List

特性：

1.列表可以包含不同类型数据，但通常为相同类型数据

2.列表元素为mutable

方法：

append(value)：在列表尾部添加value

extend(iterable)：在列表尾部添加iterable内的所有item

insert(index,value)：在index索引之前增加value

remove(value)：在列表中删除第一个值为value的内容

pop([index])：删除index索引处的值并返回，如未指定，则删除最后一个item并返回

clear()：删除列表中的所有item

index(value[,start[,end]])：从列表中找出第一个值为value的索引，如未找到，引发ValueError异常。可指定搜索的起始start和终止end位置

count(value)：返回列表中value出现的次数

sort(key=None,reverse=False)：对列表进行从小到大排序(内部)，可指定key和反向排序(reverse)

reverse()：将列表反向排序(内部)

copy()：对列表进行浅复制

\*\*赋值、浅复制、深复制差别

\*\*赋值：两个变量指向同一个数组

\*\*浅复制：复制列表内的值，但不复制子列表内容(复制子列表索引)

\*\*深复制：复制列表内的值，并且复制子列表内容

示例：

输入

import copy  
a = [1,2,3,4,5]  
b = a  
b[1] = 100  
print(a)  
print(b)  
  
c = [1,2,3,4,5]  
d = c.copy()  
d[1] = 100  
print(c)  
print(d)  
  
e = [1,[2,3,4],5,6]  
f = e.copy()  
f[1][0] = 100  
print(e)  
print(f)  
  
g = [1,[2,3,4],5,6]  
h = copy.deepcopy(g)  
h[1][0] = 100  
print(g)  
print(h)

输出

[1, 100, 3, 4, 5]  
[1, 100, 3, 4, 5]  
[1, 2, 3, 4, 5]  
[1, 100, 3, 4, 5]  
[1, [100, 3, 4], 5, 6]  
[1, [100, 3, 4], 5, 6]  
[1, [2, 3, 4], 5, 6]  
[1, [100, 3, 4], 5, 6]

要点：

1.append()和pop()组成栈(后进先出)

2.append()/insert()和pop(i)组成队列(先进先出)

\*\*方法2并不高效，替代高效方案为collections.deque类替换，方法append()和popleft()

列表推导式

squares = [x\*\*2 for x in range(10) if x > 2]

等效于

squares = []

for x in range(10):

if x > 2:

squares.append(x\*\*2)

嵌套列表推导式

transposed = [[row[i] for row in matrix] for i in range(4)]

等效于

transposed = []

for i in range(4):

transposed.append([row[i] for row in matrix])

del删除

del arr[INDEX/SLICE]

用于删除数组中的索引/切片内容

tuple

tuple为immutable，以()限定(最外层tuple可不使用括弧)

空元素tuple：tuple1 = ()

单元素tuple(和字符串区分)：tuole2 = (“str”,)

\*\*(i for i in range(10))为生成器表达式，而非元组推导式

set

set为mutable，是item具有唯一性的无序组合，以{}限定，元素item为immutable

创建空元素：set()

可使用集合推导式，例：{x for x in “apple”}

方法：

add(item)：添加item内容到set

update(\*others)：set依次与可变参数的other求并集，other为iterable

discard(elem)：从set中移除值为elem的项

remove(elem)：从set中移除值为elem的项，如无，则跑出KeyError异常

pop()：从set中随机移除一项，如set为empty，则抛出KeyError异常

clear()：清除set中的所有项

copy()：set内容浅复制

set还支持子项(<=)、超项(>=)、并集(|)、交集(&)、差集(-)、对称差集(^)

在set中，<和>代表包含比较，与数学的集合相同；而与列表的逐项按编码字典比较

dictionary

dict为mutable，是key/value对的无序组合，并且key值在dict中具有唯一性，以{key:value}限定

特征：

1.key必须为immutable对象，如：数字、字符串、tuple(不能内嵌list)

2.{}可直接创建空dict

3.增加/修改item：dict[key] = value

4.删除item：del dict[key]

方法：

keys()：返回keys

values()：返回values

get(key[,default])：当key在dict中，返回value，不然抛出KeyError异常(提供default参数，则返回该参数)

items()：返回数组，item为key/value组成的tuple

pop(key[,default))：当key在dict中，删除item并返回value；不然则抛出KeyError异常(提供default参数，则返回该参数)。

popitem()：删除一个item并返回，删除顺序遵循FILO

clear()：删除dict所有项

dict推导式：{x:x\*\*2 for x in range(4)}

dict相比于list的优缺点：

1.dict不随item的增加而增加搜索时间

2.dict占用内存较高

所以，dict即使用空间换取搜索时间

\*\*enumerate(iterable,start=0)：返回enumerate对象，对象的item为计数和iterable中的元素组成的tuple

\*\*zip(\*iterables)：返回zip对象(iterator)，对象的item为每个iterable的同位置元素组成的tuple，tuple长度由len最小的iterable决定

\*\*reversed(seq)：返回一个iterator对象，seq本身不改变

\*\*sorted(iterable, \*, key=None, reverse=False)：返回一个新list，iterable本身不变。list.sort()改变自身，并且返回None

运算符优先级：

算数运算符

比较运算符

逻辑运算符(not/and/or)

not/and/or特征

1.优先级：

not

and

or

示例1：

代码

a = True or False and False  
b = (True or False) and False  
print(a)  
print(b)

结果

True  
False

2.and和or称为短路操作符

当从前面操作数可以推导出结果时，不对后续操作数进行推导

示例2：

代码

a = True or 10/0  
print(a)

结果

True

3.当使用非True/False(即正规值)时，表达式返回最后一个调用的操作数

示例3：

代码

a = 3 and 2 or 1  
print(a)

结果

2

4.运算方向：从左至右

相同sequence比较规则：

1.每个item按字典顺序进行比较

2.当两个sequence长度不同时，在比较的item内无法分出大小，则较长的sequence大

3.当sequence内含嵌套sequence时，进行递归比较