集合是可变对象，支持原处修改

例：程序set\_test/set\_test1.py

x = set('spaam')  
y = {'h', 'a', 'm'}  
  
# 集合是无序的，是不可变对象的集合

# 集合本身是可变对象  
# 集合会自动删除重复的元素  
print(x)  
print(y)  
print()  
print(x & y)  
print(x | y)  
print(x - y)  
print()  
  
# 集合解析  
s = {x \*\* 2 for x in [1, 2, 3, 4]}  
print(s)

输出为：

{'s', 'p', 'a', 'm'}

{'h', 'a', 'm'}

{'a', 'm'}

{'s', 'h', 'p', 'a', 'm'}

{'s', 'p'}

{16, 1, 4, 9}

可hash：字符串，数字，frozenset, tuple等不可变对象

不可hash：list, set, dict等可变对象

参考：<https://blog.csdn.net/qq_25730711/article/details/53487350>

例：程序set\_test/set\_test2.py

# 集合只能包含不可变(可散列的)对象类型  
s = {1.23}  
# s.add([1, 2, 3]) # Error, not list  
# s.add({'a': 1}) # Error, not dict  
s.add((1, 2, 3))  
print(s) # {1.23, (1, 2, 3)}  
  
# 集合本身不可散列  
s1 = {1, 2, 3}  
# s2 = {4, 5, s1} # Error, not set  
s2 = {4, 5, frozenset(s1)}  
print(s2) # {frozenset({1, 2, 3}), 4, 5}  
print()  
  
# 集合不支持下标访问  
# print(s1[0]) # Error  
for item in s2:  
 if isinstance(item, frozenset):  
 for sub\_item in item:  
 print(sub\_item)  
 else:  
 print(item)

def set\_changeable(set1):  
 set1.add(3)  
  
  
s3 = {1, 2}  
print('original set: ', s3) # {1, 2}  
set\_changeable(s3)  
print('set\_changealbe set: ', s3) # {1, 2, 3}

输出为：

{1.23, (1, 2, 3)}

{frozenset({1, 2, 3}), 4, 5}

1

2

3

4

5

例：程序set\_test/set\_test3.py

s1 = set([3, 1, 2, 4.5])  
print(s1.pop()) # 删除s1中的一个元素，随机的，因为集合没有顺序  
  
s2 = set([3, 1, 2, 4.5])  
s3 = set([3, 11, 22, 33, 4.5])  
s4 = s2.union(s3) # 并集  
print(s4) # {1, 2, 3, 4.5, 33, 11, 22}  
  
s5 = s2.intersection(s3) # 交集  
print(s5) # {3, 4.5}  
  
s6 = s2.difference(s3) # s2中在s3没有的  
print(s6) # {1, 2}  
  
s7 = s3.difference(s2) # s3中在s2没有的  
print(s7) # {33, 11, 22}

# 集合中可以包含不同类型的元素  
# 但是与list类似，使用某些函数  
# 时有可能会出现Error  
s8 = {1, 2, 3, 'spam'}  
print(s8)