3.3 列表和元组使用

- 列表（list）
  - 由一系列按照指定顺序排列的元素组成。列表中的元素可以是不同类型。
  - 列表的表示用方括号（[ ]）将元素括起来，元素之间用逗号（,）分隔
  - 列表是序列类型的一种，序列所有的特性和操作对于列表都是成立的，除此之外，列表还有自己的特殊操作。

列表

- 列表的创建

1. 直接使用列表的字面量。

```
a = []          # 创建一个空列表
a = [2,3,5,7,11,13]
```

2. 使用list()将其他数据类型转换成一个列表。

```
a = list('hello')
```

a的内容是: ['h', 'e', 'l', 'l', 'o']

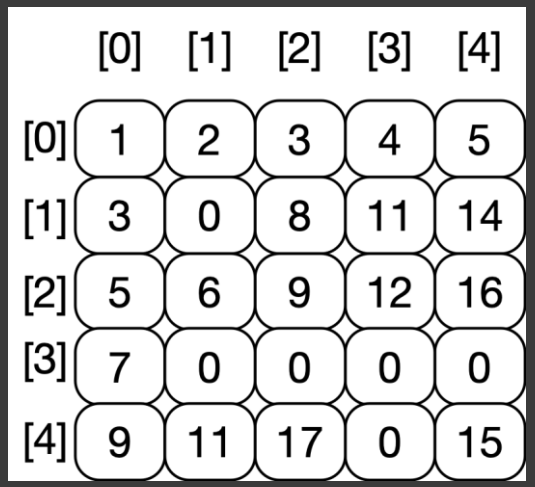
```
list(range(1,10,2))
```

结果是：[1,3,5,7,9]

列表（续）

列表的元素类型可以是任何类型，也包括列表类型。当列表的元素是列表时，可以构成多维列表，如同一个矩阵。

```
matrix = [
    [1, 2, 3, 4, 5],
    [3, 0, 8,11,14],
    [5, 6, 9,12,16],
    [7, 0, 0, 0, 0],
    [9,11,17, 0,15]
]
```



	[0]	[1]	[2]	[3]	[4]
[0]	1	2	3	4	5
[1]	3	0	8	11	14
[2]	5	6	9	12	16
[3]	7	0	0	0	0
[4]	9	11	17	0	15

- 用matrix[0][0]访问其中第一行第一列的元素
- 矩阵的每一行都是一个列表。

基本的列表操作

- 列表元素的赋值

  和字符串不同，列表中的元素可以被修改。

  ```
  a = [1,3,5,7,11]
  a[0] = 2
  print(a)
  ```

  输出:

  ```
  [2,3,5,7,11]
  ```

基本的列表操作（续）

- 删除元素

  用del语句删除列表中的元素。

```
name = ['Alice', 'Kim', 'Karl', 'John']
del name[2]
print(name)
```

结果是：

```
['Alice', 'Kim', 'John']
```

基本的列表操作（续2）

- 切片赋值

  切片表示列表的一部分，可以被赋值，接受另外一个列表，替换切片那部分元素。

```
name = list('Perl')
name[2:] = list('ar')
print(name)
```

结果是：

```
['P', 'e', 'a', 'r']
```

列表的函数或方法

表 3-4 列表的常用方法或函数

列表的常用方法或函数	描述
L.append(x)	在列表L尾部追加x
L.clear()	移除列表L的所有元素
L.count(x)	计算列表L中x出现的次数
L.copy()	列表L的备份
L.extend(x)	将列表x扩充到列表L中
L.index(value[,start[,stop]])	计算在指定范围内value的下标
L.insert(index,x)	在下标index的位置插入x
L.pop(index)	返回并删除下标为index的元素，默认是最后一个
L.remove(value)	删除值为value的第一个元素
L.reverse()	倒置列表L
L.sort()	对列表元素排序

列表的函数或方法（续）

- 追加函数append()

用函数append()在列表后面增加一个元素。

```
a = [2,3,5,7,11]
a.append(13)
print(a)
```

结果是：

```
[2, 3, 5, 7, 11, 13]
```

列表的函数或方法（续2）

- 扩展函数extend()

用函数extend()把另一个列表的内容添加到列表的后面。

```
a = [2,3,5,6,11]
a.extend([13,17])
print(a)
```

结果就是：

```
[2, 3, 5, 6, 11, 13, 17]
```

列表的函数或方法（续3）

- 插入函数insert()

函数insert()用来将一个数据插入到列表的指定位置。

```
a = [2,3,5,6,11]
a.insert(2,4)
a.insert(12,13)  #指定插入的位置(12)不存在，加到最后
print(a)
```

结果是：

```
[2, 3, 4, 5, 6, 11, 13]
```

列表的函数或方法（续4）

- 删除函数remove()

用函数remove()删除某个数据在列表中第一次出现的元素。

```
a = [2,3,5,7,11]
a.remove(5)
print(a)
```

输出：

```
[2, 3, 7, 11]
```

注:如果要删除的数据不在列表中，则会发生错误。

列表的函数或方法（续5）

- 弹出函数pop()

用函数pop()删除并返回列表中指定下标（位置）的数据，如果下标不指定，则删除最后一项。

```
a = [2,3,5,7,11]
print(a.pop())
print(a.pop(2))
print(a)
```

输出：

```
11
5
[2, 3, 7]
```

列表的函数或方法（续6）

- 反转函数reverse()

用函数reverse()将列表反转。

```
a = [2,3,5,7,11]
a.reverse()
print(a)
```

结果是：

```
[11, 7, 5, 3, 2]
```

列表的函数或方法（续7）

- 查找函数index()

函数index()用于在列表中查找某个数据第一出现的位置（下标）。

```
a = [2,3,5,7,11]
print(a.index(3))
```

输出：

```
1
```

注:如果查找的数据在列表中不存在，则会发生错误。

字符串和列表的互相操作

- 拆分字符串函数split()

函数split()用一个字符或子串将一个字符串分隔成列表的元素。

```
date = '3/11/2018'
a = date.split('/')
print(a)
```

输出：

```
['3', '11', '2018']
```

如果split()函数不带参数，就是以空格来分割字符串。

```
name = 'John Johnson'
a = name.split()
print(a)
```

输出：

```
['John', 'Johnson']
```

字符串和列表的互相操作（续）

- 聚合字符串函数join()

函数join()用于将一个列表的各个字符串类型的元素组合成一个字符串，元素之间用指定的内容填充。

```
a = ['hello','good','boy','wii']
print(' '.join(a))
print(':'.join(a))
```

输出：

```
hello good boy wii
hello:good:boy:wii
```

字符串和列表的互相操作（续2）

例3-2 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))
```

程序运行：

4

字符串和列表的互相操作（续3）

- 例3-3 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。

程序代码：

```
nums=input()
numl=[int(n) for n in nums.split()]
numl.sort()
print(numl)
```

程序输入：

5 -76 8 345 67

程序输出：

[-76, 5, 8, 67, 345]

创建列表

- 用append方法：
- lst=[]
- for i in range(4):
- 　　lst.append(input())

- 用列表解析方法：
- lst=[input() for i in range(4)]

列表各种创建方法比较

```
#创建100000个元素的列表，比较各种方法的效率
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")
```

元组

- 元组（tuple）

　　元组是不可修改的任何类型的数据序列。元组像列表一样可以表达任何类型、任意数量的数据的有序序列。

元组的字面量用圆括号()而不是方括号[]。

(1, 3.2, 5, 7.0, 9)

('not', 'and', 'or')

元组（续）

- 创建元组
  - 用元组的字面量

```
d = (100,20)
print(d)
```

输出：

```
(100, 20)
```

  - 用tuple()方法，把其他序列类型转换成元组。

```
a = tuple([2,3,5,7,11])
print(a)
```

输出：

```
(2, 3, 5, 7, 11)
```

元组（续2）

- 元组不可修改
  - 元组是不可修改的，即不能对元组中的元素进行增加，删除，修改或排序。
  - 列表中的修改函数append()、insert()、remove()以及del语句都不能用于元组。
  - 元组常用方法和函数

元组常用方法和函数	描述
T.count(x)	计算x元素出现的次数
T.index(x)	计算X元素的下标

列表加元组表示二维表

```
students=[(3180102988,"褚好"),
                    (3170102465,"王凯亮"),
                    (3160104456,"李永"),
                    (3171104169,"陈鑫"),
                    (318400429,"徐杭诚")]

for row in students: #按行存取
    print(row[0],row[1])
print()

for id,name in students: #按行拆包存取
    print(id,name)
print()

for index in range(len(students)):  #按索引存取
    print(students[index][0],students[index][1])
```

学号	姓名
3180102988	褚好
3170102465	王凯亮
3160104456	李永
3171104169	陈鑫
318400429	徐杭诚

3.4 随机函数库（random库）

- 计算机的随机函数生成的随机数，是按照一定的算法模拟产生的，其结果是确定的，是伪随机数。
- Python中的random模块用于生成伪随机数。

随机函数库（续）

表 3-5 random库的常用函数

函数名	含义	示列
random.random()	返回一个介于左闭右开[0.0, 1.0)区间的浮点数	random.random()
random.uniform(a, b)	返回一个介于【a，b】的浮点数。	random.uniform(1,10)
random.randint(a,b)	返回【a,b】的一个随机整整。	random.randint(15,30)
random.randrange([start], stop[, step])	从指定范围内，获取一个随机数	random.randrange(10, 30, 2)
random.choice(sequence)	从序列中获取一个随机元素	random.choice([3,78,43,7])
random.shuffle(x)	用于将一个列表中的元素打乱,即将列表内的元素随机排列	random.shuffle(l) ，l是序列
random.sample(sequence, k)	从指定序列中随机获取长度为k的序列并随机排列	random.sample([1,4,5,89,7],3)
random.seed(n)	对随机数生成器进行初始化的函数，n代表随机种子。参数为空时，随机种子为系统时间	random.seed(2)

随机函数库（续2）

要使用random库，先要用“import random”语句引入random库。

```
>>> import random
>>> random.random()
0.11529299890219902

>>> random.uniform(1,10)
4.6467045646433975

 >>> random.randint(20,30)
20

>>> random.randrange(10, 30, 2)
24
```

随机函数库（续3）

```
>>> random.choice([3,78,43,7])
3

>>> l=['A',1,78,'b']
>>> random.shuffle(l)
>>> l
[1, 'b', 78, 'A']

>>> random.sample([1,4,5,89,7],3)
  [7, 5, 4]

>>> random.sample("This is a sample",5)
['s', 'h', ' ', 'a', 'a']
```

随机函数库（续4）

```
>>> random.seed(2)
>>> random.random()
0.9560342718892494

>>> random.randint(1,10)
1

 >>> random.seed(2)    #重复上面产生的随机数
>>> random.random()
0.9560342718892494

>>> random.randint(1,10)
1
```

随机函数库（续5）

例3-4 掷硬币，正面向上的概率是多少？

程序代码：

```
#掷10000次硬币，正面向上用1表示，反面向上用0表示。
import random
l=[random.randint(0,1) for i in range(10000)]  #产生10000个随机数，值为0或1
print(sum(l) / len(l))
```

程序输出：

```
0.5006
```

随机函数库（续6）

例3-5 随机产生8位密码，密码由数字和字母组成。

程序代码：

```
 import random

 digits=[chr(i) for i in range(48,58)]
 ascii_letters=[chr(i) for i in range(65,91)]+[chr(i) for i  in range(97,123)]
# 数字的个数随机产生
num_of_numeric = random.randint(1,7)
# 剩下的都是字母
num_of_letter = 8 - num_of_numeric
# 随机生成数字
numerics = [random.choice(digits) for i in range(num_of_numeric)]
```

随机函数库（续7）

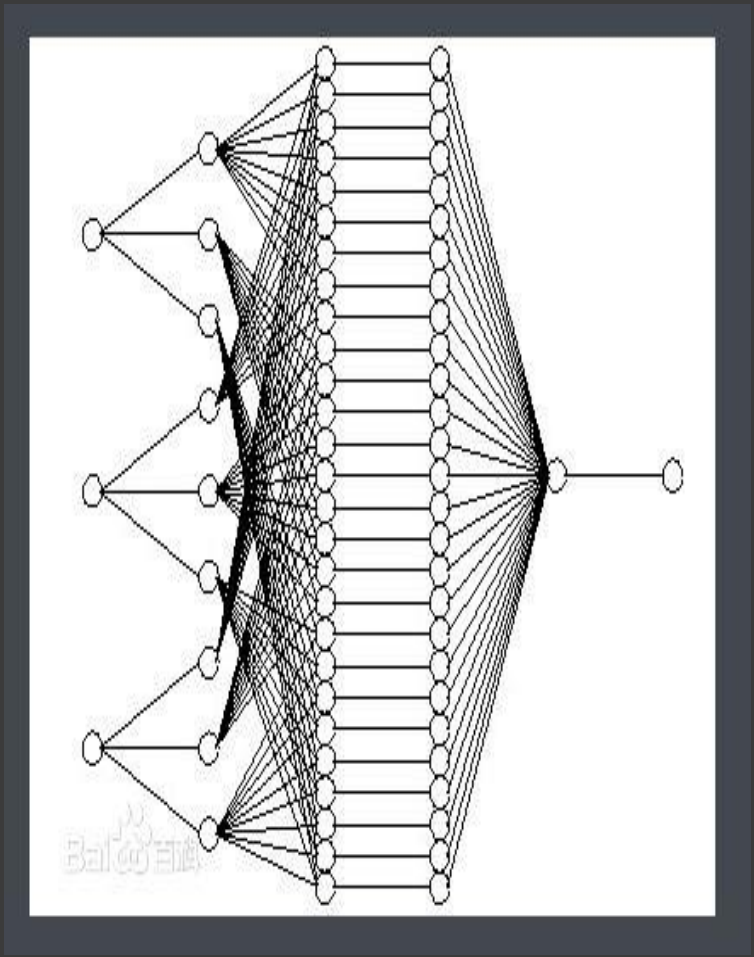
```
# 随机生成字母
letters = [random.choice(ascii_letters) for i in range(num_of_letter)]
# 结合两者
all_chars = numerics + letters
# 重新排列
random.shuffle(all_chars)
# 生成最终字符串
result = ''.join([i for i in all_chars])
print(result)
```

程序输出：

GqG5B429

输入一行字符串，并将它转换成10进制数输出。

- 神经网络分层



- 如输入：　_ahg1*B
- 新字符串：a1B
- 10进制数：2587

```
s=input()      #第一层，输入层

#第二层
lst=[t for t in s if
ord('0')<=ord(t)<=ord('9') \
or ord('a')<=ord(t)<=ord('f')        \
or ord('A')<=ord(t)<=ord('F')]

#第三层
news=''.join(lst)

#第四层  输出层
print(news)
print(int(news,16))
```

输入字符串，排序后按从小到大输出每个字符及该字符在原字符串中的索引。

- s=input()
- lst=[(s[index],index) for index in range(len(s))]
- lst.sort()
- print(lst)

- (s[index],index)是一个元组，保存输入的字符和它的位置。
- 程序输入
- hello python
-
- 程序输出
- [(' ', 5), ('e', 1), ('h', 0), ('h', 9), ('l', 2), ('l', 3), ('n', 11), ('o', 4), ('o', 10), ('p', 6), ('t', 8), ('y', 7)]

列表的复制

- a=[1,2,3,4]
- b=a
- print(id(a),id(b))
- b[2]=5
- print(a)
-
- c=a.copy()
- print(id(a),id(c))
- c[2]=6
- print(a)

- d=a[:]
- print(id(a),id(d))
- d[2]=7
- print(a)

列表其他注意点

- 1．如何判断一个列表lst为空？
- len(lst)==0
- not lst ==True

- 2如何将列表lst切成相同长度n的序列？
- [lst[i:i+n] for i in range(0, len(l), n)]

- 遍历list的时候修改某些元素？
-
-
- seq=[1,2,7,3,4,3,2,7,4,5,6,5,4,3]
-
- seq[:]=[x for x in seq if seq.count(x)<2]
- print(seq)