OBJECTIVE:

Execution of basic python code

Problem Defintion and Approach:

List Comprehension - To print all the letters in the Python

Dictionary - A simple student details

loop and Conditional statements - to print event and odd numbers in a given range

In [36]:
```
list = [ i for i in 'PYTHON']
print(list)
```

```
['P', 'Y', 'T', 'H', 'O', 'N']
```

In [41]:
```
from pprint import pprint
```

In [47]:
```
Student = {'name':'Kalki', 'class':'2MsDs', 'University':'Christ'}
pprint("{name} of class {class} from {University} University".format(**Student))
```

```
'Kalki of class 2MsDs from Christ University'
```

In [54]:
```
even=[]
odd=[]
for i in range(1,20):
    if i%2==0:
        even.append(i)
    else:
        odd.append(i)
print(even)
print(odd)
```

```
[2, 4, 6, 8, 10, 12, 14, 16, 18]
[1, 3, 5, 7, 9, 11, 13, 15, 17, 19]
```

OBJECTIVE:

Executing a few basic codes in the following libraries:

- Pandas
- NumPy
- Scikit-Learn
- Seaborn
- Theano

- SciPy
- PyTorch
- Keras
- TensorFlow

Import Iris Toy Dataset from Sklearn

Problem Definition:

To Execute codes in the above mentioned library, we have to initialize it in the notebook first. If its already installed, we can initialize directly. Else install the neccessary libraries.

Approach:

We are going to use "import" to initialize the libraries. If the library doesn't exist, we can either install it in Anaconda Navigator or use the command 'pip install (package name)' in the jupyter notebook.

We are checking which libraries already exists in our jupyter.

In [65]:
```
help("modules")
```

```
Please wait a moment while I gather a list of all available modules...

Cython              brain_sqlalchemy    mailbox             sndhdr
IPython             brain_ssl           mailcap             sniffio
OpenSSL             brain_subprocess    mako                snowballstemmer
PIL                 brain_threading     markdown            socket
PyQt5               brain_type          markupsafe          socketserver
__future__          brain_typing        marshal             socks
_abc                brain_uuid          math                sockshandler
_ast                brotli              matplotlib          sortedcollections
_asyncio            bs4                 mccabe              sortedcontainers
_bisect             builtins            menuinst            soupsieve
_black_version      bz2                 mimetypes           sphinx
_blake2             cProfile            mistune             sphinxcontrib
_bootlocale         cachetools          mkl                 sphinxify
_bz2                caffe2              mkl_fft             sphinxthread
_cffi_backend       calendar            mkl_random          spyder
_codecs             certifi             mmap                spyder_kernels
```

In [66]:
```
pip install torch
```

```
Requirement already satisfied: torch in c:\users\sridhar\anaconda3\lib\site-pac
kages (1.10.2)
Requirement already satisfied: typing-extensions in c:\users\sridhar\anaconda3
\lib\site-packages (from torch) (3.7.4.3)
Note: you may need to restart the kernel to use updated packages.
```

In [71]:
```
import pandas as pd
import numpy as np
import sklearn as sk
import seaborn as sns
import theano
import scipy as sp
import torch
import keras as k
import tensorflow as tf
```

Pandas:

Pandas is a python package which provides very fast and flexible data structures designed to make working with labeled data both easy and intuitive. It aims to be the fundamental high-level building block for doing practical, real world data analysis in python. The two main data structures of pandas are DataFrame (2-Dimensional) and Series (1-Dimensional array). Pandas is built on top of NumPy.

In [57]:
```
poke = pd.read_csv('pokemon.csv')
print(poke.head(5))
```

```
   #                    Name Type 1  Type 2  Total  HP  Attack  Defense  \
0  1               Bulbasaur  Grass  Poison    318  45      49       49
1  2                 Ivysaur  Grass  Poison    405  60      62       63
2  3                Venusaur  Grass  Poison    525  80      82       83
3  3  VenusaurMega Venusaur  Grass  Poison    625  80     100      123
4  4              Charmander   Fire     NaN    309  39      52       43

   Sp. Atk  Sp. Def  Speed  Generation  Legendary
0       65       65     45           1      False
1       80       80     60           1      False
2      100      100     80           1      False
3      122      120     80           1      False
4       60       50     65           1      False
```

In [58]:
```
print(poke.iloc[2:5])
```

```
   #                    Name Type 1  Type 2  Total  HP  Attack  Defense  \
2  3                Venusaur  Grass  Poison    525  80      82       83
3  3  VenusaurMega Venusaur  Grass  Poison    625  80     100      123
4  4              Charmander   Fire     NaN    309  39      52       43

   Sp. Atk  Sp. Def  Speed  Generation  Legendary
2      100      100     80           1      False
3      122      120     80           1      False
4       60       50     65           1      False
```

In [59]:
```
pd.Series({2:'a',1:'b',3:'c'}, index=[1,2,3])
```

Out[59]:
```
1    b
2    a
3    c
dtype: object
```

NumPy:

Numpy Stands for Numerical Python. It is a python package used for working with multidimensional arrays. It is used to perform mathematical and logical operations on arrays. It often works as an replacement for MatLab, when used along with packages like SciPy and MatplotLib. The major Advantage is that it is Open Source. The most important object defined in Numpy is an N-dimensional array type called ndarray. Each element in ndarray is an object of the data-type object called dtype.

In [60]:
```
a=np.array([1,2,3,4,5.0],dtype='int')
print(a)
type(a)
```

```
[1 2 3 4 5]
```

Out[60]:
```
numpy.ndarray
```

In [61]:
```
a=np.array([1,2,3,4,5,6,7,8,9])
np.reshape(a,(3,3))
```

Out[61]:
```
array([[1, 2, 3],
       [4, 5, 6],
       [7, 8, 9]])
```

Scikit-Learn:

Scikit-Learn also known as Sklearn is the most useful and robust library for machine learning in python. It is a simple and efficient tool for predictive analysis. It is accessible to everyone and reusable in various contexts. This library is largely written in Python, is built upon NumPy, SciPy and Matplotlib. To work with Sklearn, one has to have a basic understanding of Python, NumPy, Scipy and MatplotLib.

To import a dataset in sklearn, we have to import the module datasets.

In [2]:
```
from sklearn import datasets
iris=sk.datasets.load_iris()
```

In [3]:
```
iris.DESCR
```

Out[3]:
```
'.. _iris_dataset:\n\nIris plants dataset\n--------------------\n\n**Data Set
Characteristics:**\n\n    :Number of Instances: 150 (50 in each of three clas
ses)\n    :Number of Attributes: 4 numeric, predictive attributes and the cla
ss\n    :Attribute Information:\n        - sepal length in cm\n        - sepa
l width in cm\n        - petal length in cm\n        - petal width in cm\n
- class:\n                - Iris-Setosa\n                - Iris-Versicolour\n
- Iris-Virginica\n                \n    :Summary Statistics:\n\n    =========
===== ==== ==== ======= ===== ====================\n                    Min
Max   Mean    SD   Class Correlation\n    ============== ==== ==== ======= ==
=== ====================\n    sepal length:   4.3  7.9   5.84   0.83    0.782
6\n    sepal width:    2.0  4.4   3.05   0.43   -0.4194\n    petal length:
1.0  6.9   3.76   1.76    0.9490  (high!)\n    petal width:    0.1  2.5   1.2
0   0.76    0.9565  (high!)\n    ============== ==== ==== ======= ===== =====
==============\n\n    :Missing Attribute Values: None\n    :Class Distributi
on: 33.3% for each of 3 classes.\n    :Creator: R.A. Fisher\n    :Donor: Mich
ael Marshall (MARSHALL%PLU@io.arc.nasa.gov)\n    :Date: July, 1988\n\nThe fam
ous Iris database, first used by Sir R.A. Fisher. The dataset is taken\nfrom
Fisher\'s paper. Note that it\'s the same as in R, but not as in the UCI\nMac
hine Learning Repository, which has two wrong data points.\n\nThis is perhaps
```

In [4]:
```
iris.data
```

```
       [6. , 3.4, 4.5, 1.6],
       [6.7, 3.1, 4.7, 1.5],
       [6.3, 2.3, 4.4, 1.3],
       [5.6, 3. , 4.1, 1.3],

       [5.5, 2.5, 4. , 1.3],
       [5.5, 2.6, 4.4, 1.2],
       [6.1, 3. , 4.6, 1.4],
       [5.8, 2.6, 4. , 1.2],
       [5. , 2.3, 3.3, 1. ],
       [5.6, 2.7, 4.2, 1.3],
       [5.7, 3. , 4.2, 1.2],
       [5.7, 2.9, 4.2, 1.3],
       [6.2, 2.9, 4.3, 1.3],
       [5.1, 2.5, 3. , 1.1],
       [5.7, 2.8, 4.1, 1.3],
       [6.3, 3.3, 6. , 2.5],
       [5.8, 2.7, 5.1, 1.9],
       [7.1, 3. , 5.9, 2.1],
       [6.3, 2.9, 5.6, 1.8],
```

In [5]:
```
iris.feature_names
```

Out[5]:
```
['sepal length (cm)',
 'sepal width (cm)',
 'petal length (cm)',
 'petal width (cm)']
```

In [6]:
```
iris.target
```

Out[6]:
```
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2])
```

In [7]:
```
iris.data.shape
```

Out[7]:
```
(150, 4)
```

Seaborn:

Seaborn is a data visualization library for statistical graphics plotting, which is built on top of matplotlib and closely integrated with pandas data structures. Visualization is the central part of seaborn which helps in exploration and undertstanding of data. One has to be familiar with NumPy, MatplotLib and Pandas to learn about Seaborn. It Provides dataset based APIs, so that we can switch between different Visual representations for some variables for better understanding of datset.

In [14]:
```
sns.get_dataset_names()
```

Out[14]:
```
['anagrams',
 'anscombe',
 'attention',
 'brain_networks',
 'car_crashes',
 'diamonds',
 'dots',
 'exercise',
 'flights',
 'fmri',
 'gammas',
 'geyser',
 'iris',
 'mpg',
 'penguins',
 'planets',
 'taxis',
 'tips',
 'titanic']
```

In [15]:
```
data = sns.load_dataset('planets')
data.shape
```

Out[15]:
```
(1035, 6)
```

Theano:

Theano is a numerical computation library for python. It is a common choice for implementing neural network models as it allows you to efficiently define, optimize and evaluate mathematical expressions, include multi-dimensonal arrays. Theano makes it possible to attain high speeds that give a tough competition to hand-crafted C implementations for problems involving Large amounts of data. It has got an amazing compiler which can do various optimizations of varying complexity. Typically it manipulates matrices using numpy package, so it makes it better than any such package.

In [62]:
```
from theano import tensor
```

In [63]:
```
x = tensor.dscalar()
y = tensor.dscalar()
z = x-y
f = theano.function([x,y],z)
print(f(2.5,1.5))
```

```
1.0
```

SciPy:

SciPy stands for Scientific Python, which is a scientific Computation Library that uses Numpy underneath. It provides more utility functions for optimization, stats and signal processing. It is an open source library. Itallows users to manipulate data and visualize the data using a wide range of high-level python commands.

In [16]:
```
from scipy import constants
print(constants.liter)
```

```
0.001
```

In [19]:
```
print(sp.__version__)
```

```
1.6.2
```

In [20]:
```
print(constants.pi)
```

```
3.141592653589793
```

Keras:

Keras is a python based open source library used for training models in deep learning. It provides an interface for artificial neural networks. It runs on top of Machine Learning platform Tensorflow. It was developed with a focus on enabling fast experimentation.

In the below code samples we are exploring a few among various popular datasets which are already incorporated in the keras.datasets module.

In [68]:
```
from keras.datasets import mnist
(x_train, y_train), (x_test, y_test) = mnist.load_data()
```

In [77]:
```
from tensorflow.keras.models import Sequential
model = Sequential
```

Tensorflow:

Tensorflow is the second machine learning framework that google created and used to design, build and train deep learning models. Tensorflow library is used to do numerical computations. The name 'TensorFlow' is derived from the operations which neural networks perform on multidimensional data arrays or tensors. It's literaly a flow of tensors.

In [72]:
```
x1 = tf.constant([1,2,3,4])
x2 = tf.constant([5,6,7,8])
result = tf.multiply(x1,x2)
print(result)
```

```
tf.Tensor([ 5 12 21 32], shape=(4,), dtype=int32)
```

Sections:

Lab Overview

Executing Basic Python Code

Libraries

Installation Instructions

Installing necessary packages

Importing Libraries

Numpy

Pandas

SKLearn

Loading IRIS TOY Dataset

Seaborn

SciPy

PyTorch

Keras

Tensorflow

Conclusion

References:

https://www.journaldev.com/17840/theano-python-tutorial
(https://www.journaldev.com/17840/theano-python-tutorial)
https://www.datacamp.com/community/tutorials/tensorflow-tutorial
(https://www.datacamp.com/community/tutorials/tensorflow-tutorial)
https://www.mygreatlearning.com/blog/python-numpy-tutorial/
(https://www.mygreatlearning.com/blog/python-numpy-tutorial/)
https://stackoverflow.com/questions/57735701/cant-import-torch-in-jupyter-notebook
(https://stackoverflow.com/questions/57735701/cant-import-torch-in-jupyter-notebook)
https://pandas.pydata.org/docs/getting_started/overview.html#:~:text=pandas%20is%20a%20Pythor
(https://pandas.pydata.org/docs/getting_started/overview.html#:~:text=pandas%20is%20a%20Pytho
https://www.geeksforgeeks.org/install-python-package-using-jupyter-notebook/
(https://www.geeksforgeeks.org/install-python-package-using-jupyter-notebook/)
https://www.w3schools.com/python/scipy/scipy_intro.php
(https://www.w3schools.com/python/scipy/scipy_intro.php)
https://www.journaldev.com/18341/python-scikit-learn-tutorial
(https://www.journaldev.com/18341/python-scikit-learn-tutorial)

CONCLUSION:

We have executed a few basic codes in python.

We have Installed, imported and worked on different libraries that exists in python that are used in machine and deep learning and various aspects of Data Science.

Downloaded IRIS dataset from sklearn to perfom a few basic coding.

In [ ]: