Python核心数据类型--集合

一.集合基础

1. 集合的特点:
   确定性 互异性 无序性
2. 集合是可迭代的
3. 集合可以根据需要增长或缩短
4. 集合的创建: 可通过以下两种方式创建集合

In [1]:
```
x = set([1, 2, 3])
#或
x = {1, 2, 3, 4, 5}
```

5. 元组 列表 字符串 转换为集合

In [2]:
```
x = 'apple'
set(x)
```
Out[2]:
```
{'a', 'e', 'l', 'p'}
```

In [3]:
```
y = [1,2,3,4]
set(y)
```
Out[3]:
```
{1, 2, 3, 4}
```

In [4]:
```
z = (1,2,3,4,5)
set(z)
```
Out[4]:
```
{1, 2, 3, 4, 5}
```

6. 使用len()函数可以获取集合的长度

In [5]:
```
x = set([1,2,3,4,5])
print len(x)
```
```
5
```

In [6]:
```
#示例代码1. 集合中元素确定性的体现
x = {1,2,3,4}
print 1 in x
print 5 in x
```
```
True
False
```

In [7]:
```
#示例代码2.集合中元素互异性的体现
x = {1,1,1,1,1,2,3}
print x
```
```
set([1, 2, 3])
```

In [8]:
```
#示例代码3.集合中元素无序性的体现
x = set([])
x.add('x')
x.add('a')
x.add(1)
print x
```
```
set(['a', 'x', 1])
```

二.集合的常用方法

In [1]:
```
print(dir(set([])))
```
```
['__and__', '__class__', '__contains__', '__delattr__', '__dir__', '__doc__', '__eq__', '__format__', '__ge__', '__getattribute__', '__gt__', '__hash__', '__iand__',
'__init__', '__init_subclass__', '__ior__', '__isub__', '__iter__', '__ixor__', '__le__', '__len__', '__lt__', '__ne__', '__new__', '__or__', '__rand__', '__reduce_
_', '__reduce_ex__', '__repr__', '__ror__', '__rsub__', '__rxor__', '__setattr__', '__sizeof__', '__str__', '__sub__', '__subclasshook__', '__xor__', 'add', 'clear',
'copy', 'difference', 'difference_update', 'discard', 'intersection', 'intersection_update', 'isdisjoint', 'issubset', 'issuperset', 'pop', 'remove', 'symmetric_diff
erence', 'symmetric_difference_update', 'union', 'update']
```

In [19]:
```
#1.add() 向集合中添加元素
help(x.add)
print x
x.add(10)
x.add(20)
print x
```
```
Help on built-in function add:

add(...)
    Add an element to a set.

    This has no effect if the element is already present.

set(['a', 'x', 1])
set(['a', 'x', 10, 20, 1])
```

In [21]:
```
#2.pop() 从集合中随机删除一个元素, 并返回这个元素
help(x.pop)
print x
x.pop()
x.pop()
print x
```
```
Help on built-in function pop:

pop(...)
    Remove and return an arbitrary set element.
    Raises KeyError if the set is empty.

set(['a', 'x', 10, 20, 1])
set([10, 20, 1])
```

In [28]:

```
#3.remove() 从集合中删除指定的元素，如果删除的元素不在集合中，则报key-error错误
help(x.remove)
print x
x.remove(10)
print x
x.remove(100)
```

```
Help on built-in function remove:

remove(...)
    Remove an element from a set; it must be a member.

    If the element is not a member, raise a KeyError.

set([20, 1])

---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
<ipython-input-28-253f8b4bc226> in <module>()
      2 help(x.remove)
      3 print x
----> 4 x.remove(10)
      5 print x
      6 x.remove(100)

KeyError: 10
```

In [32]:

```
#4.discard() 从集合中删除元素，如果删除的元素不在集合中，不报任何错误
help(x.discard)
print x
x.discard(20)
print x
x.discard(1000)
```

```
Help on built-in function discard:

discard(...)
    Remove an element from a set if it is a member.

    If the element is not a member, do nothing.

set([1])
set([1])
```

In [34]:

```
#5.clear() 清空集合中的元素
help(x.clear)
x.clear()
print x
```

```
Help on built-in function clear:

clear(...)
    Remove all elements from this set.

set([])
```

In [38]:

```
#6.copy() 对集合进行浅拷贝
help(x.copy)
x = {1, 2, 3}
y = x.copy()
print x
print y
```

```
Help on built-in function copy:

copy(...)
    Return a shallow copy of a set.

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

In [40]:

```
#7.update() 将一个集合中的所有元素添加到另外一个集合中
help(x.update)
x = {1,2,3}
y = {2,3,4}
x.update(y)
print x
```

```
Help on built-in function update:

update(...)
    Update a set with the union of itself and others.

set([1, 2, 3, 4])
```

In [61]:

```
#8.issubset 判断一个集合是不是另一个集合的子集
help(x.issubset)
x = {1,2,3}
y = {1,2,3,4}
z = {2,3,4,5}

print x.issubset(y)
print x.issubset(z)
```

```
Help on built-in function issubset:

issubset(...)
    Report whether another set contains this set.

True
False
```

In [62]:

```
#9.issuperset 判断一个集合是不是另一个集合的全集
help(x.issuperset)
x = {1,2,3}
y = {1,2,3,4}
z = {2,3,4,5}

print y.issuperset(x)
print z.issuperset(x)
```

```
Help on built-in function issuperset:

issuperset(...)
    Report whether this set contains another set.

True
False
```

In [63]:

```
#10.isdisjoint 如果两个集合没有交集则返回true
help(x.isdisjoint)
x = {1,2,3}
y = {2,3,4}
z = {5, 6, 7}

print x.isdisjoint(y)
print x.isdisjoint(z)
```

```
Help on built-in function isdisjoint:

isdisjoint(...)
    Return True if two sets have a null intersection.

False
True
```

In [64]:

```
#11.intersection 求两个集合的交集
help(x.intersection)
x = {1,2,3}
y = {2,3,4}
print x.intersection(y)
print x
```

```
Help on built-in function intersection:

intersection(...)
    Return the intersection of two or more sets as a new set.

    (i.e. elements that are common to all of the sets.)

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

In [65]:

```
#12.intersection_update 求两个集合的交集并修改原集合
help(x.intersection_update)
x = {1,2,3}
y = {2,3,4}
x.intersection_update(y)
print x
```

```
Help on built-in function intersection_update:

intersection_update(...)
    Update a set with the intersection of itself and another.

set([2, 3])
```

In [68]:

```
#13.union 求两个集合的并集
help(x.union)
x = {1,2,3}
y = {2,3,4}
x.union(y)
print x
```

```
Help on built-in function union:

union(...)
    Return the union of sets as a new set.

    (i.e. all elements that are in either set.)

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

In [70]:

```
#14.difference 一个集合和另一个集合相比较，不同的元素有哪些
help(x.difference)
x = {1,2,3}
y = {2,3,4}
print x.difference(y)
print y.difference(x)
```

```
Help on built-in function difference:

difference(...)
    Return the difference of two or more sets as a new set.

    (i.e. all elements that are in this set but not the others.)

set([1])
set([4])
```

In [72]:

```
#15.difference_update 一个集合和另一个集合相比较，不同的元素有哪些，并修改原集合
help(x.difference_update)
x = {1,2,3}
y = {2,3,4}
x.difference_update(y)
print x
```

```
Help on built-in function difference_update:

difference_update(...)
    Remove all elements of another set from this set.

set([1])
```

In [75]:

```
#16.symmetric_difference 返回两个集合交集之外的元素
help(x.symmetric_difference)
x = {1,2,3}
y = {2,3,4}
print x.symmetric_difference(y)
```

```
Help on built-in function symmetric_difference:

symmetric_difference(...)
    Return the symmetric difference of two sets as a new set.

    (i.e. all elements that are in exactly one of the sets.)

set([1, 4])
```

In [77]:

```
#17.symmetric_difference_update 返回两个集合交集之外的元素，并修改原集合
help(x.symmetric_difference_update)
x = {1,2,3}
y = {2,3,4}
x.symmetric_difference_update(y)
print x
```

```
Help on built-in function symmetric_difference_update:

symmetric_difference_update(...)
    Update a set with the symmetric difference of itself and another.

set([1, 4])
```