IBM Developer SKILLS NETWORK

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

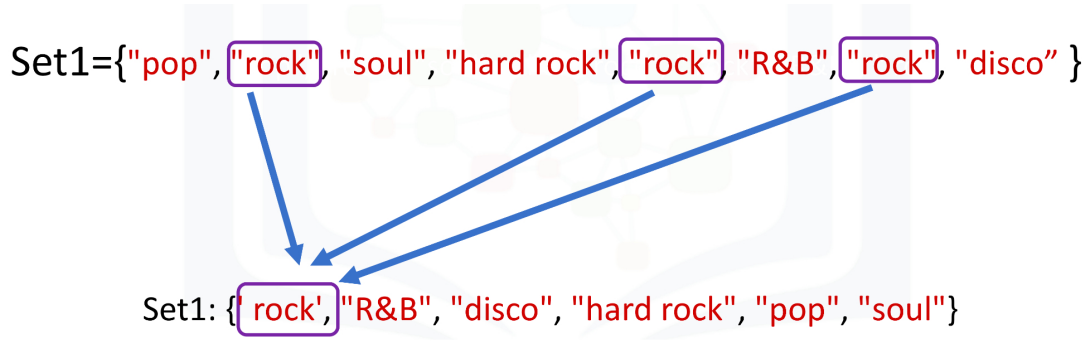
Sets

Set Content

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

[ ]:
```
# Create a set

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

The process of mapping is illustrated in the figure:



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

[ ]:
```
# 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
```

Now let us create a set of genres:

[ ]:
```
# Convert list to set

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

Set Operations

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

[ ]:
```
# Sample set

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

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

[ ]:
```
# Add element to set

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

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

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

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

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

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

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

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

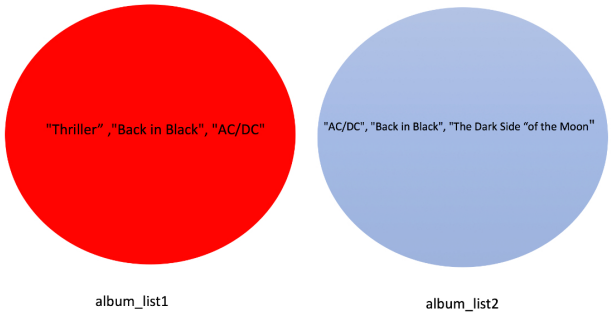
[ ]:
```
# Verify if the element is in the set

"AC/DC" in A
```

Sets Logic Operations

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

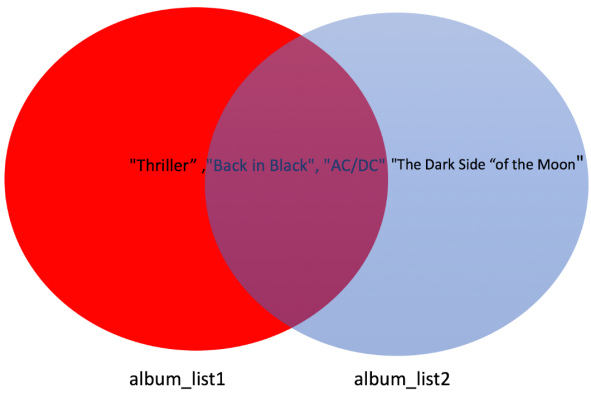
Consider the following two sets:

[ ]:
```
# 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"])
```



[ ]:
```
# Print two sets

album_set1, album_set2
```

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



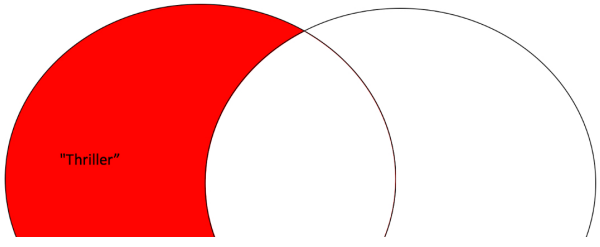
You can find the intersect of two sets as follow using `&`:

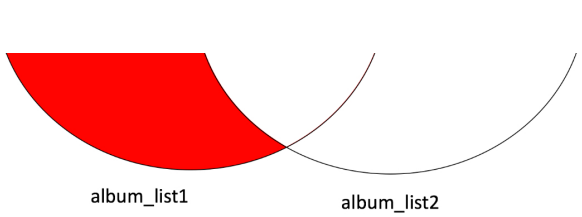
[ ]:
```
# Find the intersections

intersection = album_set1 & album_set2
intersection
```

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

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

album_set1.difference(album_set2)
```

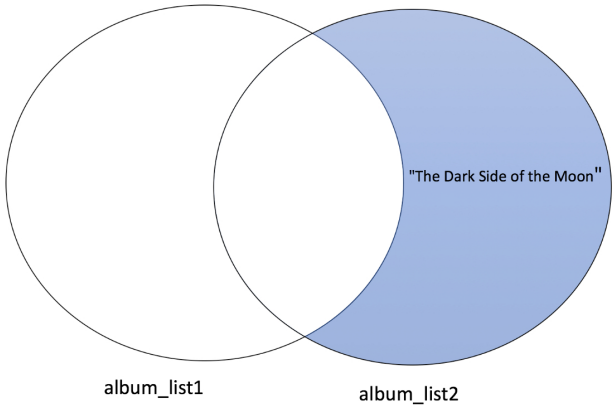
You only need to consider elements in `album_set1`; all the elements in `album_set2`, including the intersection, are not included.





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

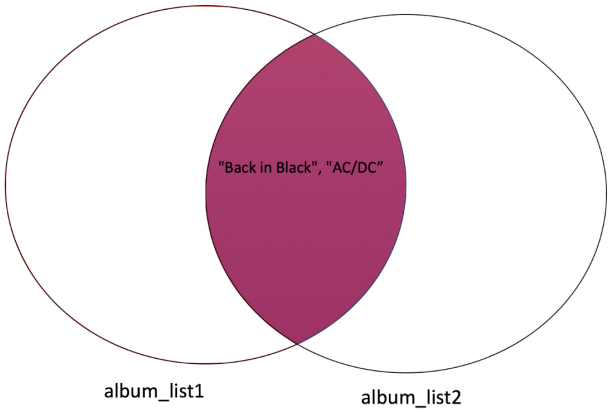
[ ]:
```
album_set2.difference(album_set1)
```



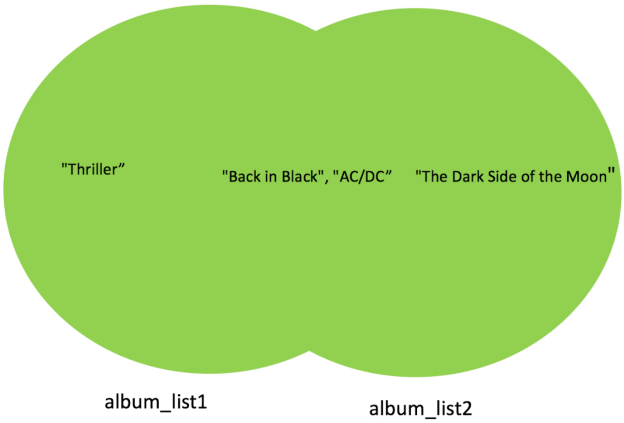
You can also find the intersection of `album_list1` and `album_list2`, using the `intersection` method:

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

album_set1.intersection(album_set2)
```

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:

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

album_set1.union(album_set2)
```

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

[ ]:
```
# Check if superset

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

[ ]:
```
# Check if subset

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

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

[ ]:
```
# Check if subset

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

[ ]:
```
# Check if superset

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

Did you know? IBM Watson Studio lets you build and deploy an AI solution, using the best of open source and IBM software and giving your team a single environment to work in. Learn more here.

Quiz on Sets

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

[3]:
```
music_set = set(['rap','house','electronic music', 'rap'])
```

```
Double-click **here** for the solution.

<!-- Your answer is below:
set(['rap','house','electronic music','rap'])
-->
```

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

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

```
Double-click **here** for the solution.

<!-- Your answer is below:
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))
-->
```

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

[ ]:
```
# 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"])
```

```
Double-click **here** for the solution.

<!-- Your answer is below:
album_set3 = album_set1.union(album_set2)
album_set3
-->
```

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

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

```
Double-click **here** for the solution.

<!-- Your answer is below:
album_set1.issubset(album_set3)
-->
```

The last exercise!

Congratulations, you have completed your first lesson and hands-on lab in Python. However, there is one more thing you need to do. The Data Science community encourages sharing work. The best way to share and showcase your work is to share it on GitHub. By sharing your notebook on GitHub you are not only building your reputation with fellow data scientists, but you can also show it off when applying for a job. Even though this was your first piece of work, it is never too early to start building good habits. So, please read and follow this article to learn how to share your work.

Author

Joseph Santarcangelo

Other contributors

Mavis Zhou

Change Log

Date (YYYY-MM-DD)	Version	Changed By	Change Description

2020-08-26	2.0	Lavanya	Moved lab to course repo in GitLab