Module 04c. Data Structures (Sets)

September 26, 2018

1 Sets

Python includes a data type for sets. A set is an unordered collection with no duplicate elements. Basic uses include membership testing and eliminating duplicate entries. Set objects also support mathematical operations like union, intersection, difference, and symmetric difference.

Curly braces or the set() function can be used to create sets. Note: to create an empty set you have to use set()

1.0.1 Creating a set

```
In [26]: s = set() # empty set
         type(s)
```

```
Out[26]: set
```

1.0.2 Updating sets

```
In [5]: a= {1,2,3,4, 'a', (1,2,3)} # Sets can contain only immutable objects
        print (a)
```

```
{'a', 2, 3, 4, 1, (1, 2, 3)}
```

```
In [6]: a.add(5)
        print (a)
```

```
{'a', 2, 3, 4, 1, 5, (1, 2, 3)}
```

```
In [7]: a.update([6,7])
        print (a)
```

```
{'a', 2, 3, 4, 1, 5, 6, 7, (1, 2, 3)}
```

```
In [8]: a.pop()
        print (a)
```

```
{2, 3, 4, 1, 5, 6, 7, (1, 2, 3)}
```

```
In [9]: a.remove(4)
        print (a)

{2, 3, 1, 5, 6, 7, (1, 2, 3)}
```

1.0.3 Operations on sets

```
In [17]: a = [1, 1, 2, 3, 5, 8, 13]
         b = [1, 2, 3, 4, 5, 6, 7]

         # Converting a list into set (this will also remove the duplicates inside the list)
         set_a = set(a)
         set_b = set(b)

In [18]: # Membership Operator :
         1 in set_a

Out[18]: True

In [19]: # set intersection :
         print(set_a & set_b)

         # OR

         print (set_a.intersection(set_b))

{1, 2, 3, 5}
{1, 2, 3, 5}


In [20]: # Set Union
         print(set_a | set_b)

         # OR

         print (set_a.union(set_b))

{1, 2, 3, 4, 5, 6, 7, 8, 13}
{1, 2, 3, 4, 5, 6, 7, 8, 13}


In [21]: # symmetric difference or uncommon elements
         print(set_a ^ set_b)

         # OR

         print (set_a.symmetric_difference(set_b))
```

```
{4, 6, 7, 8, 13}
{4, 6, 7, 8, 13}
```

```
In [22]: # set difference
         print(set_a - set_b)

         # OR

         print (set_a.difference(set_b))
```

```
{8, 13}
{8, 13}
```

1.1 Some more Set functions

```
In [15]: a= {1,2,3,4}
         print (a)

         b=set([1,4])
         print (b)

         print (a.issubset(b))
         print (b.issubset(a))
         print (a.issuperset(b))
         print (b.issuperset(a))
```

```
{1, 2, 3, 4}
{1, 4}
False
True
True
False
```

1.1.1 Set Comprehensions vs List Comprehensions

```
In [49]: a = {x for x in 'abrdacradabra' if x not in 'abc'}
         print(a)
```

```
{'r', 'd'}
```

```
In [48]: a = [x for x in 'abrdacradabra' if x not in 'abc']
         print(a)
```

```
['r', 'd', 'r', 'd', 'r']
```

1.1.2 Looping

```
In [9]: basket = ['apple', 'orange', 'apple', 'pear', 'orange', 'banana']
        for f in sorted(set(basket)):
            print(f)
```

```
apple
banana
orange
pear
```

2 Immutable Sets

Frozensets can be created using the function frozenset(). This datatype supports methods like copy(), difference(), intersection(), isdisjoint(), issubset(), issuperset(), symmetric_difference() and union(). Being immutable it does not have method that add or remove elements.

```
In [2]: x = frozenset([1, 2, 3, 4, 5])
        x
```

```
Out[2]: frozenset({1, 2, 3, 4, 5})
```