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.

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:

```
# Create a set

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

{'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:

```
# 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

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

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
```

```
{'R&B',
 'disco',
 'folk rock',
 'hard rock',
 'pop',
 'progressive rock',
 'rock',
 'soft rock',
 'soul'}
```

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
```

```
{'AC/DC', 'Back in Black', 'Thriller'}
```

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

```
# Add element to set

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

```
{'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:

```
# Try to add duplicate element to the set

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

```
{'AC/DC', 'Back in Black', 'NSYNC', 'Thriller'}
```

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

```
# Remove the element from set
```

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

{'AC/DC', 'Back in Black', 'Thriller'}
```

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

True
```

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

({'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 &:

```
# Find the intersections

intersection = album_set1 & album_set2
intersection

{'AC/DC', 'Back in Black'}
```

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)

{'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:

```
album_set2.difference(album_set1)

{'The Dark Side of the Moon'}
```

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)

{'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:

```
# Find the union of two sets

album_set1.union(album_set2)

{'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:

```
# Check if superset

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

```
False
```

```python
# Check if subset

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

```
False
```

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

```python
# Check if subset

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

```
True
```

```python
# Check if superset

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

```
True
```

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

```python
# Write your code below and press Shift+Enter to execute
set(['rap','house','electronic music','rap'])
```

```
{'electronic music', 'house', 'rap'}
```

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

```python
# 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
```

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

```python
# 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
```

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

Find out if album_set1 is a subset of album_set3:

```
# Write your code below and press Shift+Enter to execute
album_set1.issubset(album_set3)

----------------------------------------------------------------------
-----
NameError                                 Traceback (most recent call
last)
Cell In[23], line 2
      1 # Write your code below and press Shift+Enter to execute
----> 2 album_set1.issubset(album_set3)

NameError: name 'album_set3' is not defined
```

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
2022-01-10	2.1	Malika	Removed the readme for GitShare
2020-08-26	2.0	Lavanya	Moved lab to course repo in GitLab