调用的时候不需要再加模块名





上海市高等学校计算机等级考试(二级)参考教材

Python程序设计基础

(第2版)

主編 李东方 支晓勇

列表各种创建方法比较

- from time import time
- from random import random
- #用 "+"产生列表
- start=time()
- lst=[]
- for i in range(100000):
- lst=lst+[random()]
- print("addtest",str(time()-start)+"s")
- #用append产生列表
- start=time()
- lst=[]
- for i in range(100000):
- lst.append(random())
- print("appendtest",str(time()-start)+"s")

- #用insert产生列表
- start=time()
- lst=[]
- for i in range(100000):
- lst.insert(0,random())
- print("inserttest",str(time()-start)+"s")
- #用列表解析产生列表
- start=time()
- lst=[random() for i in range(100000)]
- print("listexptest",str(time()-start)+"s")