In [1]:
```
import numpy as np
```

In [2]:
```
arr_1d = np.array([1,2,3,4])
print(arr_1d)
```

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

In [3]:
```
type(arr_1d)
```

Out[3]:
```
numpy.ndarray
```

In [4]:
```
arr_1d.size
```

Out[4]:
```
4
```

In [5]:
```
arr_1d.ndim
```

Out[5]:
```
1
```

In [6]:
```
arr_2d = np.array([[1,2,3,4],[5,6,7,8]])
print(arr_2d)
```

```
[[1 2 3 4]
 [5 6 7 8]]
```

In [7]:
```
arr_2d.ndim
```

Out[7]:
```
2
```

In [8]:
```
arr_2d.size
```

Out[8]:
```
8
```

In [9]:
```
# Rows, Columns
arr_2d.shape
```

Out[9]:
```
(2, 4)
```

In [10]:
```
arr_2d.dtype
```

Out[10]:
```
dtype('int32')
```

In [11]:
```
mx_1s = np.array([[1,1,1],[1,1,1],[1,1,1]])
print(mx_1s)
```

```
[[1 1 1]
 [1 1 1]
 [1 1 1]]
```

In [12]:
```
mx_1s = np.ones(5)
print(mx_1s)
```

```
[1. 1. 1. 1. 1.]
```

In [13]:
```
mx_1s.dtype
```

Out[13]:
```
dtype('float64')
```

In [14]:
```
mx_1s = np.ones((3,4))
print(mx_1s)
```

```
[[1. 1. 1. 1.]
 [1. 1. 1. 1.]
 [1. 1. 1. 1.]]
```

In [15]:
```
mx_1s = np.ones((3,4),dtype = int)
print(mx_1s)
```

```
[[1 1 1 1]
 [1 1 1 1]
 [1 1 1 1]]
```

In [16]:
```
mx_0s = np.zeros((4,6))
print(mx_0s)
```

```
[[0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0.]]
```

In [17]:
```
mx_0s = np.zeros((4,6),dtype = bool)
print(mx_0s)
```

```
[[False False False False False False]
 [False False False False False False]
 [False False False False False False]
 [False False False False False False]]
```

In [18]:
```
mx_0s = np.zeros((4,6),dtype = str)
print(mx_0s)
```

```
[['' '' '' '' '' '']
 ['' '' '' '' '' '']
 ['' '' '' '' '' '']
 ['' '' '' '' '' '']]
```

In [19]:
```
em_str = ''
print(bool(em_str))
```

```
False
```

Out[35]:
```
array([[ 2,  4,  6],
       [ 8, 10, 12],
       [14, 16, 18]])
```

In [36]:
```
arr1 - arr2
```

Out[36]:
```
array([[0, 0, 0],
       [0, 0, 0],
       [0, 0, 0]])
```

In [37]:
```
np.subtract(arr1,arr2)
```

Out[37]:
```
array([[0, 0, 0],
       [0, 0, 0],
       [0, 0, 0]])
```

In [38]:
```
arr1 / arr2
```

Out[38]:
```
array([[1., 1., 1.],
       [1., 1., 1.],
       [1., 1., 1.]])
```

In [39]:
```
np.divide(arr1,arr2)
```

Out[39]:
```
array([[1., 1., 1.],
       [1., 1., 1.],
       [1., 1., 1.]])
```

In [40]:
```
arr1 * arr2
```

Out[40]:
```
array([[ 1,  4,  9],
       [16, 25, 36],
       [49, 64, 81]])
```

In [41]:
```
np.multiply(arr1,arr2)
```

Out[41]:
```
array([[ 1,  4,  9],
       [16, 25, 36],
       [49, 64, 81]])
```

In [42]:
```
arr1 @ arr2
```

Out[42]:
```
array([[ 30,  36,  42],
       [ 66,  81,  96],
       [102, 126, 150]])
```

In [43]:
```
arr1.dot(arr2)
```

Out[43]:
```
array([[ 30,  36,  42],
       [ 66,  81,  96],
       [102, 126, 150]])
```

In [44]:
```
arr1.max()
```

Out[44]:
```
9
```

In [45]:
```
arr1.argmax()
```

Out[45]:
```
8
```

In [46]:
```
arr1.min()
```

Out[46]:
```
1
```

In [47]:
```
# 0 Represent - Columns / 1 Represent Rows
arr1.max(axis = 0)
```

Out[47]:
```
array([7, 8, 9])
```

In [48]:
```
arr1.max(axis = 1)
```

Out[48]:
```
array([3, 6, 9])
```

In [49]:
```
arr1.argmin()
```

Out[49]:
```
0
```

In [50]:
```
arr1.min(axis = 0)
```

Out[50]:
```
array([1, 2, 3])
```

In [51]:
```
arr1.min(axis = 1)
```

Out[51]:
```
array([1, 4, 7])
```

In [52]:
```
np.sum(arr1)
```

Out[52]:
```
45
```

In [53]:
```
np.sum(arr1,axis = 0)
```

Out[53]:
```
array([12, 15, 18])
```

In [54]:
```
np.sum(arr1,axis = 1)
```

Out[54]:
```
array([ 6, 15, 24])
```

In [55]:
```
np.mean(arr1)
```

Out[55]:
```
5.0
```

In [56]:
```
np.sqrt(arr1)
```

Out[56]:
```
array([[1.        , 1.41421356, 1.73205081],
       [2.        , 2.23606798, 2.44948974],
       [2.64575131, 2.82842712, 3.        ]])
```

In [57]:
```
np.std(arr1)
```

Out[57]:
```
2.581988897471611
```

In [58]:
```
np.exp(arr1)
```

Out[58]:
```
array([[2.71828183e+00, 7.38905610e+00, 2.00855369e+01],
       [5.45981500e+01, 1.48413159e+02, 4.03428793e+02],
       [1.09663316e+03, 2.98095799e+03, 8.10308393e+03]])
```

In [59]:
```
np.log(arr1)
```

Out[59]:
```
array([[0.        , 0.69314718, 1.09861229],
       [1.38629436, 1.60943791, 1.79175947],
       [1.94591015, 2.07944154, 2.19722458]])
```

In [60]:
```
np.log10(arr1)
```

Out[60]:
```
array([[0.        , 0.30103   , 0.47712125],
       [0.60205999, 0.69897   , 0.77815125],
       [0.84509804, 0.90308999, 0.95424251]])
```

Python NumPy array_slicing()

In [61]:
```
import numpy as np
```

In [62]:
```
mx = np.arange(1,101).reshape(10,10)
print(mx)
```

```
[[  1   2   3   4   5   6   7   8   9  10]
 [ 11  12  13  14  15  16  17  18  19  20]
 [ 21  22  23  24  25  26  27  28  29  30]
 [ 31  32  33  34  35  36  37  38  39  40]
 [ 41  42  43  44  45  46  47  48  49  50]
 [ 51  52  53  54  55  56  57  58  59  60]
 [ 61  62  63  64  65  66  67  68  69  70]
 [ 71  72  73  74  75  76  77  78  79  80]
 [ 81  82  83  84  85  86  87  88  89  90]
 [ 91  92  93  94  95  96  97  98  99 100]]
```

In [63]:
```
mx[0,0]
```

Out[63]:
```
1
```

In [64]:
```
mx[2,6]
```

Out[64]:
```
27
```

In [65]:
```
mx[2,6].ndim
```

Out[65]:
```
0
```

In [66]:
```
mx[0]
```

Out[66]:
```
array([ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10])
```

In [67]:
```
mx[:,0]
```

Out[67]:
```
array([ 1, 11, 21, 31, 41, 51, 61, 71, 81, 91])
```

In [68]:
```
mx[:,0:1]
```

Out[68]:
```
array([[ 1],
       [11],
       [21],
       [31],
       [41],
       [51],
       [61],
       [71],
       [81],
       [91]])
```

In [69]:
```
mx[:,0:1].ndim
```

Out[69]:
```
2
```

In [70]:
```
mx[1:4,1:4]
```

Out[70]:
```
array([[12, 13, 14],
       [22, 23, 24],
       [32, 33, 34]])
```

In [71]:
```
mx[:,1:3]
```

Out[71]:
```
array([[ 2,  3],
       [12, 13],
       [22, 23],
       [32, 33],
       [42, 43],
       [52, 53],
       [62, 63],
       [72, 73],
       [82, 83],
       [92, 93]])
```

In [72]:
```
# mx[:]
```

In [73]:
```
# mx[::]
```

In [74]:
```
# mx[:,:]
```

In [75]:
```
mx.itemsize
```

Out[75]:
```
4
```

In [76]:
```
mx.dtype
```

Out[76]:
```
dtype('int32')
```

In [77]:
```
32/8
```

Out[77]:
```
4.0
```

Python NumPy array Conctination and split ()

In [78]:
```
import numpy as np
```

In [79]:
```
arr1 = np.arange(1,17).reshape(4,4)
print(arr1)
```

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

In [80]:
```
arr2 = np.arange(17,33).reshape(4,4)
print(arr2)
```

```
[[17 18 19 20]
 [21 22 23 24]
 [25 26 27 28]
 [29 30 31 32]]
```

In [81]:
```
# Concatenate
np.concatenate((arr1,arr2))
```

Out[81]:
```
array([[ 1,  2,  3,  4],
       [ 5,  6,  7,  8],
       [ 9, 10, 11, 12],
       [13, 14, 15, 16],
       [17, 18, 19, 20],
       [21, 22, 23, 24],
       [25, 26, 27, 28],
       [29, 30, 31, 32]])
```

In [82]:
```
np.concatenate((arr1,arr2),axis = 1)
```

Out[82]:
```
array([[ 1,  2,  3,  4, 17, 18, 19, 20],
       [ 5,  6,  7,  8, 21, 22, 23, 24],
       [ 9, 10, 11, 12, 25, 26, 27, 28],
       [13, 14, 15, 16, 29, 30, 31, 32]])
```

In [83]:
```
np.vstack((arr1,arr2))
```

Out[83]:
```
array([[ 1,  2,  3,  4],
       [ 5,  6,  7,  8],
       [ 9, 10, 11, 12],
       [13, 14, 15, 16],
       [17, 18, 19, 20],
       [21, 22, 23, 24],
       [25, 26, 27, 28],
       [29, 30, 31, 32]])
```

In [84]:
```
np.hstack((arr1,arr2))
```

Out[84]:
```
array([[ 1,  2,  3,  4, 17, 18, 19, 20],
       [ 5,  6,  7,  8, 21, 22, 23, 24],
       [ 9, 10, 11, 12, 25, 26, 27, 28],
       [13, 14, 15, 16, 29, 30, 31, 32]])
```

In [85]:
```
arr3 = np.arange(33,49).reshape(4,4)
```

In [86]:
```
np.hstack((arr1,arr2,arr3))
```

Out[86]:
```
array([[ 1,  2,  3,  4, 17, 18, 19, 20, 33, 34, 35, 36],
       [ 5,  6,  7,  8, 21, 22, 23, 24, 37, 38, 39, 40],
       [ 9, 10, 11, 12, 25, 26, 27, 28, 41, 42, 43, 44],
       [13, 14, 15, 16, 29, 30, 31, 32, 45, 46, 47, 48]])
```

In [87]:
```
np.split(arr1,2)
```

Out[87]:
```
[array([[1, 2, 3, 4],
        [5, 6, 7, 8]]),
 array([[ 9, 10, 11, 12],
        [13, 14, 15, 16]])]
```

In [88]:
```
list1 = np.split(arr1,2)
type(list1)
```

Out[88]:
```
list
```

In [89]:
```
list1[0]
```

Out[89]:
```
array([[1, 2, 3, 4],
       [5, 6, 7, 8]])
```

In [90]:
```
type(list1[0])
```

Out[90]:
```
numpy.ndarray
```

In [91]:
```
np.split(arr1,2,axis = 1)
```

Out[91]:
```
[array([[ 1,  2],
        [ 5,  6],
        [ 9, 10],
        [13, 14]]),
 array([[ 3,  4],
        [ 7,  8],
        [11, 12],
        [15, 16]])]
```

In [92]:
```
_1d = np.array([4,5,6,7,8])
```

In [93]:
```
np.split(_1d,[1,3])
```

Out[93]:
```
[array([4]), array([5, 6]), array([7, 8])]
```

Find Trignometry sin(), cos(), and tan() using NumPy Trignometry Functions

In [94]:
```
import numpy as np
import matplotlib.pyplot as plt
```

In [95]:
```
np.sin(180)
```

Out[95]:
```
-0.8011526357338304
```

In [96]:
```
# np.sin(180 * np.pi/180)
```

In [97]:
```
np.sin(90)
```

Out[97]:
```
0.8939966636005579
```

In [98]:
```
np.cos(180)
```

Out[98]:
```
-0.5984600690578582
```

In [99]:
```
np.tan(180)
```

Out[99]:
```
1.3386902103511544
```

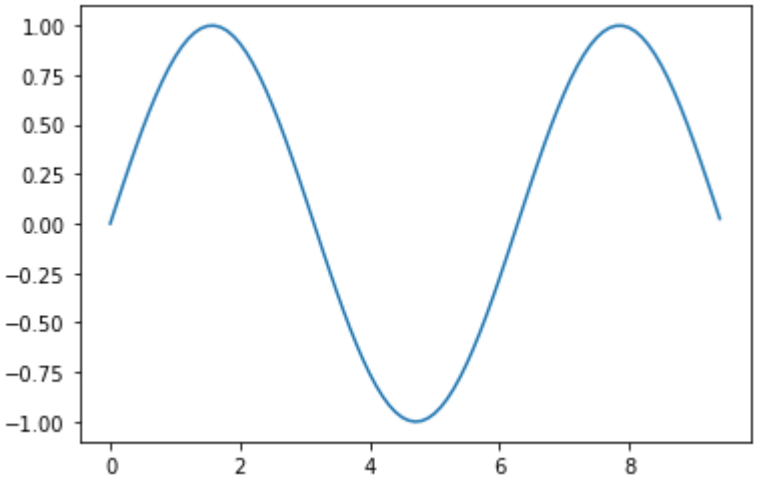
In [100…
```
x_sin = np.arange(0,3*np.pi,0.1)
print(x_sin)
```

```
[0.  0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.  1.1 1.2 1.3 1.4 1.5 1.6 1.7
 1.8 1.9 2.  2.1 2.2 2.3 2.4 2.5 2.6 2.7 2.8 2.9 3.  3.1 3.2 3.3 3.4 3.5
 3.6 3.7 3.8 3.9 4.  4.1 4.2 4.3 4.4 4.5 4.6 4.7 4.8 4.9 5.  5.1 5.2 5.3
 5.4 5.5 5.6 5.7 5.8 5.9 6.  6.1 6.2 6.3 6.4 6.5 6.6 6.7 6.8 6.9 7.  7.1
 7.2 7.3 7.4 7.5 7.6 7.7 7.8 7.9 8.  8.1 8.2 8.3 8.4 8.5 8.6 8.7 8.8 8.9
 9.  9.1 9.2 9.3 9.4]
```

In [101…
```
y_sin = np.sin(x_sin)
print(y_sin)
```

```
[ 0.          0.09983342  0.19866933  0.29552021  0.38941834  0.47942554
  0.56464247  0.64421769  0.71735609  0.78332691  0.84147098  0.89120736
  0.93203909  0.96355819  0.98544973  0.99749499  0.9995736   0.99166481
  0.97384763  0.94630009  0.90929743  0.86320937  0.8084964   0.74570521
  0.67546318  0.59847214  0.51550137  0.42737988  0.33498815  0.23924933
  0.14112001  0.04158066 -0.05837414 -0.15774569 -0.2555411  -0.35078323
 -0.44252044 -0.52983614 -0.61185789 -0.68776616 -0.7568025  -0.81827711
 -0.87157577 -0.91616594 -0.95160207 -0.97753012 -0.993691   -0.99992326
 -0.99616461 -0.98245261 -0.95892427 -0.92581468 -0.88345466 -0.83226744
 -0.77276449 -0.70554033 -0.63126664 -0.55068554 -0.46460218 -0.37387666
 -0.2794155  -0.1821625  -0.0830894   0.0168139   0.1165492   0.21511999
  0.31154136  0.40484992  0.49411335  0.57843976  0.6569866   0.72896904
  0.79366786  0.85043662  0.8987081   0.93799998  0.96791967  0.98816823
  0.99854335  0.99894134  0.98935825  0.96988981  0.94073056  0.90217183
  0.85459891  0.79848711  0.7343971   0.66296923  0.58491719  0.50102086
  0.41211849  0.31909836  0.22288991  0.12445442  0.02477543]
```

In [102…
```
plt.plot(x_sin,y_sin)
plt.show()
```

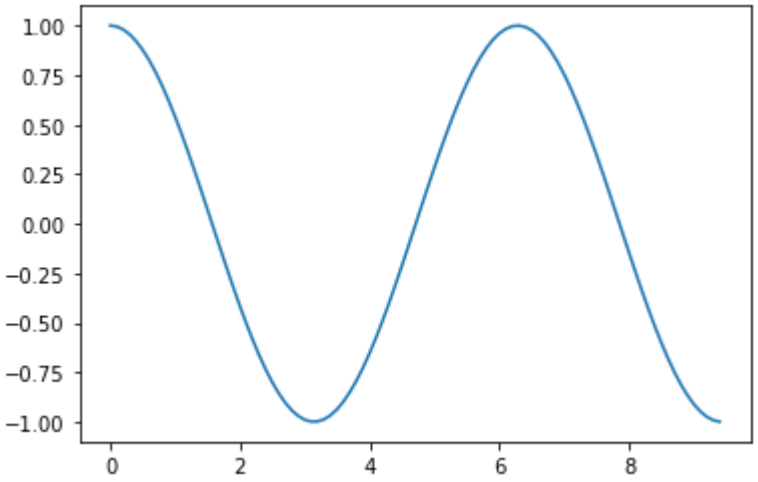


In [103…
```
y_cos = np.cos(x_sin)
print(y_cos)
```

```
[ 1.          0.99500417  0.98006658  0.95533649  0.92106099  0.87758256
  0.82533561  0.76484219  0.69670671  0.62160997  0.54030231  0.45359612
  0.36235775  0.26749883  0.16996714  0.0707372  -0.02919952 -0.12884449
 -0.22720209 -0.32328957 -0.41614684 -0.5048461  -0.58850112 -0.66627602
 -0.73739372 -0.80114362 -0.85688875 -0.90407214 -0.94222234 -0.97095817
 -0.9899925  -0.99913515 -0.99829478 -0.98747977 -0.96679819 -0.93645669
 -0.89675842 -0.84810003 -0.79096771 -0.7259323  -0.65364362 -0.57482395
 -0.49026082 -0.40079917 -0.30733287 -0.2107958  -0.11215253 -0.01238866
  0.08749898  0.18651237  0.28366219  0.37797774  0.46851667  0.55437434
  0.63469288  0.70866977  0.77556588  0.83471278  0.88551952  0.92747843
  0.96017029  0.98326844  0.9965421   0.99985864  0.99318492  0.97658763
  0.95023259  0.91438315  0.86939749  0.8157251   0.75390225  0.68454667
  0.60835131  0.52607752  0.43854733  0.34663532  0.25125984  0.15337386
  0.05395542 -0.04600213 -0.14550003 -0.24354415 -0.33915486 -0.43137684
 -0.51928865 -0.6020119  -0.67872005 -0.74864665 -0.81109301 -0.86543521
 -0.91113026 -0.9477216  -0.97484362 -0.99222533 -0.99969304]
```

In [104…

```
plt.plot(x_sin,y_cos)
plt.show()
```

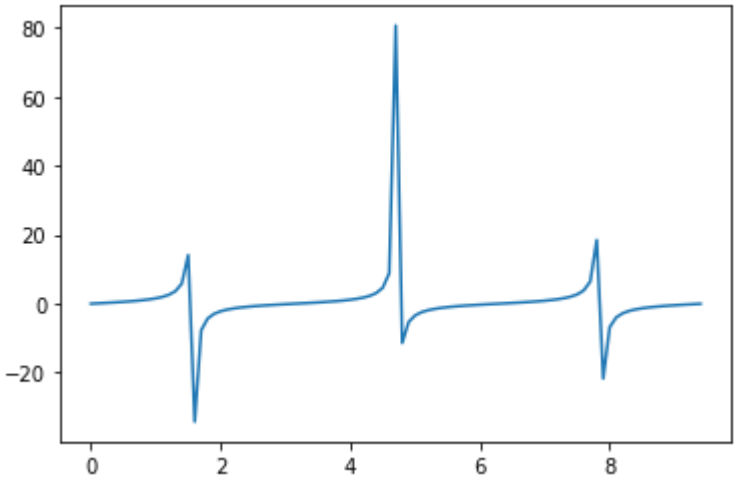


In [105…

```
y_tan = np.tan(x_sin)
print(y_tan)
```

```
[ 0.00000000e+00  1.00334672e-01  2.02710036e-01  3.09336250e-01
  4.22793219e-01  5.46302490e-01  6.84136808e-01  8.42288380e-01
  1.02963856e+00  1.26015822e+00  1.55740772e+00  1.96475966e+00
  2.57215162e+00  3.60210245e+00  5.79788372e+00  1.41014199e+01
 -3.42325327e+01 -7.69660214e+00 -4.28626167e+00 -2.92709751e+00
 -2.18503986e+00 -1.70984654e+00 -1.37382306e+00 -1.11921364e+00
 -9.16014290e-01 -7.47022297e-01 -6.01596613e-01 -4.72727629e-01
 -3.55529832e-01 -2.46405394e-01 -1.42546543e-01 -4.16166546e-02
  5.84738545e-02  1.59745748e-01  2.64316901e-01  3.74585640e-01
  4.93466730e-01  6.24733075e-01  7.73556091e-01  9.47424650e-01
  1.15782128e+00  1.42352648e+00  1.77777977e+00  2.28584788e+00
  3.09632378e+00  4.63733205e+00  8.86017490e+00  8.07127630e+01
 -1.13848707e+01 -5.26749307e+00 -3.38051501e+00 -2.44938942e+00
 -1.88564188e+00 -1.50127340e+00 -1.21754082e+00 -9.95584052e-01
 -8.13943284e-01 -6.59730572e-01 -5.24666222e-01 -4.03110900e-01
 -2.91006191e-01 -1.85262231e-01 -8.33777149e-02  1.68162777e-02
  1.17348947e-01  2.20277200e-01  3.27858007e-01  4.42757417e-01
  5.68339979e-01  7.09111151e-01  8.71447983e-01  1.06489313e+00
  1.30462094e+00  1.61656142e+00  2.04928417e+00  2.70601387e+00
  3.85226569e+00  6.44287247e+00  1.85068216e+01 -2.17151127e+01
 -6.79971146e+00 -3.98239825e+00 -2.77374930e+00 -2.09137751e+00
 -1.64571073e+00 -1.32636433e+00 -1.08203242e+00 -8.85556937e-01
 -7.21146876e-01 -5.78923588e-01 -4.52315659e-01 -3.36700526e-01
 -2.28641712e-01 -1.25429598e-01 -2.47830328e-02]
```

In [106…
```
plt.plot(x_sin,y_tan)
plt.show()
```



Random Sampling with NumPy

In [107…
```
import numpy as np
import random
```

In [108…
```
np.random.random(1)
```

Out[108]:
```
array([0.74363081])
```

In [109…
```
np.random.random((3,3))
```

Out[109]:
```
array([[0.37085518, 0.7827431 , 0.95679292],
       [0.10894362, 0.42139576, 0.65819178],
       [0.73286477, 0.59318662, 0.49459163]])
```

In [110…
```
np.random.randint(1,4)
```

Out[110]:
```
3
```

In [111…
```
np.random.randint(1,4,(4,4))
```

Out[111]:
```
array([[3, 1, 2, 2],
       [1, 1, 3, 2],
       [3, 3, 3, 3],
       [3, 2, 3, 2]])
```

In [112…
```
np.random.randint(1,4,(2,4,4))
```

Out[112]:
```
array([[[2, 1, 1, 1],
        [2, 3, 2, 3],
        [3, 2, 1, 3],
        [1, 2, 2, 1]],

       [[3, 3, 1, 2],
        [3, 1, 3, 1],
        [3, 3, 1, 2],
        [3, 2, 1, 2]]])
```

In [113…
```
# 2**32 -1
np.random.seed(10)
np.random.randint(1,4,(2,4,4))
```

Out[113]:
```
array([[[2, 2, 1, 1],
        [2, 1, 2, 2],
        [1, 2, 2, 3],
        [1, 2, 1, 3]],

       [[1, 3, 1, 1],
        [1, 3, 1, 3],
        [3, 2, 1, 1],
        [3, 2, 3, 2]]])
```

In [114…
```
np.random.rand(3)
```

Out[114]:
```
array([0.13145815, 0.41366737, 0.77872881])
```

In [115…
```
np.random.rand(3,3)
```

Out[115]:
```
array([[0.58390137, 0.18263144, 0.82608225],
       [0.10540183, 0.28357668, 0.06556327],
       [0.05644419, 0.76545582, 0.01178803]])
```

In [116…
```
# std norm dist
np.random.randn(3,3)
```

Out[116]:
```
array([[-1.58494101,  1.05535316, -1.92657911],
       [ 0.69858388, -0.74620143, -0.15662666],
       [-0.19363594,  1.13912535,  0.36221796]])
```

In [117…
```
x = [1,2,3,4]
np.random.choice(x)
```

Out[117]:
```
1
```

In [118…]
```
for i in range(10):
    print(np.random.choice(x))
```

```
1
1
2
1
4
1
1
3
1
3
```

In [119…]
```
np.random.permutation(x)
```

Out[119]:
```
array([4, 2, 1, 3])
```

In [120…]
```
# np.random.shuffle(x)
```

String Operatione, Comparison and Information

In [121…]
```
import numpy as np
```

In [122…]
```
pip = "Python"
lib = "NumPy"
```

In [123…]
```
np.char.add(pip,lib)
```

Out[123]:
```
array('PythonNumPy', dtype='<U11')
```

In [124…]
```
np.char.lower(pip)
```

Out[124]:
```
array('python', dtype='<U6')
```

In [125…]
```
np.char.upper(lib)
```

Out[125]:
```
array('NUMPY', dtype='<U5')
```

In [126…]
```
np.char.center(lib,75,fillchar="_")
```

Out[126]:
```
array('_____________________________________NumPy_____________________________________
_____',
      dtype='<U75')
```

In [127…]
```
# Split
pack = "Python NumPy Learning"
```

In [128…]
```
np.char.split(pack)
```

Out[128]:
```
array(list(['Python', 'NumPy', 'Learning']), dtype=object)
```

In [129…]
```
np.char.splitlines("NumPy\nPython")
```

Out[129]:
```
array(list(['NumPy', 'Python']), dtype=object)
```

In [130…
```
str1 = "dmy"
str2 = "dmy"

np.char.join([":","/"],[str1,str2])
```

Out[130]:
```
array(['d:m:y', 'd/m/y'], dtype='<U5')
```

In [131…
```
np.char.replace(pack,"Learning","Participation")
```

Out[131]:
```
array('Python NumPy Participation', dtype='<U26')
```

In [132…
```
np.char.equal(str1,str2)
```

Out[132]:
```
array(True)
```

In [133…
```
np.char.count(pack,"n")
```

Out[133]:
```
array(3)
```

In [134…
```
np.char.find(pack,"N")
```

Out[134]:
```
array(7)
```

CampusX

In [135…
```
import numpy as np
```

In [136…
```
arr1 = np.array([1,2,3,4,5])
arr1
```

Out[136]:
```
array([1, 2, 3, 4, 5])
```

In [137…
```
type(arr1)
```

Out[137]:
```
numpy.ndarray
```

In [138…
```
arr2 = np.array([[1,2,3],[4,5,6]])
arr2
```

Out[138]:
```
array([[1, 2, 3],
       [4, 5, 6]])
```

In [139…
```
arr3 = np.zeros((2,3))
arr3
```

Out[139]:
```
array([[0., 0., 0.],
       [0., 0., 0.]])
```

In [140…
```
arr4 = np.ones((3,3))
arr4
```

Out[140]:
```
array([[1., 1., 1.],
       [1., 1., 1.],
       [1., 1., 1.]])
```

In [141…
```
arr5 = np.identity((5))
arr5
```

Out[141]:
```
array([[1., 0., 0., 0., 0.],
       [0., 1., 0., 0., 0.],
       [0., 0., 1., 0., 0.],
       [0., 0., 0., 1., 0.],
       [0., 0., 0., 0., 1.]])
```

In [142…
```
arr6 = np.arange(5,16,2)
arr6
```

Out[142]:
```
array([ 5,  7,  9, 11, 13, 15])
```

In [143…
```
arr7 = np.linspace(10,20,10)
arr7
```

Out[143]:
```
array([10.        , 11.11111111, 12.22222222, 13.33333333, 14.44444444,
       15.55555556, 16.66666667, 17.77777778, 18.88888889, 20.        ])
```

In [144…
```
# Copy
arr8 = arr7.copy()
arr8
```

Out[144]:
```
array([10.        , 11.11111111, 12.22222222, 13.33333333, 14.44444444,
       15.55555556, 16.66666667, 17.77777778, 18.88888889, 20.        ])
```

In [145…
```
arr1.shape
```

Out[145]:
```
(5,)
```

In [146…
```
arr2.shape
```

Out[146]:
```
(2, 3)
```

In [147…
```
# 3
arr9 = np.array([[[1,2],[3,4]],[[5,6],[7,8]]])
```

In [148…
```
arr9.shape
```

Out[148]:
```
(2, 2, 2)
```

In [149…
```
arr9.ndim
```

Out[149]:
```
3
```

In [150…
```
arr9.size
```

Out[150]:
```
8
```

In [151…
```
arr9.itemsize
```

Out[151]:
```
4
```

In [152…
```
arr8.itemsize
```

Out[152]:
```
8
```

In [153…
```
arr8.dtype
```

Out[153]:
```
dtype('float64')
```

In [154…
```
arr9.dtype
```

Out[154]:
```
dtype('int32')
```

In [155…
```
arr9.astype('float')
```

Out[155]:
```
array([[[1., 2.],
        [3., 4.]],

       [[5., 6.],
        [7., 8.]]])
```

In [156…
```
list1 = range(100)
arr11 = np.arange(100)
```

In [157…
```
import sys
```

In [158…
```
print(sys.getsizeof(87)*len(list1))
```

```
2800
```

In [159…
```
print(arr11.itemsize*arr11.size)
```

```
400
```

In [160…
```
import time
```

In [161…
```
x = range(100000)
y = range(100000,200000)

start_time = time.time()

# c = [(x,y) for x,y in zip (x,y)]
c = [x + y for x,y in zip (x,y)]

print(time.time() - start_time)
```

```
0.010995864868164062
```

In [162…
```
a = np.arange(100000)
b = np.arange(100000,200000)

start_time = time.time()

c = c + b

print(time.time() - start_time)
```

```
0.009999513626098633
```

In [163…
```
arr12 = np.arange(24).reshape(6,4)
arr12
```

Out[163]:
```
array([[ 0,  1,  2,  3],
       [ 4,  5,  6,  7],
       [ 8,  9, 10, 11],
       [12, 13, 14, 15],
       [16, 17, 18, 19],
       [20, 21, 22, 23]])
```

In [164…
```
arr12[:,1:3]
```

Out[164]:
```
array([[ 1,  2],
       [ 5,  6],
       [ 9, 10],
       [13, 14],
       [17, 18],
       [21, 22]])
```

In [165…
```
arr12[2:4,1:3]
```

Out[165]:
```
array([[ 9, 10],
       [13, 14]])
```

In [166…
```
arr12[4:,2:]
```

Out[166]:
```
array([[18, 19],
       [22, 23]])
```

In [167…
```
for i in arr12:
    print(i)
```

```
[0 1 2 3]
[4 5 6 7]
[ 8  9 10 11]
[12 13 14 15]
[16 17 18 19]
[20 21 22 23]
```

In [168…
```
for i in np.nditer(arr12):
    print(i)
```

```
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
```

In [169…
```
arr_1 = np.array([1,2,3,4,5,6])
arr_2 = np.array([4,5,6,7,8,9])
```

In [170…
```
arr_1 - arr_2
```

Out[170]:
```
array([-3, -3, -3, -3, -3, -3])
```

In [171…
```
arr_1 * arr_2
```

Out[171]:
```
array([ 4, 10, 18, 28, 40, 54])
```

In [173…
```
arr_1 * 2
```

Out[173]:
```
array([ 2,  4,  6,  8, 10, 12])
```

In [174…
```
arr_2 > 3
```

Out[174]:
```
array([ True,  True,  True,  True,  True,  True])
```

In [175…
```
arr_3 = np.arange(6).reshape(2,3)
arr_4 = np.arange(6,12).reshape(3,2)
```

In [176…
```
arr_3.dot(arr_4)
```

Out[176]:
```
array([[ 28,  31],
       [100, 112]])
```

In [181…
```
arr_1.dot(arr_2)
```

Out[181]:
```
154
```

In [183…
```
arr_4.max()
```

Out[183]:
```
11
```

In [184…
```
arr_4.min()
```

Out[184]:
```
6
```

In [185…
```
arr_4.min(axis = 0)
```

Out[185]:
```
array([6, 7])
```

In [186…
```
arr_4.min(axis = 1)
```

Out[186]:
```
array([ 6,  8, 10])
```

In [187…
```
arr_4.sum()
```

Out[187]:
```
51
```

In [188…
```
arr_4.sum(axis = 0)
```

Out[188]:
```
array([24, 27])
```

In [189…
```
arr_4.sum(axis = 1)
```

Out[189]:
```
array([13, 17, 21])
```

In [190…
```
arr_4.mean()
```

Out[190]:
```
8.5
```

In [191…
```
arr_4.std()
```

Out[191]:
```
1.707825127659933
```

In [193…
```
np.sin(arr_4)
```

Out[193]:
```
array([[-0.2794155 ,  0.6569866 ],
       [ 0.98935825,  0.41211849],
       [-0.54402111, -0.99999021]])
```

In [195…
```
np.median(arr_4)
```

Out[195]:
```
8.5
```

In [196…
```
np.exp(arr_4)
```

Out[196]:
```
array([[  403.42879349,  1096.63315843],
       [ 2980.95798704,  8103.08392758],
       [22026.46579481, 59874.1417152 ]])
```

Reshaping_NumPy_Array

In [198…
```
arr_4
```

Out[198]:
```
array([[ 6,  7],
       [ 8,  9],
       [10, 11]])
```

In [200…
```
arr_4.ndim
```

Out[200]:
```
2
```

In [201…
```
arr_4.ravel()
```

Out[201]:
```
array([ 6,  7,  8,  9, 10, 11])
```

In [203…
```
# Row_Col * Col_Row
arr_4.transpose()
```

Out[203]:
```
array([[ 6,  8, 10],
       [ 7,  9, 11]])
```

In [206…
```
arr_5 = np.arange(12,18).reshape(2,3)
```

In [209…
```
arr_5
```

Out[209]:
```
array([[12, 13, 14],
       [15, 16, 17]])
```

In [210…
```
np.hstack((arr_3,arr_5))
```

Out[210]:
```
array([[ 0,  1,  2, 12, 13, 14],
       [ 3,  4,  5, 15, 16, 17]])
```

In [211…
```
np.vstack((arr_3,arr_5))
```

Out[211]:
```
array([[ 0,  1,  2],
       [ 3,  4,  5],
       [12, 13, 14],
       [15, 16, 17]])
```

In [213…
```
np.hsplit(arr_3,3)
```

Out[213]:
```
[array([[0],
        [3]]),
 array([[1],
        [4]]),
 array([[2],
        [5]])]
```

In [214…
```
np.vsplit(arr_3,2)
```

Out[214]:
```
[array([[0, 1, 2]]), array([[3, 4, 5]])]
```

In [215…
```
arr_8 = np.arange(24).reshape(6,4)
```

In [216…
```
arr_8
```

Out[216]:
```
array([[ 0,  1,  2,  3],
       [ 4,  5,  6,  7],
       [ 8,  9, 10, 11],
       [12, 13, 14, 15],
       [16, 17, 18, 19],
       [20, 21, 22, 23]])
```

In [218…
```
arr_8[[0,2,4]]
```

Out[218]:
```
array([[ 0,  1,  2,  3],
       [ 8,  9, 10, 11],
       [16, 17, 18, 19]])
```

Indexing Useing Boolean Array

In [223…
```
arr = np.random.randint(low = 1,high = 100,size = 20).reshape(4,5)
arr
```

Out[223]:
```
array([[28, 83, 63, 78, 49],
       [94, 76, 87, 38, 12],
       [22, 34, 96, 44, 89],
       [97, 74, 41, 44, 91]])
```

In [224…
```
arr[0]
```

Out[224]:
```
array([28, 83, 63, 78, 49])
```

In [232…
```
arr > 50
```

Out[232]:
```
array([[False,  True,  True,  True, False],
       [ True,  True,  True, False, False],
       [False, False,  True, False,  True],
       [ True,  True, False, False,  True]])
```

In [233…
```
(arr > 50).shape
```

Out[233]:
```
(4, 5)
```

In [237…
```
arr[arr > 50]
```

Out[237]:
```
array([83, 63, 78, 94, 76, 87, 96, 89, 97, 74, 91])
```

In [238…
```
arr[(arr > 50) & (arr % 2 != 0)]
```

Out[238]:
```
array([83, 63, 87, 89, 97, 91])
```

In [240…
```
arr[(arr > 50) & (arr % 2 != 0)] = 0
```

In [241…
```
arr
```

Out[241]:
```
array([[28,  0,  0, 78, 49],
       [94, 76,  0, 38, 12],
       [22, 34, 96, 44,  0],
       [ 0, 74, 41, 44,  0]])
```

In [242…
```
x = np.linspace(-40,40,100)
```

In [244…
```
x.size
```

Out[244]: 100

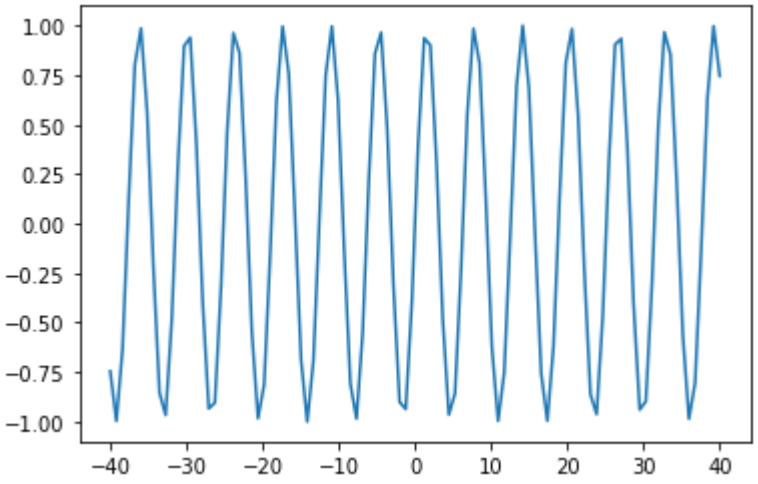
In [245…
```
y = np.sin(x)
```

In [247…
```
y.size
```

Out[247]: 100

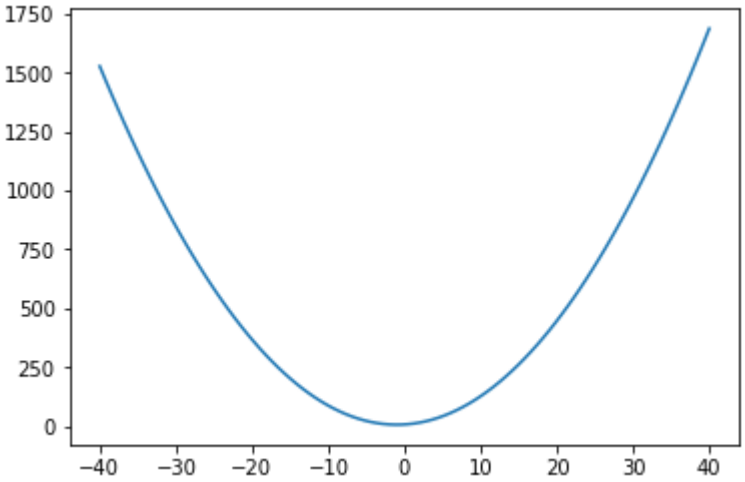
In [249…
```
import matplotlib.pyplot as plt
%matplotlib inline
```

In [252…
```
plt.plot(x,y)
plt.show()
```



In [253…
```
y = x * x + 2 * x + 6
```

In [254…
```
plt.plot(x,y)
plt.show()
```



Broadcasting

In [256…
```
a_1 = np.arange(8).reshape(2,4)
a_2 = np.arange(8,16).reshape(2,4)
```

```
print(a_1)
print(a_2)
```

```
[[0 1 2 3]
 [4 5 6 7]]
[[ 8  9 10 11]
 [12 13 14 15]]
```

In [257…
```
a_1 + a_2
```

Out[257]:
```
array([[ 8, 10, 12, 14],
       [16, 18, 20, 22]])
```

In [260…
```
a_3 = np.arange(9).reshape(3,3)
a_4 = np.arange(3).reshape(1,3)

print(a_3)
print(a_4)
```

```
[[0 1 2]
 [3 4 5]
 [6 7 8]]
[[0 1 2]]
```

In [261…
```
a_3 + a_4
```

Out[261]:
```
array([[ 0,  2,  4],
       [ 3,  5,  7],
       [ 6,  8, 10]])
```

In [264…
```
a_5 = np.arange(3).reshape(1,3)
a_6 = np.arange(12).reshape(4,3)

print(a_5)
print(a_6)
```

```
[[0 1 2]]
[[ 0  1  2]
 [ 3  4  5]
 [ 6  7  8]
 [ 9 10 11]]
```

In [265…
```
a_5 + a_6
```

Out[265]:
```
array([[ 0,  2,  4],
       [ 3,  5,  7],
       [ 6,  8, 10],
       [ 9, 11, 13]])
```

Important NumPy Functions

CampusX

In [266…
```
import numpy as np
```

In [272…
```
np.random.random()
```

Out[272]:
```
0.1033810697475811
```

In [275…
```
np.random.seed(1)
np.random.random()
```

Out[275]:
```
0.417022004702574
```

In [281…
```
np.random.uniform(1,100)
```

Out[281]:
```
10.141520882110981
```

In [283…
```
np.random.uniform(1,100,10).reshape(2,5)
```

Out[283]:
```
array([[42.31317543, 56.31029302, 14.89830692, 20.61204742, 80.2737123 ],
       [96.857896  , 32.02899364, 69.53993895, 87.76252608, 89.56605969]])
```

In [286…
```
np.random.randint(1,10,15).reshape(3,5)
```

Out[286]:
```
array([[7, 6, 2, 4, 5],
       [9, 2, 5, 1, 4],
       [3, 1, 5, 3, 8]])
```

In [299…
```
a = np.random.randint(1,10,6)
print(a)
```

```
[3 8 8 8 4 1]
```

In [289…
```
np.max(a)
```

Out[289]:
```
9
```

In [290…
```
np.min(a)
```

Out[290]:
```
1
```

In [301…
```
a[np.argmax(a)]
```

Out[301]:
```
8
```

In [300…
```
a[np.argmin(a)]
```

Out[300]:
```
1
```

In [306…
```
a
```

Out[306]:
```
array([-1,  8,  8,  8,  4, -1])
```

In [305…
```
a[a % 2 == 1] = -1
a
```

Out[305]:
```
array([-1,  8,  8,  8,  4, -1])
```

In [311…
```
b = np.random.randint(1,50,6)
print(b)
```

```
[37 40  8 46  5 49]
```

In [312…
```
np.where(b % 2 == 1,-1,b)
```

Out[312]:
```
array([-1, 40,  8, 46, -1, -1])
```

In [313…
```
b
```

Out[313]:
```
array([37, 40,  8, 46,  5, 49])
```

In [314…
```
out = np.where(b % 2 == 1,-1,b)
out
```

Out[314]:
```
array([-1, 40,  8, 46, -1, -1])
```

In [316…
```
c = np.random.randint(1,50,10)
print(c)
```

```
[21 33 13  2 31 42 25 19 34  3]
```

In [318…
```
np.sort(c)
c
```

Out[318]:
```
array([21, 33, 13,  2, 31, 42, 25, 19, 34,  3])
```

In [319…
```
np.percentile(c,25)
```

Out[319]:
```
14.5
```

In [320…
```
np.percentile(c,50)
```

Out[320]:
```
23.0
```

In [323…
```
np.percentile(c,99.8)
```

Out[323]:
```
41.855999999999995
```

Thank You

CONNECT WITH ME :

LinkedIn GitHub kaggle Medium

PRASADMJADHAV2