

# Sets

Sets are a type of collection. This means that like lists and tuples, you can input different Python types.

Unlike lists and tuples, they are unordered. This means sets do not record element position.

Sets only have unique elements. This means there is only one of a particular element in a set.

To define a set, you use curly brackets. You place the elements of a set within the curly brackets. You notice there are duplicate items.

## Sets: Creating a Set

```
Set1={"pop", "rock", "soul", "hard rock", "rock", "R&B", "rock", "disco" }
```

```
Set1: {'rock', 'R&B', 'disco', 'hard rock', 'pop', 'soul'}
```

When the actual set is created, duplicate items will not be present.

You can convert a list to a set by using the function set, this is called type casting.

## Sets: Creating a Set

```
album_list = ["Michael Jackson", "Thriller", "Thriller", 1982]
```

```
album_set = set(album_list)
```

```
album_set : {'Michael Jackson', 'Thriller', 1982}
```

set()

album\_set

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

You simply use the list as the input to the function set. The result will be a list converted to a set. Let's go over an example.

We start off with a list. We input the list to the function set. The function set returns a set.

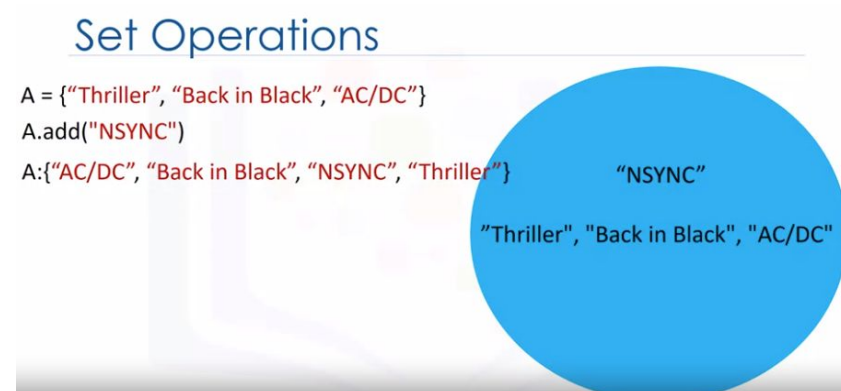
Notice how there are no duplicate elements.

Let's go over set operations. These could be used to change the set.

Consider the set A. Let's represent this set with a circle. If you are familiar with sets, this could be part of a Venn diagram. A Venn diagram is a tool that uses shapes usually to represent sets.

### Add Method

We can add an item to a set using the add method. We just put the set name followed by a dot, then the add method. The argument is the new element of the set we would like to add, in this case, NSYNC.



The set A now has in NSYNC as an item. If we add the same item twice, nothing will happen as there can be no duplicates in a set.

### Remove Method

Let's say we would like to remove NSYNC from set A. We can also remove an item from a set using the remove method. We just put the set name followed by a dot, then the remove method.

The argument is the element of the set we would like to remove, in this case, NSYNC. After the remove method is applied to the set, set A does not contain the item NSYNC. You can use this method for any item in the set.



### “In” command

We can verify if an element is in the set using the in command as follows. The command checks that the item, in this case AC/DC, is in the set.

## Set Operations

A:{"AC/DC", "Back in Black", "Thriller"}

"AC/DC" in A

"Thriller", "Back in Black", "AC/DC"

If the item is in the set, it returns **true**. If we look for an item that is not in the set, in this case for the item Who, adds the item is not in the set, we will get a **false**.

These are types of mathematical set operations. There are other operations we can do.

There are lots of useful mathematical operations we can do between sets. Let's define the set album set one. We can represent it using a red circle or Venn diagram.

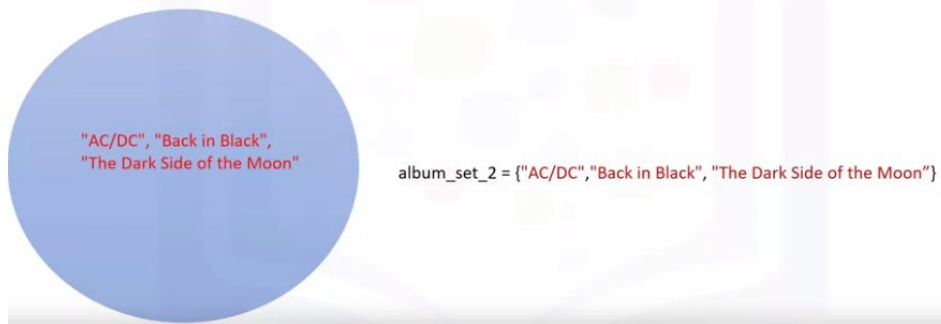
## Sets: Mathematical set operations

"AC/DC", "Back in Black", "Thriller"

album\_set\_1 = {"AC/DC", "Back in Black", "Thriller"}

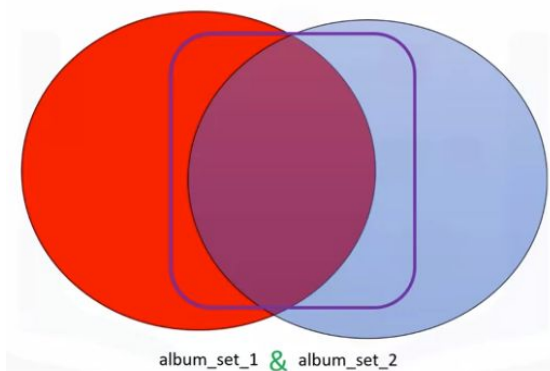
Similarly, we can define the set album set two. We can also represent it using a blue circle or Venn diagram. The intersection of two sets is a new set containing elements which are in both of those sets. It's helpful to use Venn diagrams.

## Sets: Mathematical set operations

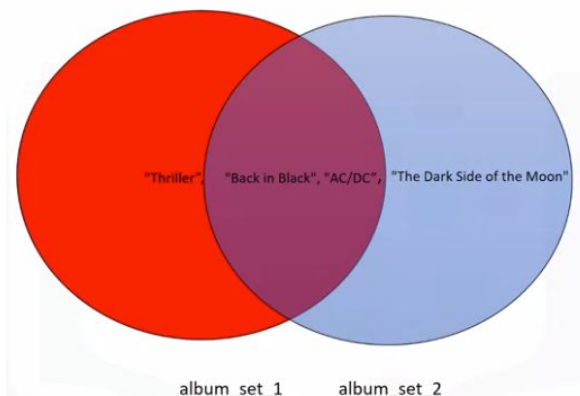


The two circles that represent the sets combine, the overlap, represents the new set.

As the overlap is comprised with the red circle and blue circle, **we define the intersection in terms of and**.



In Python, we use an ampersand to find the union of two sets. If we overlay the values of the set over the circle placing the common elements in the overlapping area, we see the correspondence.



After applying the intersection operation, all the items that are not in both sets disappear.

In Python, we simply just place the ampersand between the two sets.

```
album_set_1={'AC/DC','Back in black', 'Thriller'}  
album_set_2={'AC/DC','Back in black', 'Dark side of the moon'}
```

```
album_set_3=album_set_1&album_set_2
```

```
Album_set_3 = {'AC/DC', 'Back in black'}
```

Question:

What is the result of the following operation :  $\{'a','b'\} \& \{'a'\}$

$\{'a','b'\} \& \{'a'\} = \{'a'\}$

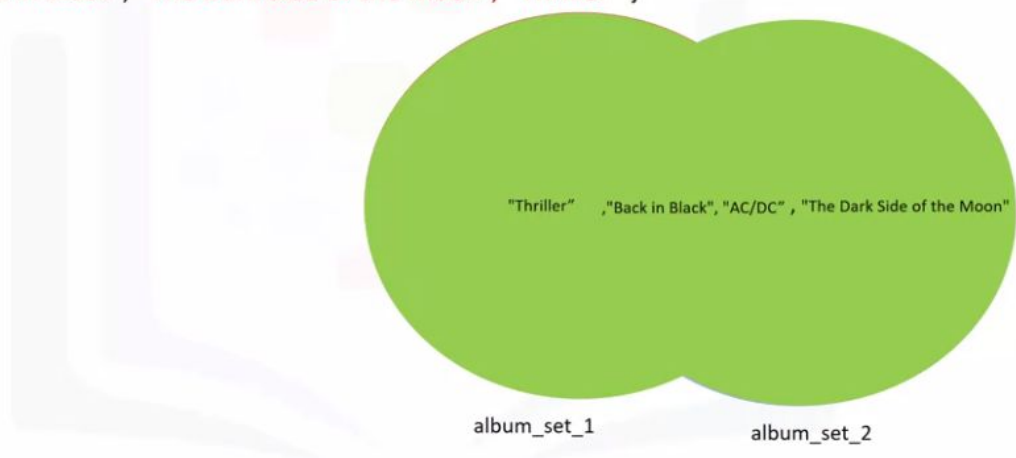
We see that both AC /DC and Back in Black are in both sets. The result is a new set album; set three, containing all the elements in both albums set one and album set two.

## UNION

The union of two sets is the new set of elements which contain all the items in both sets. We can find the union of the sets album set one and album set two as follows.

```
album_set_1.union(album_set_2)
```

```
{'AC/DC', 'Back in Black', 'The Dark Side of the Moon', 'Thriller' }
```

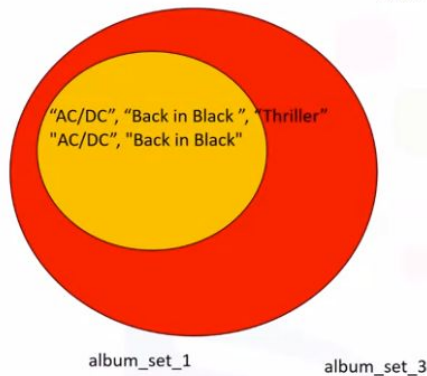


The result is a new set that has all the elements of album set one and album set two. This new set is represented in green.

Consider the new album set, album set three. The set contains the elements AC/DC and Back in Black. We can represent this with a Venn diagram, as all the elements and album

set three are in album set one.

```
album_set_1 = {"AC/DC", "Back in Black ", "Thriller "}
album_set_3 = {"AC/DC", "Back in Black "}
```



The circle representing album set one encapsulates the circle representing album set three.

We can check if a set is a subset using the `issubset` method.

```
album_set_3.issubset(album_set1)
```

As album set three is a subset of the album set one, the result is true. There is a lot more you can do with sets. Check out the lab for more examples.