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

Array Manipulation

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

In [5]:
```
1 myarr
```

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

Append an item

In [6]:
```
1 np.append(arr = myarr, values = 10)
```

Out[6]:
```
array([ 1,  2,  3, 10])
```

In [7]:
```
1 myarr
```

Out[7]:
```
array([1, 2, 3])
```

In [8]:
```
1 myarr = np.append(arr = myarr, values = 10)
```

In [9]:
```
1 myarr
```

Out[9]:
```
array([ 1,  2,  3, 10])
```

In [10]:
```
1 # How to add multiple elements in the array
2 myarr = np.append(arr = myarr, values = [5,7])
```

In [11]:
```
1 myarr
```

Out[11]:
```
array([ 1,  2,  3, 10,  5,  7])
```

In [13]:
```
1 myarr = np.append(arr = myarr, values = (11,12))
```

In [14]:
```
1 myarr
```

Out[14]:
```
array([ 1,  2,  3, 10,  5,  7, 11, 12])
```

In [16]:
```
myarr = np.append(arr = myarr, values = np.array([55,77]))
```

In [17]:
```
myarr
```

Out[17]:
```
array([ 1,  2,  3, 10,  5,  7, 11, 12, 55, 77])
```

Removing particular element from the array

In [18]:
```
np.delete(arr = myarr, obj=-3)
```

Out[18]:
```
array([ 1,  2,  3, 10,  5,  7, 11, 55, 77])
```

Can we delete multiple elements

In [19]:
```
np.delete(myarr, [3,-2, 0])
```

Out[19]:
```
array([ 2,  3,  5,  7, 11, 12, 77])
```

How we can replace an element / item

In [20]:
```
myarr
```

Out[20]:
```
array([ 1,  2,  3, 10,  5,  7, 11, 12, 55, 77])
```

In [21]:
```
myarr[3]
```

Out[21]:
```
10
```

In [22]:
```
myarr[3] = 99
```

In [23]:
```
myarr
```

Out[23]:
```
array([ 1,  2,  3, 99,  5,  7, 11, 12, 55, 77])
```

In [24]:
```
myarr
```

Out[24]:
```
array([ 1,  2,  3, 99,  5,  7, 11, 12, 55, 77])
```

In [25]:
```
np.delete(myarr, np.where(myarr == 11))
```

Out[25]:
```
array([ 1,  2,  3, 99,  5,  7, 12, 55, 77])
```

In [26]:
```
1 np.delete(myarr, np.where(myarr % 11 == 0))
```

Out[26]:
```
array([ 1,  2,  3,  5,  7, 12])
```

In [27]:
```
1 myarr
```

Out[27]:
```
array([ 1,  2,  3, 99,  5,  7, 11, 12, 55, 77])
```

In [28]:
```
1 myarr[-2:]
```

Out[28]:
```
array([55, 77])
```

In [29]:
```
1 myarr[-2:] = [10,20]
```

In [30]:
```
1 myarr
```

Out[30]:
```
array([ 1,  2,  3, 99,  5,  7, 11, 12, 10, 20])
```

In [32]:
```
1 myarr[[3,8]]
```

Out[32]:
```
array([99, 10])
```

In [33]:
```
1 myarr[[3,8]] = [55,99]
```

In [34]:
```
1 myarr
```

Out[34]:
```
array([ 1,  2,  3, 55,  5,  7, 11, 12, 99, 20])
```

In [35]:
```
1 myarr[[3,8]] = [100]
```

In [36]:
```
1 myarr
```

Out[36]:
```
array([  1,   2,   3, 100,   5,   7,  11,  12, 100,  20])
```

In [37]:
```
1 myarr[[0,-1]] = [50]
```

In [38]:
```
1 myarr
```

Out[38]:
```
array([ 50,   2,   3, 100,   5,   7,  11,  12, 100,  50])
```

In [39]:
```
np_arr = np.array(["apple",'orange','lemon'])
np_arr = np.append(np_arr,['bnana','grapes'])
```

In [40]:
```
np_arr
```

Out[40]:
```
array(['apple', 'orange', 'lemon', 'bnana', 'grapes'], dtype='<U6')
```

In [41]:
```
np_arr = np.delete(np_arr, [0,1])
```

In [42]:
```
np_arr
```

Out[42]:
```
array(['lemon', 'bnana', 'grapes'], dtype='<U6')
```

Adding elements to 2d Arrays

In [43]:
```
np_arr = np.array([[1,2,3,4],[-4,-3,-2,-1]])
```

In [44]:
```
np_arr
```

Out[44]:
```
array([[ 1,  2,  3,  4],
       [-4, -3, -2, -1]])
```

In [45]:
```
np_arr.shape
```

Out[45]:
```
(2, 4)
```

In [56]:
```
np_arr = np.append(np_arr, 0) # by default works on 1-Dimensional array
```

In [47]:
```
np_arr
```

Out[47]:
```
array([ 1,  2,  3,  4, -4, -3, -2, -1,  0])
```

In [49]:
```
np_arr.shape
```

Out[49]:
```
(9,)
```

In [50]:
```
myarr
```

Out[50]:
```
array([ 50,   2,   3, 100,   5,   7,  11,  12, 100,  50])
```

In [51]:
```
1 myarr[-1]
```

Out[51]:
```
50
```

In [52]:
```
1 myarr[-1,0]
```

```
---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
Cell In[52], line 1
----> 1 myarr[-1,0]

IndexError: too many indices for array: array is 1-dimensional, but 2 were i
ndexed
```

In [53]:
```
1 myarr[[-1,0]]
```

Out[53]:
```
array([50, 50])
```

In [54]:
```
1 myarr
```

Out[54]:
```
array([ 50,   2,   3, 100,   5,   7,  11,  12, 100,  50])
```

In [55]:
```
1 myarr[[0,4,8,-1]]
```

Out[55]:
```
array([ 50,   5, 100,  50])
```

In [57]:
```
1 arr = np.array([-2,4,5,7])
```

In [59]:
```
1 np.append(np_arr, arr,axis = 0)
```

Out[59]:
```
array([ 1,  2,  3,  4, -4, -3, -2, -1,  0,  0, -2,  4,  5,  7])
```

Delete elements from 2d

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

In [66]:
```
1 arr
```

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

In [67]:
```
1 np.delete(arr,0, axis = None) # here by default axis = None
```

Out[67]:
```
array([2, 3, 4, 5, 6, 7, 8, 9])
```

In [64]:
```
1 arr
```

Out[64]:
```
array([[1, 2, 3],
       [4, 5, 6],
       [7, 8, 9]])
```

In [65]:
```
1 np.delete(arr, 0, axis = 0) # axis = 0 -> for row
```

Out[65]:
```
array([[4, 5, 6],
       [7, 8, 9]])
```

In [68]:
```
1 arr
```

Out[68]:
```
array([[1, 2, 3],
       [4, 5, 6],
       [7, 8, 9]])
```

In [69]:
```
1 np.delete(arr, 0, axis = 1) # axis = 1 -> for col
```

Out[69]:
```
array([[2, 3],
       [5, 6],
       [8, 9]])
```

How can I detete 1st row and last row

In [70]:
```
1 arr
```

Out[70]:
```
array([[1, 2, 3],
       [4, 5, 6],
       [7, 8, 9]])
```

In [71]:
```
1 np.delete(arr, [0,2], axis=0)
2
```

Out[71]:
```
array([[4, 5, 6]])
```

Replacing elements in 2d array

In [73]:
```
arr = np.array([["Abhinav","Amrit","Grace","Yatin"],
                ['E101','E102','E142','E832'], [45,67,89,30]])
```

In [74]:
```
arr
```

Out[74]:
```
array([['Abhinav', 'Amrit', 'Grace', 'Yatin'],
       ['E101', 'E102', 'E142', 'E832'],
       ['45', '67', '89', '30']], dtype='<U11')
```

In [75]:
```
arr[0]
```

Out[75]:
```
array(['Abhinav', 'Amrit', 'Grace', 'Yatin'], dtype='<U11')
```

In [77]:
```
arr[0] = ["Abhishek","Ram","Vivek","Firdos"]
```

In [78]:
```
arr
```

Out[78]:
```
array([['Abhishek', 'Ram', 'Vivek', 'Firdos'],
       ['E101', 'E102', 'E142', 'E832'],
       ['45', '67', '89', '30']], dtype='<U11')
```

In [80]:
```
arr = np.array([[["Abhinav","Amrit","Grace","Yatin"],
                ['E101','E102','E142','E832'], [45,67,89,30]]])
```

In [81]:
```
arr
```

Out[81]:
```
array([[['Abhinav', 'Amrit', 'Grace', 'Yatin'],
        ['E101', 'E102', 'E142', 'E832'],
        ['45', '67', '89', '30']]], dtype='<U11')
```

In [82]:
```
arr[0]
```

Out[82]:
```
array([['Abhinav', 'Amrit', 'Grace', 'Yatin'],
       ['E101', 'E102', 'E142', 'E832'],
       ['45', '67', '89', '30']], dtype='<U11')
```

In [83]:
```
arr[0][0]
```

Out[83]:
```
array(['Abhinav', 'Amrit', 'Grace', 'Yatin'], dtype='<U11')
```

In [84]:
```
arr[0][-1]
```

Out[84]:
```
array(['45', '67', '89', '30'], dtype='<U11')
```

In [85]:
```
arr[0][-2]
```

Out[85]:
```
array(['E101', 'E102', 'E142', 'E832'], dtype='<U11')
```

In [86]:
```
arr
```

Out[86]:
```
array([[['Abhinav', 'Amrit', 'Grace', 'Yatin'],
        ['E101', 'E102', 'E142', 'E832'],
        ['45', '67', '89', '30']]], dtype='<U11')
```

In [87]:
```
arr[0][0][-1]
```

Out[87]:
```
'Yatin'
```

In [89]:
```
arr[0][1][3]
```

Out[89]:
```
'E832'
```

In [90]:
```
arr[0][0][-1] = "Corey"
arr[0][1][3] = "Emp145"
```

In [91]:
```
arr
```

Out[91]:
```
array([[['Abhinav', 'Amrit', 'Grace', 'Corey'],
        ['E101', 'E102', 'E142', 'Emp145'],
        ['45', '67', '89', '30']]], dtype='<U11')
```

In [92]:
```
arr
```

Out[92]:
```
array([[['Abhinav', 'Amrit', 'Grace', 'Corey'],
        ['E101', 'E102', 'E142', 'Emp145'],
        ['45', '67', '89', '30']]], dtype='<U11')
```

how to print a particular column

In [112]:
```
arr
```

Out[112]:
```
array([[['Abhinav', 'Amrit', 'Grace', 'Corey'],
        ['E101', 'E102', 'E142', 'Emp145'],
        ['45', '67', '89', '30']]], dtype='<U11')
```

In [114]:
```
arr.ndim
```

Out[114]:
```
3
```

In [115]:
```
arr[:,-1]
```

Out[115]:
```
array([['45', '67', '89', '30']], dtype='<U11')
```

In [117]:
```
arr[0][:,-1]
```

Out[117]:
```
array(['Corey', 'Emp145', '30'], dtype='<U11')
```

In [118]:
```
arr[0][:,0]
```

Out[118]:
```
array(['Abhinav', 'E101', '45'], dtype='<U11')
```

In [119]:
```
arr[0][:,-1][::-1]
```

Out[119]:
```
array(['30', 'Emp145', 'Corey'], dtype='<U11')
```

In [121]:
```
arr[0][:,-1] = arr[0][:,-1][::-1]
```

In [122]:
```
arr
```

Out[122]:
```
array([[['Abhinav', 'Amrit', 'Grace', '30'],
        ['E101', 'E102', 'E142', 'Emp145'],
        ['45', '67', '89', 'Corey']]], dtype='<U11')
```

In [123]:
```
lst = [[['Abhinav', 'Amrit', 'Grace', 'Yatin'],
        ['E101', 'E102', 'E142', 'E832'],
        ['45', '67', '89', '30']]]
```

In [124]:
```
type(lst)
```

Out[124]:
```
list
```

In [125]:
```
len(lst)
```

Out[125]:
```
1
```

In [126]:
```
lst[0]
```

Out[126]:
```
[['Abhinav', 'Amrit', 'Grace', 'Yatin'],
 ['E101', 'E102', 'E142', 'E832'],
 ['45', '67', '89', '30']]
```

In [127]:
```
lst[0]
```

Out[127]:
```
[['Abhinav', 'Amrit', 'Grace', 'Yatin'],
 ['E101', 'E102', 'E142', 'E832'],
 ['45', '67', '89', '30']]
```

In [128]:
```
len(lst[0])
```

Out[128]:
```
3
```

In [129]:
```
lst[0][0]
```

Out[129]:
```
['Abhinav', 'Amrit', 'Grace', 'Yatin']
```

In [130]:
```
lst[0][0] = lst[0][0][::-1]
```

In [131]:
```
lst
```

Out[131]:
```
[['Yatin', 'Grace', 'Amrit', 'Abhinav'],
  ['E101', 'E102', 'E142', 'E832'],
  ['45', '67', '89', '30']]]
```

In [134]:
```
lst[0][0].sort()
```

In [135]:
```
lst
```

Out[135]:
```
[['Abhinav', 'Amrit', 'Grace', 'Yatin'],
  ['E101', 'E102', 'E142', 'E832'],
  ['45', '67', '89', '30']]]
```

In [136]:
```
array = np.array(lst)
```

In [137]:
```
array
```

Out[137]:
```
array([[['Abhinav', 'Amrit', 'Grace', 'Yatin'],
        ['E101', 'E102', 'E142', 'E832'],
        ['45', '67', '89', '30']]], dtype='<U7')
```

In [140]:
```
array[0][0] = array[0][0][::-1]
```

In [141]:
```
array
```

Out[141]:
```
array([[['Yatin', 'Grace', 'Amrit', 'Abhinav'],
        ['E101', 'E102', 'E142', 'E832'],
        ['45', '67', '89', '30']]], dtype='<U7')
```

How to concatenate an array?

In [145]:
```
1 arr_1 = np.array([[1,2],[2,4],[-3,-5]])
```

In [146]:
```
1 arr_1
```

Out[146]:
```
array([[ 1,  2],
       [ 2,  4],
       [-3, -5]])
```

In [147]:
```
1 arr_2 = np.array([[1,2],[3,4],[-3,-7]])
```

In [148]:
```
1 arr_2
```

Out[148]:
```
array([[ 1,  2],
       [ 3,  4],
       [-3, -7]])
```

In [154]:
```
1 # concatenate
2 np.concatenate((arr_1,arr_2), axis = 0)  # by row (default)
```

Out[154]:
```
array([[ 1,  2],
       [ 2,  4],
       [-3, -5],
       [ 1,  2],
       [ 3,  4],
       [-3, -7]])
```

In [150]:
```
1 # concatenate
2 np.concatenate((arr_1,arr_2), axis = 1)  # by col
```

Out[150]:
```
array([[ 1,  2,  1,  2],
       [ 2,  4,  3,  4],
       [-3, -5, -3, -7]])
```

In [152]:
```
1 np.concatenate((arr_1,arr_2), axis = None)
```

Out[152]:
```
array([ 1,  2,  2,  4, -3, -5,  1,  2,  3,  4, -3, -7])
```

In [153]:
```
1 np.concatenate?
```

In [155]:
```
1 arr = np.concatenate((arr_1,arr_2), axis = None)
```

In [156]:
```
arr
```

Out[156]:
```
array([ 1,  2,  2,  4, -3, -5,  1,  2,  3,  4, -3, -7])
```

How we can sort an element

In [157]:
```
np.argsort(arr) #argsort function returns an ouput in the form item's ind
```

Out[157]:
```
array([11,  5,  4, 10,  0,  6,  1,  2,  7,  8,  3,  9], dtype=int64)
```

In [158]:
```
arr[np.argsort(arr)] # asc
```

Out[158]:
```
array([-7, -5, -3, -3,  1,  1,  2,  2,  2,  3,  4,  4])
```

In [163]:
```
arr[np.argsort(arr)[::-1]]
```

Out[163]:
```
array([ 4,  4,  3,  2,  2,  2,  1,  1, -3, -3, -5, -7])
```

In [164]:
```
np.argsort(arr)[::-1]
```

Out[164]:
```
array([ 9,  3,  8,  7,  2,  1,  6,  0, 10,  4,  5, 11], dtype=int64)
```

Arithmetic Operations

In [165]:
```
arr1 = np.array([10,20,30,40,55])
arr2 = np.array([5,70,20,35,75])
```

In [166]:
```
arr1 + arr2
```

Out[166]:
```
array([ 15,  90,  50,  75, 130])
```

In [167]:
```
lst1 = [1,2,3,4]
lst2 = [4,5,6,7]
```

In [168]:
```
lst1 + lst2
```

Out[168]:
```
[1, 2, 3, 4, 4, 5, 6, 7]
```

In [169]:
```
np.array(lst1) + np.array(lst2)
```

Out[169]:
```
array([ 5,  7,  9, 11])
```

In [176]:
```
for i in range(len(lst1)):
    print(lst1[i]+lst2[i])
```

```
5
7
9
11
```

In [177]:
```
[lst1[i] + lst2[i] for i in range(len(lst1))]
```

Out[177]: [5, 7, 9, 11]

In [178]:
```
lst1
```

Out[178]: [1, 2, 3, 4]

In [179]:
```
lst2
```

Out[179]: [4, 5, 6, 7]

In [180]:
```
arr1
```

Out[180]: array([10, 20, 30, 40, 55])

In [181]:
```
arr2
```

Out[181]: array([5, 70, 20, 35, 75])

In [182]:
```
arr1 - arr2
```

Out[182]: array([5, -50, 10,   5, -20])

In [183]:
```
arr1 * 5
```

Out[183]: array([50, 100, 150, 200, 275])

In [184]:
```
arr1 = np.array([[5,6],[7,8]])
arr2 = np.array([[2,8],[-4,-3]])
```

In [185]:
```
arr1
```

Out[185]:
```
array([[5, 6],
       [7, 8]])
```

In [186]:
```
arr2
```

Out[186]:
```
array([[ 2,  8],
       [-4, -3]])
```

In [187]:
```
arr1 + arr2
```

Out[187]:
```
array([[ 7, 14],
       [ 3,  5]])
```

In [189]:
```
arr1 - arr2
```

Out[189]:
```
array([[ 3, -2],
       [11, 11]])
```

In [190]:
```
arr1
```

Out[190]:
```
array([[5, 6],
       [7, 8]])
```

In [192]:
```
arr1 + 2
```

Out[192]:
```
array([[ 7,  8],
       [ 9, 10]])
```

In [193]:
```
arr1 * arr2
```

Out[193]:
```
array([[ 10,  48],
       [-28, -24]])
```

In [194]:
```
arr2 * 2
```

Out[194]:
```
array([[ 4, 16],
       [-8, -6]])
```

In [195]:
```
np_arr = np.array([[4,5,6],[5,8,9],[4,8,1]])
```

In [196]:
```
np_arr
```

Out[196]:
```
array([[4, 5, 6],
       [5, 8, 9],
       [4, 8, 1]])
```

In [198]:
```
np_arr.sum()
```

Out[198]:
```
50
```

In [199]:
```
1 np_arr.mean()
```

Out[199]:
```
5.555555555555555
```

In [202]:
```
1 np_arr.max()
```

Out[202]:
```
9
```

In [203]:
```
1 np_arr.min()
```

Out[203]:
```
1
```

In [204]:
```
1 np_arr.std()
```

Out[204]:
```
2.3622546250521443
```

In [205]:
```
1 np_arr
```

Out[205]:
```
array([[4, 5, 6],
       [5, 8, 9],
       [4, 8, 1]])
```

In [207]:
```
1 np_arr.prod()
```

Out[207]:
```
1382400
```

In [209]:
```
1 newarr = np.array([[1,2],[3,4]])
```

In [210]:
```
1 newarr.prod()
```

Out[210]:
```
24
```

In [211]:
```
1 np_arr
```

Out[211]:
```
array([[4, 5, 6],
       [5, 8, 9],
       [4, 8, 1]])
```

In [212]:
```
1 np_arr.prod(axis = 0)
```

Out[212]:
```
array([ 80, 320,  54])
```

In [213]:
```
1 np_arr.prod(axis = 1)
```

Out[213]:
```
array([120, 360,  32])
```

In [214]:
```
1 np_arr.mean()
```

Out[214]:
```
5.555555555555555
```

In [218]:
```
1 np_arr
```

Out[218]:
```
array([[4, 5, 6],
       [5, 8, 9],
       [4, 8, 1]])
```

In [215]:
```
1 np_arr.mean(axis = 0) # col
```

Out[215]:
```
array([4.33333333, 7.        , 5.33333333])
```

In [216]:
```
1 np_arr.mean(axis = 1) # row
```

Out[216]:
```
array([5.        , 7.33333333, 4.33333333])
```

In [217]:
```
1 np.sqrt(25)
```

Out[217]:
```
5.0
```

Random Number Generation

In [219]:
```
1 np.random.randint(5,100,10)
```

Out[219]:
```
array([16, 67, 38, 25, 35, 48, 83, 58, 98, 18])
```

In [223]:
```
1 np.random.seed(5)
2 np.random.randint(5,100,10)
```

Out[223]:
```
array([83, 66, 21, 78, 13, 67, 32, 35, 85, 12])
```

In [224]:
```
1 np.random.seed(6)
2 np.random.randint(5,100,10)
```

Out[224]:
```
array([15, 78, 89, 84, 85, 67, 30,  6, 80, 82])
```

In [225]:
```
1 np.random.seed(1)
2 np.random.randint(5,100,10)
```

Out[225]:
```
array([42, 17, 77, 14, 80, 10, 84, 69, 21,  6])
```

In [226]:
```
np.random.seed(100)
np.random.randint(5,100,10)
```

Out[226]:
```
array([13, 29, 72, 92, 84, 53, 15, 99, 57, 58])
```

In [227]:
```
np.random.seed(6)
np.random.randint(5,100,10)
```

Out[227]:
```
array([15, 78, 89, 84, 85, 67, 30,  6, 80, 82])
```

In [232]:
```
np.random.randint(low = 8888888888, high = 9999999999, size = 50,
                  dtype='int64')
```

Out[232]:
```
array([9988673719, 9720688573, 9500181774, 9718348847, 9952374105,
       9757625950, 9099080665, 9793311741, 9911652321, 9563093780,
       9877031818, 9559023299, 9135562124, 9143734944, 8922909389,
       9555213947, 9871357086, 9095926268, 9736954445, 9972412222,
       9083316001, 8898401749, 8902539855, 9516354456, 9720612972,
       9939398772, 9926671962, 8969492071, 9476798670, 9620277560,
       9414542365, 8950334645, 9306060757, 9842662690, 9406639602,
       9510471120, 9778329342, 9939852185, 9941394086, 9376409058,
       9813696996, 9306163114, 9153725759, 8922796586, 9454830527,
       9498223768, 9510209718, 9663062263, 9224555456, 8996847163],
      dtype=int64)
```

In [235]:
```
np.random.seed(50)
np.random.randint(low = 8888888888, high = 9999999999, size = 50,
                  dtype='int64')
```

Out[235]:
```
array([9443689496, 9868498371, 9208333989, 9986141017, 9361069590,
       8980956798, 9542585982, 9937849138, 9467800980, 9295722194,
       9513664274, 9269476707, 9146402701, 9915574946, 8960235384,
       9086332979, 9638674494, 9624903303, 8919744470, 8992412635,
       9376065652, 9195433899, 9605254522, 9611893252, 8967270936,
       9113338235, 9131132070, 9661242936, 9012243093, 9717482646,
       9115411139, 9808410070, 9888274559, 9752972907, 8964014800,
       9637975493, 9675751679, 9532772636, 9773916856, 9313082098,
       9842540792, 9882895909, 9672106955, 9706442069, 9559441379,
       9858478936, 9425135719, 9830993556, 9744028025, 8977838988],
      dtype=int64)
```

In [236]:
```
np.random.seed?
```

Linspace

This function will include your start and end values.

In [237]:
```
np.linspace(start = 0, stop = 5, num = 10)
```

Out[237]:
```
array([0.        , 0.55555556, 1.11111111, 1.66666667, 2.22222222,
       2.77777778, 3.33333333, 3.88888889, 4.44444444, 5.        ])
```

In [238]:
```
np.linspace(start = 0, stop = 5, num = 3)
```

Out[238]:
```
array([0. , 2.5, 5. ])
```

In [239]:
```
np.linspace(start = 0, stop = 5, num = 2)
```

Out[239]:
```
array([0., 5.])
```

In [240]:
```
np.linspace(start = 0, stop = 5, num = 1)
```

Out[240]:
```
array([0.])
```

In [241]:
```
np.linspace(start = 10, stop = 70, num = 10)
```

Out[241]:
```
array([10.        , 16.66666667, 23.33333333, 30.        , 36.66666667,
       43.33333333, 50.        , 56.66666667, 63.33333333, 70.        ])
```

In [243]:
```
np.linspace(start = 10, stop = 70, num = 10, dtype = 'int')
```

Out[243]:
```
array([10, 16, 23, 30, 36, 43, 50, 56, 63, 70])
```

In [244]:
```
arr_1
```

Out[244]:
```
array([[ 1,  2],
       [ 2,  4],
       [-3, -5]])
```

argmax or argmin

In [245]:
```
np.argmax(arr_1)
```

Out[245]:
```
3
```

In [246]:
```
np.argmin(arr_1)
```

Out[246]:
```
5
```

In [258]:
```
arr = np.random.randint(5,50,25).reshape(5,5)
```

In [259]:
```
1 arr
```

Out[259]:
```
array([[35, 38, 19, 31, 27],
       [42, 11, 35, 46, 41],
       [36, 14,  5, 21, 31],
       [ 9, 48,  6, 42, 18],
       [40, 19,  7, 45, 27]])
```

In [260]:
```
1 np.argmax(arr)
```

Out[260]:
```
16
```

In [261]:
```
1 arr.max()
```

Out[261]:
```
48
```

In [262]:
```
1 arr.min()
```

Out[262]:
```
5
```

In [263]:
```
1 np.argmin(arr)
```

Out[263]:
```
12
```

In [264]:
```
1 arr
```

Out[264]:
```
array([[35, 38, 19, 31, 27],
       [42, 11, 35, 46, 41],
       [36, 14,  5, 21, 31],
       [ 9, 48,  6, 42, 18],
       [40, 19,  7, 45, 27]])
```

In [265]:
```
1 arr[0]
```

Out[265]:
```
array([35, 38, 19, 31, 27])
```

In [266]:
```
1 arr[0].max()
```

Out[266]:
```
38
```

In [267]:
```
1 np.argmin(arr[0])
```

Out[267]:
```
2
```

In [268]:
```
1 np.argmax(arr[0])
```

Out[268]:
```
1
```

In [269]:
```
1 arr
```

Out[269]:
```
array([[35, 38, 19, 31, 27],
       [42, 11, 35, 46, 41],
       [36, 14,  5, 21, 31],
       [ 9, 48,  6, 42, 18],
       [40, 19,  7, 45, 27]])
```

In [270]:
```
1 arr.max(axis = 0) # Each column wise Maximum value
```

Out[270]:
```
array([42, 48, 35, 46, 41])
```

In [271]:
```
1 arr.min(axis = 0) # Each col wise minimum value
```

Out[271]:
```
array([ 9, 11,  5, 21, 18])
```

In [272]:
```
1 arr.max(axis = 1) # Each row wise Maximum value
```

Out[272]:
```
array([38, 46, 36, 48, 45])
```

In [273]:
```
1 arr.min(axis = 1)
```

Out[273]:
```
array([19, 11,  5,  6,  7])
```

In [274]:
```
1 np.pi
```

Out[274]:
```
3.141592653589793
```

In [275]:
```
1 np.e
```

Out[275]:
```
2.718281828459045
```

In [276]:
```
1 np.nan
```

Out[276]:
```
nan
```

In [277]:
```
1 np.nan + 5
```

Out[277]:
```
nan
```

In [278]:
```
1 np.nan * 2
```

Out[278]:
```
nan
```

In [279]:
```
1 np.nan + np.nan
```

Out[279]:
```
nan
```

In [280]:
```
1 arr
```

Out[280]:
```
array([[35, 38, 19, 31, 27],
       [42, 11, 35, 46, 41],
       [36, 14,  5, 21, 31],
       [ 9, 48,  6, 42, 18],
       [40, 19,  7, 45, 27]])
```

In [281]:
```
1 arr + np.nan
```

Out[281]:
```
array([[nan, nan, nan, nan, nan],
       [nan, nan, nan, nan, nan],
       [nan, nan, nan, nan, nan],
       [nan, nan, nan, nan, nan],
       [nan, nan, nan, nan, nan]])
```

Some Other Functions

In [282]:
```
1 np.arange(50)
```

Out[282]:
```
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, 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])
```

In [283]:
```
1 np.arange(100)
```

Out[283]:
```
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, 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])
```

In [287]:
```
1 np.arange(5000)
```

Out[287]:
```
array([   0,    1,    2, ..., 4997, 4998, 4999])
```

In [288]:
```
import sys
np.set_printoptions(threshold=sys.maxsize)
```

In [290]:
```
# np.arange(5000)
```

In [291]:
```
np.set_printoptions(threshold=3)
```

In [292]:
```
np.arange(5000)
```

Out[292]:
```
array([   0,    1,    2, ..., 4997, 4998, 4999])
```

In [293]:
```
arr
```

Out[293]:
```
array([[35, 38, 19, 31, 27],
       [42, 11, 35, 46, 41],
       [36, 14,  5, 21, 31],
       [ 9, 48,  6, 42, 18],
       [40, 19,  7, 45, 27]])
```

In [294]:
```
np.sum(arr, axis = 0)
```

Out[294]:
```
array([162, 130,  72, 185, 144])
```

In [295]:
```
np.sum(arr, axis = 1)
```

Out[295]:
```
array([150, 175, 107, 123, 138])
```

In [296]:
```
np.sum(arr, axis = None)
```

Out[296]:
```
693
```

In [297]:
```
arr1 = np.array([4,5,8,9])
arr2 = np.array([2,3,3,5])
```

In [298]:
```
np.remainder(arr1, arr2)
```

Out[298]:
```
array([0, 2, 2, 4])
```

In [299]:
```
np.floor_divide(arr1, arr2)
```

Out[299]:
```
array([2, 1, 2, 1])
```

In [300]:
```
np.divmod(arr1, arr2)
```

Out[300]:
```
(array([2, 1, 2, 1]), array([0, 2, 2, 4]))
```

In [301]:
```
a = np.array([1.24,4.16,4.965])
```

In [302]:
```
np.round(a)
```

Out[302]:
```
array([1., 4., 5.])
```

In [303]:
```
np.round_(a)
```

Out[303]:
```
array([1., 4., 5.])
```

In [304]:
```
np.floor(a)
```

Out[304]:
```
array([1., 4., 4.])
```

In [305]:
```
np.arange(10)
```

Out[305]:
```
array([0, 1, 2, ..., 7, 8, 9])
```

In [307]:
```
arr = np.arange(10, dtype='float')
```

In [308]:
```
arr
```

Out[308]:
```
array([0., 1., 2., ..., 7., 8., 9.])
```

In [309]:
```
np.array(arr, dtype='int')
```

Out[309]:
```
array([0, 1, 2, ..., 7, 8, 9])
```

In [310]:
```
np.array(arr, dtype='str')
```

Out[310]:
```
array(['0.0', '1.0', '2.0', ..., '7.0', '8.0', '9.0'], dtype='<U32')
```

In [312]:
```
np.array(np.array(arr, dtype='int'), dtype='str')
```

Out[312]:
```
array(['0', '1', '2', ..., '7', '8', '9'], dtype='<U11')
```

In [313]:
```
arr1
```

Out[313]:
```
array([4, 5, 8, 9])
```

In [314]:
```
arr2
```

Out[314]:
```
array([2, 3, 3, 5])
```

In [315]:
```
np.maximum(arr1, arr2)
```

Out[315]:
```
array([4, 5, 8, 9])
```

In [316]:
```
np.minimum(arr1, arr2)
```

Out[316]:
```
array([2, 3, 3, 5])
```

In [318]:
```
np.setdiff1d(arr1,arr2) # 4, 8, 9 are not presented in arr 2
```

Out[318]:
```
array([4, 8, 9])
```

In [319]:
```
set1 = {1,4,5,7,8,9}
set2 = {4,7,0,3,1}
```

In [320]:
```
set1 - set2
```

Out[320]:
```
{5, 8, 9}
```

In [321]:
```
set2 - set1
```

Out[321]:
```
{0, 3}
```

In [322]:
```
np.setdiff1d(arr2,arr1)
```

Out[322]:
```
array([2, 3])
```

In [323]:
```
arr1
```

Out[323]:
```
array([4, 5, 8, 9])
```

NumPy Dot method

In [324]:
```
np.dot(4,8)
```

Out[324]:
```
32
```

In [325]:
```
np.dot([2,3],[4,5])
# 2*4 = 8
# 3*5 = 15
# +___________
#          23
```

Out[325]: 23

In [326]:
```
np.dot(2,[4,5])
```

Out[326]: array([8, 10])

NumPy:

NumPy (Numerical Python) is a fundamental package in the Python scientific computing ecosystem. It provides powerful tools for working with arrays, mathematical functions, linear algebra, random number generation, and more. The importance of NumPy lies in several key areas:

- Efficient array operations
- Numerical computing
- Memory efficiency
- Integration with other libraries
- Vectorized operations
- High-performance computing

Overall, NumPy plays a crucial role in scientific computing, data analysis, and numerical programming in Python. It provides a solid foundation for efficient numerical operations, allowing developers to write concise, readable, and performant code.

NumPy Complete Series:

https://www.youtube.com/playlist?list=PLWuFHho1zKhUGq1be3zUT3Ek5myJrKiIP (https://www.youtube.com/playlist?list=PLWuFHho1zKhUGq1be3zUT3Ek5myJrKiIP)

In [ ]:
```
```