Python核心数据类型-集合

集合的特点 确定性 互异性 无序性

集合是可迭代的 可以根据需要增长或缩短

集合的创建

x = set([1, 2, 3])

set(l) ##将列表l转化为字典

len() ##返回集合的长度

#判断元素是否在集合中
In [80]: x = set([1,2,3,4])

In [81]: 1 in x
Out[81]: True

In [82]: 5 in x
Out[82]: False

集合的常用方法
In [1]: a = set([1, 2, 3, 4, 5])
In [2]: b = set([1, 2, 3])

1. add 向集合中添加新的元素
n [6]: a
Out[6]: set([1, 2, 3, 4, 5])
In [7]: a.add(6)
In [8]: a.add('a')

In [9]: a
Out[9]: set(['a', 1, 2, 3, 4, 5, 6]) #集合中添加了‘a’与 6

2.clear 删除集合中的所有元素
In [10]: a
Out[10]: set(['a', 1, 2, 3, 4, 5, 6])
In [11]: a.clear()
In [12]: a
Out[12]: set([])

3.copy复制集合
In [13]: b
Out[13]: set([1, 2, 3])
In [14]: a.copy(b)
In [16]: a
Out[16]: set([1, 2, 3])

4.difference 求两个集合的不同部分
In [22]: a
Out[22]: set([1, 2, 3, 4])
In [23]: b
Out[23]: set([1, 2, 3])
In [24]: a.difference(b) #返回a中有但是b中没有的元素
Out[24]: set([4])

difference_update #求两个集合的不同部分并修改原集合

5.intersection 求交集
In [25]: a
Out[25]: set([1, 2, 3, 4])
In [26]: b
Out[26]: set([1, 2, 3])
In [27]: a.intersection(b)
Out[27]: set([1, 2, 3])

intersection_update 求两个集合的交集 并修改原集合

6. union 求并集 不修改原集合

```
In [28]: a
Out[28]: set([1, 2, 3, 4])
In [29]: b
Out[29]: set([1, 2, 3])
In [30]: a.union(b)
Out[30]: set([1, 2, 3, 4])
```

7.pop 删除集合中的任意一个元素

```
In [37]: a = set(['a', 'b', 'c', 1, 2, 3, 4, 5])
In [38]: a
Out[38]: set(['a', 1, 'c', 'b', 4, 5, 2, 3])
In [39]: a.pop()
Out[39]: 'a'
```

8.remove 删除集合中的指定元素 如果删除一个没有的元素则引发异常

```
In [42]: a
Out[42]: set(['b', 4, 5, 2, 3])
In [43]: a.remove(5)
In [44]: a.remove('b')
In [45]: a
Out[45]: set([4, 2, 3])
```

a.discard() 删除没有的元素不会引发异常

9 update 向集合中添加多个元素 add是添加一个元素 并且修改了原集合

```
In [46]: a
Out[46]: set([4, 2, 3])
In [47]: a.update([4, 5, 6])
In [48]: a
Out[48]: set([4, 5, 6, 2, 3])
```

10. issubset #测试一个集合中的元素是否都在另一个集合中

#测试一个集合是否小于等于另一个集合　测试一个集合是不是另一个集合的子集
In [83]: a
Out[83]: set([2, 3])
In [84]: b
Out[84]: set([1, 2, 3])
In [85]: a.issubset(b)
Out[85]: True
In [86]: b.issubset(a)
Out[86]: False

11. issuperset #测试一个集合是否大于等于另一个集合
测试一个集合是不是另一个集合的全集
In [87]: a
Out[87]: set([2, 3])
In [88]: b
Out[88]: set([1, 2, 3])
In [89]: a.issuperset(b)
Out[89]: False
In [90]: b.issuperset(a)
Out[90]: True

12. symmetric_difference　#返回a b中不重复的元素　与 difference 是有区别的
In [96]: a
Out[96]: set([2, 3])
In [97]: b
Out[97]: set([1, 2, 3])

In [98]: a.difference(b) #返回a中有但是b中没有的元素
Out[98]: set([])

In [99]: b.difference(a) #b中有但是a中没有的元素
Out[99]: set([1])

In [100]: a
Out[100]: set([2, 3])

In [101]: b
Out[101]: set([1, 2, 3])

In [102]: a.symmetric_difference(b)
Out[102]: set([1])

In [103]: b.symmetric_difference(a)
Out[103]: set([1])

symmetric_difference_update #返回a b中不重复的元素 并修改原来的列表

a.isdisjoint 如果两个集合没有交集则返回true

a = set([1,2,3])
b = set([2, 3, 4])
c = set(['5'])

In [3]: a.isdisjoint(b)
Out[3]: False

In [5]: a.isdisjoint(c)
Out[5]: True