

Set

date: 4/11/2020, 7:41:15 AM

- Unordered collection of unique elements.
- is mutable. (Elements can be removed and added to the sets)
- elements in a set must be immutable.

```
1  >>> p = {5,324,123,23142123}
2  >>> p
3  {123, 324, 5, 23142123} #set is unordered
4  >>> type(p)
5  <class 'set'>
6  >>> d = {} #empty curly braces creates a dictionary
7  >>> type(d)
8  <class 'dict'>
9  >>> e = set() #for creating an empty set, use set constructor
10 >>> e
11 set() #python echos back to us.
```

Set constructor can create a set from any iterable series, such as list

PS. Duplicates are discarded

```
1  >>> s = set([1,2,3,2,4,3])
2  >>> s
3  {1,4,3,2}
```

Common use of them is efficiently remove duplicate items from a series of objects.

Sets are iterable, but the order is arbitrary

```
1
2  >>> for x in {1,2,4,8,16,32}:
3  ...     print(x)
4  ...
5  32
6  1
7  2
8  4
9  8
10 16
```

Membership (in and not in) is fundamental operation for sets:

```
1  >>> q= {2,9,6,4,2,4,5,7,8}
2  >>> q
3  {2, 4, 5, 6, 7, 8, 9}
```

```
4 >>> 3 in q
5 False
6 >>> 2 in q
7 True
8 >>> 3 not in q
9 True
10
```

Adding to the set:

Single element:

```
1 >>> k = {1,2,3}
2 >>> k.add(54)
3 >>> k
4 {1, 2, 3, 54}
```

PS. Adding element that is already exist has no effect on set, and not produces error.

Multiple elements:

```
1 >>> k.update([1,2,4,7,5,4564,3453654,98])
2 >>> k
3 {1, 2, 3, 4, 5, 98, 7, 4564, 54, 3453654}
```

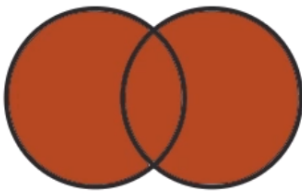
Removing from the set:

1. `.remove()` — requires element to be removed is present in the set, or it will throw `Key Error` is produced.
2. `.discard()` — has no effect if element not in the set.

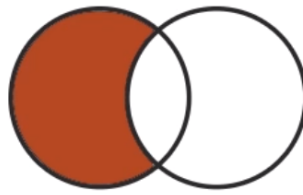
```
1 >>> k
2 {1, 2, 3, 4, 5, 98, 7, 4564, 54, 3453654}
3 >>> k.remove(2)
4 >>> k
5 {1, 3, 4, 5, 98, 7, 4564, 54, 3453654}
6 >>> k.remove(345678234123)
7 Traceback (most recent call last):
8   File "<stdin>", line 1, in <module>
9   KeyError: 345678234123
10 >>> k.discard(345678234123)
11 >>> k.discard(3)
12 >>> k
13 {1, 4, 5, 98, 7, 4564, 54, 3453654}
14
```

Set algebra operations:

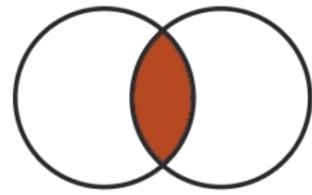
Set Algebra



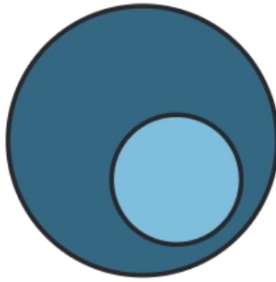
union



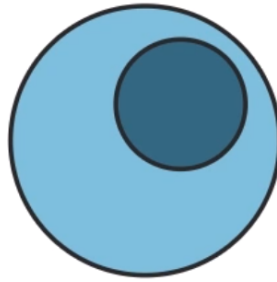
difference



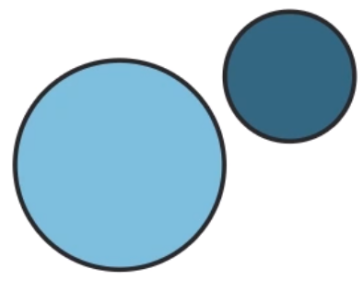
intersection



subset



superset



disjoint

```
1 >>> group_mates = {"Kamil", "Azer", "Linus", "Robert"}
2 >>> university_people = {"Sahil", "Kamil", "Imran"}
3 >>> group_mates.union(university_people)
4 {'Sahil', 'Robert', 'Azer', 'Kamil', 'Linus', 'Imran'}
5 >>> group_mates.difference(university_people)
6 {'Robert', 'Linus', 'Azer'}
7 >>> group_mates.intersection(university_people)
8 {'Kamil'}
9 >>> group_mates.union(university_people) == university_people.union(group_mates)
10 True
```