**What is NumPy?**

NumPy stands for ‘Numerical Python’ or ‘Numeric Python’.

**import numpy as np**

**a = np.array([1, 4, 5, 8], float)**

**a**

**type(a)**

**a[:2]**

**a[3]**

**a[0] = 5**

**a**

**a = np.array([[1, 2, 3], [4, 5, 6]], float)**

**a[0,0]**

**a[0,1]**

a[1,:] //row

**a[:,2] //column**

**a.shape**

a.dtype //8-byte) real numbers

**len(a) // number of rows**

**2 in a //true or false**

**0 in a //true or false**

a = np.array(range(10), float)

a = a.reshape((5, 2))

**a.shape**

**a = np.array([1, 2, 3], float)**

**>>> b = a**

**>>> c = a.copy()**

**>>> a[0] = 0**

**>>> a**

**a = np.array(range(6), float).reshape((2, 3))**

**a.transpose()**

**a.flatten()**

**>>> a = np.array([1,2], float)**

**>>> b = np.array([3,4,5,6], float)**

**>>> c = np.array([7,8,9], float)**

**>>> np.concatenate((a, b, c))**

**array([1., 2., 3., 4., 5., 6., 7., 8., 9.])**

**a = np.array([[1, 2], [3, 4]], float)**

**b = np.array([[5, 6], [7,8]], float)**

**np.concatenate((a,b))**

**np.concatenate((a,b), axis=0)**

**np.concatenate((a,b), axis=1)**

**a = np.array([1, 2, 3], float)**

**a[:,np.newaxis].shape**

**a[:,np.newaxis].shape**

**b[np.newaxis,:]**

**b[np.newaxis,:].shape**

**Other ways to create arrays**

**np.arange(5, dtype=float)**

np.arange(1, 6, 2, dtype=int)

**np.ones((2,3), dtype=float)**

**np.zeros(7, dtype=int)**

np.zeros\_like(a)

**np.ones\_like(a)**

**np.identity(4, dtype=float)**

**np.eye(4, k=1, dtype=float)**

**Array mathematics**

**a = np.array([1,2,3], float)**

**b = np.array([5,2,6], float)**

**a + b**

**a – b**

**a \* b**

**b / a**

**a % b**

a = np.array([[1, 2], [3, 4], [5, 6]], float)

b = np.array([-1, 3], float)

a + b //broad casting

**a = np.zeros((2,2), float)**

abs, sign, sqrt, log, log10, exp, sin, cos, tan, arcsin, arccos,

arctan, sinh, cosh, tanh, arcsinh, arccosh, and arctanh.

a = np.zeros((2,2), float)

**b = np.array([-1., 3.], float)**

**a + b**

**a + b[np.newaxis,:]**

**a + b[:,np.newaxis]**

**a = np.array([1, 4, 9], float)**

**np.sqrt(a)**

**a = np.array([1.1, 1.5, 1.9], float)**

**np.floor(a)**

**np.ceil(a)**

**np.rint(a)**

**np.pi**

**np.e**

**Array iteration**

**a = np.array([1, 4, 5], int)**

**for x in a:**

**print (x);**

a = np.array([[1, 2], [3, 4], [5, 6]], float)

for x in a:

print (x);

a = np.array([[1, 2], [3, 4], [5, 6]], float)

for (x, y) in a:

print (x \* y);

**Basic array operations**

**a = np.array([2, 4, 3], float)**

**a.sum()**

**a.prod()**

**np.sum(a)**

**np.prod(a)**

**a = np.array([2, 1, 9], float)**

**a.mean()**

**a.var()**

**a.std()**

**a.min()**

**a.max()**

**a.argmin()**

**a.argmax()**

**a = np.array([[0, 2], [3, -1], [3, 5]], float)**

**a.mean(axis=0)**

**a.mean(axis=1)**

**a.min(axis=1)**

**a.max(axis=0)**

**a = np.array([6, 2, 5, -1, 0], float)**

**sorted(a)**

**a.sort()**

**a**

**a = np.array([6, 2, 5, -1, 0], float)**

**a.clip(0, 5) //** min(max(x, *minval*), *maxval*)

**a = np.array([1, 1, 4, 5, 5, 5, 7], float)**

**np.unique(a)**

**a = np.array([[1, 2], [3, 4]], float)**

**a.diagonal()**

**Comparison operators and value testing**

**a = np.array([1, 3, 0], float)**

**b = np.array([0, 3, 2], float)**

**a > b**

**>>> a == b**

**>>> a <= b**

**c = a > b**

**a = np.array([1, 3, 0], float)**

**a > 2**

**c = np.array([ True, False, False], bool)**

**any(c)**

**all(c)**

**a = np.array([1, 3, 0], float)**

**np.logical\_and(a > 0, a < 3)**

**b = np.array([True, False, True], bool)**

**np.logical\_not(b)**

**c = np.array([False, True, False], bool)**

**np.logical\_or(b, c)**

**Vector and matrix mathematics**

**a = np.array([1, 2, 3], float)**

**b = np.array([0, 1, 1], float)**

**np.dot(a, b)**

**a = np.array([[0, 1], [2, 3]], float)**

**b = np.array([2, 3], float)**

**c = np.array([[1, 1], [4, 0]], float)**

**a**

**np.dot(b, a)**

**np.dot(a, b)**

**np.dot(a, c)**

**np.dot(c, a)**

**np.random.rand(5)**

np.random.rand(2,3)

**np.random.random()**

**np.random.randint(5, 10)**

## What is Matplotlib?

Matplotlib is a low level graph plotting library in python that serves as a visualization utility.

Matplotlib was created by John D. Hunter.

Matplotlib is open source and we can use it freely.

Matplotlib is mostly written in python, a few segments are written in C, Objective-C and Javascript for Platform compatibility.

import matplotlib

print(matplotlib.\_\_version\_\_)

import matplotlib.pyplot as plt  
import numpy as np  
xpoints = np.array([0, 6])  
ypoints = np.array([0, 250])  
  
plt.plot(xpoints, ypoints)  
plt.show()

## Plotting x and y points

The plot() function is used to draw points (markers) in a diagram.

By default, the plot() function draws a line from point to point.

The function takes parameters for specifying points in the diagram.

Parameter 1 is an array containing the points on the **x-axis**.

Parameter 2 is an array containing the points on the **y-axis**.

If we need to plot a line from (1, 3) to (8, 10), we have to pass two arrays [1, 8] and [3, 10] to the plot function.

import matplotlib.pyplot as plt  
import numpy as np  
  
xpoints = np.array([1, 8])  
ypoints = np.array([3, 10])  
  
plt.plot(xpoints, ypoints)  
plt.show()

import matplotlib.pyplot as plt  
import numpy as np  
  
xpoints = np.array([1, 8])  
ypoints = np.array([3, 10])  
  
plt.plot(xpoints, ypoints, 'o')  
plt.show()

import matplotlib.pyplot as plt  
import numpy as np  
xpoints = np.array([1, 2, 6, 8])  
ypoints = np.array([3, 8, 1, 10])  
plt.plot(xpoints, ypoints)  
plt.show()

import matplotlib.pyplot as plt  
import numpy as np  
ypoints = np.array([3, 8, 1, 10, 5, 7])  
plt.plot(ypoints)  
plt.show()

import matplotlib.pyplot as plt  
import numpy as np  
ypoints = np.array([3, 8, 1, 10])  
plt.plot(ypoints, marker = '\*')  
plt.show()

import matplotlib.pyplot as plt  
import numpy as np  
ypoints = np.array([3, 8, 1, 10])  
plt.plot(ypoints, 'o:r')  
plt.show()

import matplotlib.pyplot as plt  
import numpy as np  
ypoints = np.array([3, 8, 1, 10])  
plt.plot(ypoints, marker = 'o', ms = 20)  
plt.show()

import matplotlib.pyplot as plt  
import numpy as np  
ypoints = np.array([3, 8, 1, 10])  
plt.plot(ypoints, marker = 'o', ms = 20, mec = 'r')  
plt.show()

import matplotlib.pyplot as plt  
import numpy as np  
ypoints = np.array([3, 8, 1, 10])  
plt.plot(ypoints, marker = 'o', ms = 20, mfc = 'r')  
plt.show()

import matplotlib.pyplot as plt

import numpy as np

ypoints = np.array([3, 8, 1, 10])

plt.plot(ypoints, linestyle = 'dotted')

plt.show()

plt.plot(ypoints, linestyle = 'dashed')

// line color

import matplotlib.pyplot as plt  
import numpy as np  
ypoints = np.array([3, 8, 1, 10])  
plt.plot(ypoints, color = 'r')  
plt.show()

//line width

import matplotlib.pyplot as plt

import numpy as np

ypoints = np.array([3, 8, 1, 10])

plt.plot(ypoints, linewidth = '20.5')

plt.show()

//label

import numpy as np

import matplotlib.pyplot as plt

x = np.array([80, 85, 90, 95, 100, 105, 110, 115, 120, 125])

y = np.array([240, 250, 260, 270, 280, 290, 300, 310, 320, 330])

plt.plot(x, y)

plt.xlabel("Average Pulse")

plt.ylabel("Calorie Burnage")

plt.show()

//add title

plt.title("Sports Watch Data")

//font change

mport numpy as np  
import matplotlib.pyplot as plt  
x = np.array([80, 85, 90, 95, 100, 105, 110, 115, 120, 125])  
y = np.array([240, 250, 260, 270, 280, 290, 300, 310, 320, 330])  
font1 = {'family':'serif','color':'blue','size':20}  
font2 = {'family':'serif','color':'darkred','size':15}  
plt.title("Sports Watch Data", fontdict = font1)  
plt.xlabel("Average Pulse", fontdict = font2)  
plt.ylabel("Calorie Burnage", fontdict = font2)  
plt.plot(x, y)  
plt.show()

//title position

import numpy as np  
import matplotlib.pyplot as plt  
x = np.array([80, 85, 90, 95, 100, 105, 110, 115, 120, 125])  
y = np.array([240, 250, 260, 270, 280, 290, 300, 310, 320, 330])  
plt.title("Sports Watch Data", loc = 'left')  
plt.xlabel("Average Pulse")  
plt.ylabel("Calorie Burnage")  
plt.plot(x, y)  
plt.show()

//grid

import numpy as np  
import matplotlib.pyplot as plt  
x = np.array([80, 85, 90, 95, 100, 105, 110, 115, 120, 125])  
y = np.array([240, 250, 260, 270, 280, 290, 300, 310, 320, 330])  
plt.title("Sports Watch Data")  
plt.xlabel("Average Pulse")  
plt.ylabel("Calorie Burnage")  
plt.plot(x, y)  
plt.grid()  
plt.show()

**Displaying multiple Plot**

With the subplot() function you can draw multiple plots in one figure:

The subplot() function takes three arguments that describes the layout of the figure.

The layout is organized in rows and columns, which are represented by the first and second argument.

The third argument represents the index of the current plot.

plt.subplot(1, 2, 1)  
#the figure has 1 row, 2 columns, and this plot is the first plot.

import matplotlib.pyplot as plt

import numpy as np

#plot 1:

x = np.array([0, 1, 2, 3])

y = np.array([3, 8, 1, 10])

plt.subplot(1, 2, 1)

plt.plot(x,y)

#plot 2:

x = np.array([0, 1, 2, 3])

y = np.array([10, 20, 30, 40])

plt.subplot(1, 2, 2)

plt.plot(x,y)

plt.show()

import matplotlib.pyplot as plt  
import numpy as np  
#plot 1:  
x = np.array([0, 1, 2, 3])  
y = np.array([3, 8, 1, 10])  
plt.subplot(2, 1, 1)  
plt.plot(x,y)  
#plot 2:  
x = np.array([0, 1, 2, 3])  
y = np.array([10, 20, 30, 40])  
plt.subplot(2, 1, 2)  
plt.plot(x,y)  
plt.show()

## Creating Scatter Plots

With Pyplot, you can use the scatter() function to draw a scatter plot.

import matplotlib.pyplot as plt

import numpy as np

x = np.array([5,7,8,7,2,17,2,9,4,11,12,9,6])

y = np.array([99,86,87,88,111,86,103,87,94,78,77,85,86])

plt.scatter(x, y)

plt.show()

//compare two plots

import matplotlib.pyplot as plt  
import numpy as np  
  
#day one, the age and speed of 13 cars:  
x = np.array([5,7,8,7,2,17,2,9,4,11,12,9,6])  
y = np.array([99,86,87,88,111,86,103,87,94,78,77,85,86])  
plt.scatter(x, y)  
  
#day two, the age and speed of 15 cars:  
x = np.array([2,2,8,1,15,8,12,9,7,3,11,4,7,14,12])  
y = np.array([100,105,84,105,90,99,90,95,94,100,79,112,91,80,85])  
plt.scatter(x, y)  
plt.show()

// creating bars

import matplotlib.pyplot as plt  
import numpy as np  
  
x = np.array(["A", "B", "C", "D"])  
y = np.array([3, 8, 1, 10])  
plt.bar(x,y)  
plt.show()

// If you want the bars to be displayed horizontally instead of vertically, use the barh() function:

import matplotlib.pyplot as plt  
import numpy as np  
x = np.array(["A", "B", "C", "D"])  
y = np.array([3, 8, 1, 10])  
plt.barh(x, y)  
plt.show()

//bar width

import matplotlib.pyplot as plt  
import numpy as np  
x = np.array(["A", "B", "C", "D"])  
y = np.array([3, 8, 1, 10])  
plt.bar(x, y, width = 0.1)  
plt.show()

//Histogram

import matplotlib.pyplot as plt  
import numpy as np  
x = np.random.normal(170, 10, 250)  
plt.hist(x)  
plt.show()

//Pie chart

import matplotlib.pyplot as plt  
import numpy as np  
y = np.array([35, 25, 25, 15])  
plt.pie(y)  
plt.show()

import matplotlib.pyplot as plt  
import numpy as np  
y = np.array([35, 25, 25, 15])  
mylabels = ["Apples", "Bananas", "Cherries", "Dates"]  
plt.pie(y, labels = mylabels)  
plt.show()

import matplotlib.pyplot as plt  
import numpy as np  
y = np.array([35, 25, 25, 15])  
mylabels = ["Apples", "Bananas", "Cherries", "Dates"]  
  
plt.pie(y, labels = mylabels)  
plt.legend()  
plt.show()

==================

data = np.loadtxt("./ex1data1.txt", delimiter = ",")

data.shape

x=data[:,0]

y=data[:,1]

plt.scatter(x, y)

plt.show()

========================

**vectorized vs non vectorized version(almost 300 time difference)**

import time

a=np.random.rand(1000000)

b=np.random.rand(1000000)

tic=time.time()

c=np.dot(a,b)

toc=time.time()

print(c)

print("vectorized version " + str(1000\*(toc-tic))+ "ms")

c=0

tic=time.time()

for i in range(1000000):

c+=a[i]\*b[i]

toc=time.time()

print(c)

print("Non Vectorized " + str(1000\*(toc-tic))+ "ms")