容器

- 列表(list)
- 元祖(tuple)
- 字典(dict)
- 集合(set)

列表

```
void list = []/void list = list()
a = [1, 2, 3, 4]
b = [‘dog’, ‘bird’, ‘cat’, ‘cow’, ‘Crocodile’]
c = [1, ‘dog’, 3.5, 4, ‘kinkin’, False]
```

获取元素方法：列表名[索引值]

例：a[0] b[4] c[2]

列表的基础操作

```
arr = [3, 10, 7, 29, 30, 8]
```

- 追加 append
- 插入 insert
- 删除 remove/pop
- 排序 sort
- 倒序 reverse
- 清空 clear

列表的遍历(迭代)

```
arr = [3, 10, 7, 29, 30, 8]
```

- 索引遍历 `for idx in range(len(arr)):`
- 内容遍历 `for num in arr:`

*range

*列表的切片

arr = [1, 2, 3, 4, 5, 6, 7, 8]
索引(0, 1, 2, 3, 4, 5, 6, 7)或(–8, –7, –6, –5, –4, –3, –2, –1)
arr[i:j:k]
i为起始值start，j为终止值end，k为步长step。
[start,end)为前闭后开的区间
k>0时，i缺省值为0，j缺省值为len(arr) – 1
k<0时，i缺省值为–1，j缺省值为–len(arr) – 1
k的缺省值为1

字典

特点：

1. 键值对，数据结构：散列表（hash），查找快

2. 键只能是基本数据类型，值可以是任意类型，键的类型可以不同

```
void_dict = dict()
d = {'apple':3, 'banana':10, 'pineapple':7}
```

获取值方法：字典名[键值]
例：d['apple']

字典的基础操作

```
d = {'apple':3, 'banana':10, 'pineapple':7}
```

- 列出一个键值对的view：`items`
- 列出key的view：`keys`
- 列出value的view：`values`
- 增加/赋值修改 `d[key] = value`
- 删除 `pop`
- 清空 `clear`

字典的遍历(迭代)

```
d = {'apple':3, 'banana':10, 'pineapple':7}
```

- 键值遍历 `for key in d:/for key in d.keys():`
- 内容遍历 `for value in d.values():`
- 键值对遍历 `for k,v in d.items():`

集合

特点：

1. 一个容器，集合的元素不重复
2. 集合的操作：并集，交集，差集
3. 集合判断：是否是子集，是否有交集

```
void_set = set()
x = set([1, 1, 2, 2, 3, 3])
y = set(“jinsheng”)
z = set(['d', 'i', 'm', 'i', 't', 'e'])
```

集合的基础操作

```
x = set([a, p, l, e, g, q])
y = set([a, i, k, n, g])
```

- 交集 `x & y`
- 合集 `x | y`
- 差集 `x - y`
- 对称差(并–交) `x ^ y`
- 增加 `add`
- 删除 `remove/discard/pop`
- 清空 `clear`

练习

- 已知列表l = [1, 1, 6, 3, 1, 5, 2] ，删除列表中的重复元素
- 下面代码输出结果

```
def f(x,l=[]):
    for i in range(x):
        l.append(i*i)
    print(l)

f(2)
f(3,[3,2,1])
f(3)
```

- 合并两个list：l1 = [2,3,8], l2 = [5,6,10]

```python
#第一种方式
l=[1,1,6,3,1,5,2]
list(set(l))

#第二种方式
l=[1,1,6,3,1,5,2]
def duplicate(lists):
    L=[]
    for i in lists:
        if i not in L:
            L.append(i)
    return L
print(duplicate(l))
```

```python
#第一种方式
list1 = [2,3,8]
list2 = [5,6,10]

def list_union(list1,list2):
    for i in list2:
        list1.append(i)
    return(list1)
list_union(list1,list2)

#第二种方式
list1+list2
```

猜数字

一个人出数字，一方猜。出数字的人要先想好一个没有重复数字的4位数，例:8123，不能让猜的人知道。猜的人就可以开始猜。每猜一个数，出数者就要根据这个数字给出几A几B，例猜1562，则为 0A2B ，其中A前面的数字表示位置正确的数的个数，而B前的数字表示数字正确而位置不对的数的个数。接着猜的人再根据出题者的几A几B继续猜，直到猜中为止。

猜数字

一星难度：
写定一个4位数，自己猜。
二星难度：
程序随机生成谜底（无重复数字的4位数），自己猜。
三星难度：
增加多人模式，输入玩家人数，开启游戏。轮流猜，最先猜对的人为赢家。

注意：玩家的输入是自由的，需要判断玩家输入合法性