Lesson 10

Asg 10.3

Sets

Make sure to run the code in the following cell before you start the assignment!!</i>

In [31]:

```
# set up notebook to display multiple output in one cell

from IPython.core.interactiveshell import InteractiveShell
InteractiveShell.ast_node_interactivity = "all"

print("The notebook is now set up to display multiple output in one cell'")
```

```
The notebook is now set up to display multiple output in one cell'
```

Problem 1: Given:

Important Python Libraries for Data Science:

Pandas, NumPy, SciPy, Scikit-Learn, Matplotlib, Plotly, Seaborm. Statsmodels

a. Pass a list to the set() function to intialize a set called **libraries_set1** that contains the **Important Python Libraries for Data Science** as its values.

b. Print out the the set that you initialized in Part (a) and use the type() function to check that it is a set.

c. Pass a tuple to the set() function to intialize a set called **libraries_set2** that contains the **Important Python Libraries for Data Science** as its values.

d. Print out the the set that you initialized in Part (c) and use the type() function to check that it is a set.

e. Use curly braces { } to intialize a set called **libraries_set3** that contains the **Important Python Libraries for Data Science** as its values.

f. Print out the the set that you initialized in Part (e) and use the type() function to check that it is a set.

In [4]:

```
libraries_list = ["Pandas", "NumPy", "SciPy", "Scikit-Learn", "Matplotlib", "Plotly", "
libraries_set1 = set(libraries_list)
print(libraries_set1, type(libraries_set1))

libraries_tuple = ("Pandas", "NumPy", "SciPy", "Scikit-Learn", "Matplotlib", "Plotly", '

libraries_set2 = set(libraries_tuple)
print(libraries_set2, type(libraries_set2))
```

```
libraries_set3 = {"Pandas", "NumPy", "SciPy", "Scikit-Learn", "Matplotlib", "Plotly", "S
print(libraries_set3, type(libraries_set3))
```

```
{'SciPy', 'NumPy', 'Scikit-Learn', 'Statsmodels', 'Pandas', 'Seaborm', 'Plotly', 'Matpl
otlib'} <class 'set'>
{'SciPy', 'NumPy', 'Scikit-Learn', 'Statsmodels', 'Pandas', 'Seaborm', 'Plotly', 'Matpl
otlib'} <class 'set'>
{'Scikit-Learn', 'SciPy', 'NumPy', 'Statsmodels', 'Pandas', 'Seaborm', 'Plotly', 'Matpl
otlib'} <class 'set'>
```

Problem 2: a. Initialize a set called **greater_metro_set** that includes the values Menomonee Falls, Germantown, Marquette, West Allis Hale, Sussex Hamilton, Brookfield Central, and Wauwatosa West.

b. Print out the set that you initialized in Part(a).

c. Write code to add the values Brookfield East and Wauwatosa East to the set that you initialized in Part(a).

d. Print out the set that resulted from doing Part (c).

In [16]:

```
greater_metro_set = {"Menomonee Falls", "Germantown", "Marquette", "West Allis Hale", "S
print(greater_metro_set)
greater_metro_set.add("Brookfield East")
greater_metro_set.add("Wauwatosa East")
print(greater_metro_set)
```

```
{'Sussex Hamilton', 'Marquette', 'West Allis Hale', 'Wauwatosa West', 'Brookfield Centr
al', 'Germantown', 'Menomonee Falls'}
{'Sussex Hamilton', 'Brookfield East', 'Germantown', 'Wauwatosa West', 'Wauwatosa Eas
t', 'West Allis Hale', 'Marquette', 'Menomonee Falls', 'Brookfield Central'}
```

Problem 3: a. Initialize a set called **states** that includes the values Texas, Florida, California, Chicago, Maryland, San Francisco, Oregon, and, Vermont.

b. Print out the set that you initialized in Part(a).

c. Use both the remove() method and the discard() method to remove Chicago and San Francisco from the set that you initialized in Part(a).

d. Print out the set that resulted from doing Part (c).

e. Write code to remove all of the values from the set that you printed in Part (d).

In [25]:

```
states = {"Texas", "Florida", "California", "Chicago","Maryland","San Francisco", "Orego
print(states)
states.remove('Chicago')
states.discard('San Francisco')
print(states)
states.clear()
print(states)
```

```
{'San Francisco', 'Florida', 'Oregon', 'Texas', 'Chicago', 'California', 'Maryland', 'V
ermont'}
{'Florida', 'Oregon', 'Texas', 'California', 'Maryland', 'Vermont'}
set()
```

Problem 4: a. Initialize a set called **big_cities** that includes the values New York, Chicago, Los Angeles, Boston, Houston, and Philadelphia.

b. Print out the set that you initialized in Part(a).

c. Write and print out code to put the values of the set from Parts a & b in an ordered form.

d. Use the type() function to check the type of your result from Part (c)

In [26]:

```python
big_cities = {"New York", "Chicago", "Los Angeles", "Boston", "Houston", "Philadelphia"
print(big_cities)
print(sorted(big_cities))
print(type(big_cities))
```

```
{'Chicago', 'New York', 'Los Angeles', 'Philadelphia', 'Boston', 'Houston'}
['Boston', 'Chicago', 'Houston', 'Los Angeles', 'New York', 'Philadelphia']
<class 'set'>
```

Problem 5:

a. Create a list called **foods_list** that includes the values pizza, cake, apples, cake, chicken, apples, lobster, pizza, and cake.

b. Print out the list that you created in Part (a).

c. Write code that uses sets to remove all of the duplicates that are found in the list that you created in Part (a).

d. Print out the set that resulted from doing Part (c).

e. Convert the set from Part (d) into a list and print out the result.

In [28]:

```python
foods_list = ["pizza", "cake", "apples", "cake", "chicken", "apples", "lobster", "pizza
print(foods_list)
foods_set = set(foods_list)
print(foods_set)
print(list(foods_set))
```

```
['pizza', 'cake', 'apples', 'cake', 'chicken', 'apples', 'lobster', 'pizza', 'cake']
{'apples', 'pizza', 'cake', 'chicken', 'lobster'}
['apples', 'pizza', 'cake', 'chicken', 'lobster']
```

Problem 6:

a. Initialize the two sets **subjects_set1** and **subjects_set2** which have the values indicated below. - subjects_set1: Calculus, Chemistry, Spanish, APUSH, PE - subjects_set2: Calculus, Biology, French, APUSH, Health b. Print out the two sets that you initialized in Part (a).

c. Print out the union of the two sets that you initialized in Part (a).

d. Print out the intersection of the two sets that you initialized in Part (a).

e. Print out the difference of the two sets that you initialized in Part (a).

f. Print out the symmetric difference the two sets that you initialized in Part (a).

In [30]:

```python
subjects_set1 = {"Calculus", "Chemistry", "Spanish", "APUSH", "PE"}
subjects_set2 = {"Calculus", "Biology", "French", "APUSH", "Health"}
print(subjects_set1, subjects_set2)
print(subjects_set1.union(subjects_set2))
print(subjects_set1.intersection(subjects_set2))
print(subjects_set1.difference(subjects_set2))
print(subjects_set1.symmetric_difference(subjects_set2))
```

```
{'PE', 'Chemistry', 'APUSH', 'Spanish', 'Calculus'} {'French', 'APUSH', 'Calculus', 'Bi
ology', 'Health'}
{'PE', 'Chemistry', 'French', 'Calculus', 'Health', 'APUSH', 'Spanish', 'Biology'}
{'Calculus', 'APUSH'}
{'PE', 'Chemistry', 'Spanish'}
{'PE', 'French', 'Chemistry', 'Spanish', 'Biology', 'Health'}
```

```
In [ ]:
```