Sets in Python

Estimated time needed: **20** minutes

Objectives

After completing this lab you will be able to:

- Work with sets in Python, including operations and logic operations.

Table of Contents

Sets

Set Content

A set is a unique collection of objects in Python. You can denote a set with a pair of curly brackets {}. Python will automatically remove duplicate items:

In [1]:
```
# Create a set

set1 = {"pop", "rock", "soul", "hard rock", "rock", "R&B", "rock", "disco"}
set1
```

Out[1]:
```
{'R&B', 'disco', 'hard rock', 'pop', 'rock', 'soul'}
```

The process of mapping is illustrated in the figure:

You can also create a set from a list as follows:

In [2]:
```
# Convert list to set

album_list = ["Michael Jackson", "Thriller", 1982, "00-42-19", \
              "Pop, Rock, R&B", 46.0, 65, "30-Nov-82", None, 10.0]
album_set = set(album_list)
album_set
```

Out[2]:
```
{'00-42-19',
 10.0,
 1982,
 '30-Nov-82',
 46.0,
 65,
 None,
 'Michael Jackson',
 None,
 'Pop, Rock, R&B',
 'Thriller'}
```

Now let us create a set of genres:

In [3]:
```
# Convert list to set

music_genres = set(["pop", "pop", "rock", "folk rock", "hard rock", "soul", \
                    "progressive rock", "soft rock", "R&B", "disco"])
music_genres
```

Out[3]:
```
{'R&B',
 'disco',
 'folk rock',
 'hard rock',
 'pop',
 'progressive rock',
 'rock',
 'soft rock',
 'soul'}
```

Set Operations

Let us go over set operations, as these can be used to change the set. Consider the set **A**

In [4]:
```
# Sample set

A = set(["Thriller", "Back in Black", "AC/DC"])
A
```

Out[4]:
```
{'AC/DC', 'Back in Black', 'Thriller'}
```

We can add an element to a set using the `add()` method:

In [5]:
```
# Add element to set

A.add("NSYNC")
A
```

Out[5]:
```
{'AC/DC', 'Back in Black', 'NSYNC', 'Thriller'}
```

If we add the same element twice, nothing will happen as there can be no duplicates in a set:

In [6]:
```
# Try to add duplicate element to the set

A.add("NSYNC")
A
```

Out[6]:
```
{'AC/DC', 'Back in Black', 'NSYNC', 'Thriller'}
```

We can remove an item from a set using the `remove` method:

In [7]:
```
# Remove the element from set

A.remove("NSYNC")
A
```

Out[7]:
```
{'AC/DC', 'Back in Black', 'Thriller'}
```

We can verify if an element is in the set using the `in` command:

In [8]:
```
# Verify if the element is in the set

"AC/DC" in A
```

Out[8]:
```
True
```

Sets Logic Operations

Remember that with sets you can check the difference between sets, as well as the symmetric difference, intersection, and union:

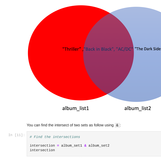
Consider the following two sets:

In [9]:
```
# Sample Sets

album_set1 = set(["Thriller", 'AC/DC', 'Back in Black'])
album_set2 = set([ "AC/DC", "Back in Black", "The Dark Side of the Moon"])
```

In [10]:
```
# Print two sets

album_set1, album_set2
```

Out[10]:
```
({'AC/DC', 'Back in Black', 'Thriller'},
 {'AC/DC', 'Back in Black', 'The Dark Side of the Moon'})
```

As both sets contain **AC/DC** and **Back in Black** we represent these common elements with the intersection of two circles.



You can find the intersect of two sets as follow using `&`:

In [11]:
```
# Find the intersection

intersection = album_set1 & album_set2
intersection
```

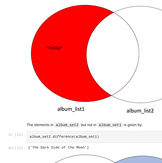
Out[11]:
```
{'AC/DC', 'Back in Black'}
```

You can find all the elements that are only contained in `album_set1` using the `difference` method:

In [12]:
```
# Find the difference in set1 but not set2

album_set1.difference(album_set2)
```

Out[12]:
```
{'Thriller'}
```

You only need to consider elements in `album_set1`; all the elements in `album_set2`, including the intersection, are not included.



The elements in `album_set2` but not in `album_set1` is given by:

In [13]:
```
album_set2.difference(album_set1)
```

Out[13]:
```
{"The Dark Side of the Moon"}
```

You can also find the intersection of `album_list1` and `album_list2` using the `intersection` method:

In [14]:
```
# Use intersection method to find the intersection of album_list1 and album_list2

album_set1.intersection(album_set2)
```

Out[14]:
```
{'AC/DC', 'Back in Black'}
```

This corresponds to the intersection of the two circles:

The union corresponds to all the elements in both sets, which is represented by coloring both circles:

The union is given by:

In [15]:
```
# Find the union of two sets

album_set1.union(album_set2)
```

Out[15]:
```
{'AC/DC', 'Back in Black', 'The Dark Side of the Moon', 'Thriller'}
```

And you can check if a set is a superset or subset of another set, respectively, like this:

In [16]:
```
# Check if superset

set(album_set1).issuperset(album_set2)
```

Out[16]:
```
False
```

In [17]:
```
# Check if subset

set(album_set2).issubset(album_set1)
```

Out[17]:
```
False
```

Here is an example where `issubset()` and `issuperset()` return true:

In [18]:
```
# Check if subset

set({"Back in Black", "AC/DC"}).issubset(album_set1)
```

Out[18]:
```
True
```

In [19]:
```
# Check if superset

album_set1.issuperset({"Back in Black", "AC/DC"})
```

Out[19]:
```
True
```

Quiz on Sets

Convert the list `['rap','house','electronic music','rap']` to a set.

In [ ]:
```
# Write your code below and press Shift+Enter to execute

set(['rap','house','electronic music','rap'])
```

▶ Click here for the solution

Consider the list `A = [1, 2, 2, 1]` and set `B = set([1, 2, 2, 1])`, does `sum(A) == sum(B)`?

In [ ]:
```
# Write your code below and press Shift+Enter to execute

A = [1, 2, 2, 1]
B = set([1, 2, 2, 1])
print("The sum of A is: ", sum(A))
print("The sum of B is: ", sum(B))
```

```
The sum of A is:  6
The sum of B is:  3
```

▶ Click here for the solution

Create a new set `album_set3` that is the union of `album_set1` and `album_set2`:

In [ ]:
```
# Write your code below and press Shift+Enter to execute

album_set1 = set(["Thriller", 'AC/DC', 'Back in Black'])
album_set2 = set([ "AC/DC", "Back in Black", "The Dark Side of the Moon"])
album_set3 = album_set1.union(album_set2)
album_set3
```

Out[20]:
```
{'AC/DC', 'Back in Black', 'The Dark Side of the Moon', 'Thriller'}
```

▶ Click here for the solution

Find out if `album_set1` is a subset of `album_set3`

In [ ]:
```
# Write your code below and press Shift+Enter to execute

album_set1.issubset(album_set3)
```

```
True
```

▶ Click here for the solution

The last exercise!

Congratulations, you have completed your first lesson and hands-on lab in Python.

Author

Joseph Santarcangelo

Other contributors

Mavis Zhou

Change Log

Date (YYYY-MM-DD)	Version	Changed By	Change Description
2020-06-18	2.1	Malika	Removed the markdow for GitShare
2020-06-26	2.0	Lavanya	Moved lab to course repo in GitLab