Numpy

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

In [2]: l = [1,2,3,4]

In [3]: ar = np.array(l)

In [4]: ar
Out[4]: array([1, 2, 3, 4])

In [5]: type(ar)
Out[5]: numpy.ndarray

In [6]: np.array([[1,2], [3,4]])
Out[6]: array([[1, 2],
               [3, 4]])

In [7]: np.asarray(l)
Out[7]: array([1, 2, 3, 4])

In [8]: a = [2,3,4]

In [9]: np.asanyarray(a)
Out[9]: array([2, 3, 4])

In [10]: b = np.matrix(l)

In [11]: b
Out[11]: matrix([[1, 2, 3, 4]])

In [12]: np.asanyarray(b)
Out[12]: matrix([[1, 2, 3, 4]])

In [13]: a = np.array(l)

In [14]: a
Out[14]: array([1, 2, 3, 4])

In [15]: c = a

In [16]: c
Out[16]: array([1, 2, 3, 4])

In [17]: a
Out[17]: array([1, 2, 3, 4])

In [18]: c[0]
Out[18]: 1

In [19]: c[0] =100

In [20]: c
Out[20]: array([100,   2,   3,   4])

In [21]: a
Out[21]: array([100,   2,   3,   4])

In [22]: d = np.copy(a)

In [23]: d
Out[23]: array([100,   2,   3,   4])

In [24]: a
Out[24]: array([100,   2,   3,   4])

In [25]: a[1] = 400

In [26]: a
Out[26]: array([100, 400,   3,   4])

In [27]: d
Out[27]: array([100,   2,   3,   4])

In [28]: np.fromfunction(lambda i,j : i==j , (3,3))
Out[28]: array([[ True, False, False],
               [False,  True, False],
               [False, False,  True]])

In [29]: np.fromfunction(lambda i,j : i*j , (3,3))
Out[29]: array([[0., 0., 0.],
               [0., 1., 2.],
               [0., 2., 4.]])

In [30]: iterable = (i*i for i in range(5))

In [31]: np.fromiter(iterable, float)
Out[31]: array([ 0.,  1.,  4.,  9., 16.])

In [32]: np.fromstring('234 234' , sep= ' ')
Out[32]: array([234., 234.])

In [33]: np.fromstring('5,7' , sep= ',')
Out[33]: array([5., 7.])
```

Numpy - Data Types

```
In [34]: l = [2,3,4,5,6]

In [35]: ar = np.array(l)

In [36]: ar
Out[36]: array([2, 3, 4, 5, 6])

In [37]: ar.ndim
Out[37]: 1

In [38]: ar2 = np.array([[1,2,3,4],[2,3,4,5]])

In [39]: ar2
Out[39]: array([[1, 2, 3, 4],
                [2, 3, 4, 5]])

In [40]: ar2.ndim
Out[40]: 2

In [41]: ar.size
Out[41]: 5

In [42]: ar2.size
Out[42]: 8

In [43]: ar
Out[43]: array([2, 3, 4, 5, 6])

In [44]: ar2
Out[44]: array([[1, 2, 3, 4],
                [2, 3, 4, 5]])

In [45]: ar.shape
Out[45]: (5,)

In [46]: ar2.shape
Out[46]: (2, 4)

In [47]: ar.dtype
Out[47]: dtype('int64')

In [48]: ar2.dtype
Out[48]: dtype('int64')

In [49]: ar22 = np.array([(1.4,45,45), (23,45,66)])

In [50]: ar22
Out[50]: array([[ 1.4, 45. , 45. ],
                [23. , 45. , 66. ]])

In [51]: ar22.dtype
Out[51]: dtype('float64')

In [52]: list(range(5))
Out[52]: [0, 1, 2, 3, 4]

In [54]: list(range(1.5,5))

         ---------------------------------------------------------------------------
         TypeError                                 Traceback (most recent call last)
         Cell In[54], line 1
         ----> 1 list(range(1.5,5))

         TypeError: 'float' object cannot be interpreted as an integer

In [55]: np.arange(2.1,5)
Out[55]: array([2.1, 3.1, 4.1])

In [56]: np.arange(2.1,7.4)
Out[56]: array([2.1, 3.1, 4.1, 5.1, 6.1, 7.1])

In [57]: np.arange(2.1,5.7,.4)
Out[57]: array([2.1, 2.5, 2.9, 3.3, 3.7, 4.1, 4.5, 4.9, 5.3])

In [58]: list(np.arange(2.1,5.7,.4))
Out[58]: [2.1,
          2.5,
          2.9,
          3.3,
          3.6999999999999997,
          4.1,
          4.5,
          4.8999999999999995,
          5.299999999999999]

In [59]: np.linspace(1,5,10)
Out[59]: array([1.        , 1.44444444, 1.88888889, 2.33333333, 2.77777778,
                3.22222222, 3.66666667, 4.11111111, 4.55555556, 5.        ])

In [60]: np.zeros(5)
Out[60]: array([0., 0., 0., 0., 0.])

In [61]: np.zeros((3,4))
Out[61]: array([[0., 0., 0., 0.],
                [0., 0., 0., 0.],
                [0., 0., 0., 0.]])
```

```
In [62]: np.zeros((3,4,2))

Out[62]: 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.]]])

In [63]: np.zeros((3,4,2,3))

Out[63]: 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.]]],


                [[[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 [64]: np.ones(4)

Out[64]: array([1., 1., 1., 1.])

In [65]: np.ones((2,3))

Out[65]: array([[1., 1., 1.],
                [1., 1., 1.]])

In [66]: np.ones((2,3,2))

Out[66]: array([[[1., 1.],
                 [1., 1.],
                 [1., 1.]],

                [[1., 1.],
                 [1., 1.],
                 [1., 1.]]])

In [67]: on = np.ones((2,3,2))

In [68]: on

Out[68]: array([[[1., 1.],
                 [1., 1.],
                 [1., 1.]],

                [[1., 1.],
                 [1., 1.],
                 [1., 1.]]])

In [69]: on + 5

Out[69]: array([[[6., 6.],
                 [6., 6.],
                 [6., 6.]],

                [[6., 6.],
                 [6., 6.],
                 [6., 6.]]])

In [70]: on * 4

Out[70]: array([[[4., 4.],
                 [4., 4.],
                 [4., 4.]],

                [[4., 4.],
                 [4., 4.],
                 [4., 4.]]])

In [71]: np.empty((3,5))

Out[71]: array([[4.67923045e-310, 0.00000000e+000, 0.00000000e+000,
                 0.00000000e+000, 0.00000000e+000],
                [0.00000000e+000, 0.00000000e+000, 0.00000000e+000,
                 0.00000000e+000, 0.00000000e+000],
                [0.00000000e+000, 0.00000000e+000, 0.00000000e+000,
                 0.00000000e+000, 6.32404027e-322]])

In [72]: np.empty((3,4))

Out[72]: array([[4., 4., 4., 4.],
                [4., 4., 4., 4.],
                [4., 4., 4., 4.]])

In [73]: np.eye(4)

Out[73]: array([[1., 0., 0., 0.],
                [0., 1., 0., 0.],
                [0., 0., 1., 0.],
                [0., 0., 0., 1.]])
```

```
In [74]: np.linspace(2,4 ,20)

Out[74]: array([2.        , 2.10526316, 2.21052632, 2.31578947, 2.42105263,
                2.52631579, 2.63157895, 2.73684211, 2.84210526, 2.94736842,
                3.05263158, 3.15789474, 3.26315789, 3.36842105, 3.47368421,
                3.57894737, 3.68421053, 3.78947368, 3.89473684, 4.        ])

In [75]: np.logspace(2,5, 10)

Out[75]: array([   100.        ,    215.443469  ,    464.15888336,   1000.        ,
                  2154.43469003,   4641.58883361,  10000.        ,  21544.34690032,
                 46415.88833613, 100000.        ])

In [76]: np.logspace(2,5, 10, base = 2)

Out[76]: array([ 4.        ,  5.0396842 ,  6.34960421,  8.        , 10.0793684 ,
                12.69920842, 16.        , 20.1587368 , 25.39841683, 32.        ])

In [77]: np.random.randn(3,4)

Out[77]: array([[ 0.41303987,  1.29299529,  0.85418433, -0.27669442],
                [ 0.31638597,  1.56008085,  0.18180611,  1.61819728],
                [ 1.36232164, -0.37523951,  0.0352562 ,  1.0827592 ]])

In [78]: arr = np.random.randn(3,4)

In [79]: import pandas as pd

In [80]: pd.DataFrame(arr)

Out[80]:
```

	0	1	2	3
0	0.207879	-0.973871	-1.448091	1.137070
1	0.160819	0.969992	-2.365448	-0.869218
2	1.147999	0.347573	0.163600	-1.267925

```
In [81]: np.random.rand(3,4)

Out[81]: array([[0.23849987, 0.4522432 , 0.03958691, 0.0774191 ],
                [0.7653942 , 0.7576803 , 0.26203362, 0.65275911],
                [0.72003273, 0.84306809, 0.34480282, 0.84412332]])

In [82]: np.random.randint(1,110 , (3,4))

Out[82]: array([[ 26,  73,  97,  23],
                [ 10, 107, 101,  66],
                [ 57,  16,   9, 100]])

In [83]: np.random.randint(1,110 , (300,400))

Out[83]: array([[100,  60,  13, ...,  13, 104,  17],
                [102, 103,  17, ...,  76,   3, 106],
                [ 46,  21,  44, ...,  94,  68,  56],
                ...,
                [ 93,  14,  68, ...,  48,  62,  70],
                [ 99, 102,  47, ...,  68,   2,  80],
                [ 74,  72,  14, ..., 100,  52,  11]])

In [84]: pd.DataFrame(np.random.randint(1,110 , (300,400)))

Out[84]:
```

	0	1	2	3	4	5	6	7	8	9	...	390	391	392	393	394	395	396	397	398	399
0	46	99	30	92	100	91	87	75	107	50	...	76	19	23	28	100	63	21	34	82	43
1	11	71	93	49	80	40	13	9	52	58	...	21	6	1	47	101	56	95	76	100	73
2	50	17	55	24	55	90	42	49	1	4	...	70	47	17	76	30	86	104	59	72	25
3	33	34	3	80	25	33	44	13	102	45	...	72	70	42	80	19	35	53	102	91	60
4	60	49	77	45	80	70	97	94	68	55	...	52	11	71	101	56	32	20	99	40	101
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
295	81	50	75	85	17	62	50	95	95	19	...	40	87	99	3	80	94	12	54	20	15
296	7	62	90	103	41	69	26	31	40	14	...	83	57	48	36	66	3	65	85	1	102
297	52	73	90	85	100	53	74	54	92	34	...	30	56	34	34	7	3	21	50	103	30
298	61	62	83	20	53	93	90	23	57	94	...	42	58	105	97	70	86	37	105	28	49
299	34	37	98	53	58	75	102	57	45	47	...	39	86	47	18	11	70	44	18	64	57

300 rows × 400 columns

```
In [85]: pd.DataFrame(np.random.randint(1,110 , (300,400))).to_csv('test.csv')

In [86]: ar = np.random.rand(3,4)

In [87]: arr

Out[87]: array([[ 0.20787867, -0.97387149, -1.44809102,  1.13707014],
                [ 0.16081877,  0.96999184, -2.36544793, -0.86921809],
                [ 1.14799937,  0.3475729 ,  0.16359952, -1.26792514]])

In [88]: arr.reshape(6,3)

         ---------------------------------------------------------------------------
         ValueError                                Traceback (most recent call last)
         Cell In[88], line 1
         ----> 1 arr.reshape(6,3)

         ValueError: cannot reshape array of size 12 into shape (6,3)

In [89]: arr.reshape(6,2)

Out[89]: array([[ 0.20787867, -0.97387149],
                [-1.44809102,  1.13707014],
                [ 0.16081877,  0.96999184],
                [-2.36544793, -0.86921809],
                [ 1.14799937,  0.3475729 ],
                [ 0.16359952, -1.26792514]])

In [90]: arr.reshape(6,-1)

Out[90]: array([[ 0.20787867, -0.97387149],
                [-1.44809102,  1.13707014],
                [ 0.16081877,  0.96999184],
                [-2.36544793, -0.86921809],
                [ 1.14799937,  0.3475729 ],
                [ 0.16359952, -1.26792514]])

In [91]: arr.reshape(6,-1232344243)

Out[91]: array([[ 0.20787867, -0.97387149],
                [-1.44809102,  1.13707014],
                [ 0.16081877,  0.96999184],
                [-2.36544793, -0.86921809],
                [ 1.14799937,  0.3475729 ],
                [ 0.16359952, -1.26792514]])
```

```
In [92]: arr1 = arr.reshape(6,-1232344243)

In [93]: arr1

Out[93]: array([[ 0.20787867, -0.97387149],
               [-1.44809102,  1.13707014],
               [ 0.16081877,  0.96999184],
               [-2.36544793, -0.86921809],
               [ 1.14799937,  0.3475729 ],
               [ 0.16359952, -1.26792514]])

In [ ]:

         arr1[1][1]

In [95]: arr1[2:5,1]

Out[95]: array([ 0.96999184, -0.86921809,  0.3475729 ])

In [96]: np.random.randint(1,100, (5,5))

Out[96]: array([[63, 12, 69,  4, 71],
               [71, 14, 14, 53, 50],
               [92, 21, 65, 77, 53],
               [32, 61, 47, 90, 95],
               [94, 28, 39, 11, 23]])

In [97]: arr = np.random.randint(1,100, (5,5))

In [98]: arr

Out[98]: array([[22, 81, 17, 84, 83],
               [70, 53,  3, 66, 59],
               [73, 69, 61, 32, 64],
               [62, 50, 96, 33, 39],
               [29, 31, 73, 31, 50]])

In [99]: arr>50

Out[99]: array([[False,  True, False,  True,  True],
               [ True,  True, False,  True,  True],
               [ True,  True,  True, False,  True],
               [ True, False,  True, False, False],
               [False, False,  True, False, False]])

In [100]: arr[arr>50]

Out[100]: array([81, 84, 83, 70, 53, 66, 59, 73, 69, 61, 64, 62, 96, 73])

In [101]: arr

Out[101]: array([[22, 81, 17, 84, 83],
                [70, 53,  3, 66, 59],
                [73, 69, 61, 32, 64],
                [62, 50, 96, 33, 39],
                [29, 31, 73, 31, 50]])

In [103]: arr[2:4 ,[1,2]]

Out[103]: array([[69, 61],
                [50, 96]])

In [104]: arr[0][0]

Out[104]: 22

In [105]: arr[0][0] = 5000

In [106]: arr

Out[106]: array([[5000,   81,   17,   84,   83],
                [  70,   53,    3,   66,   59],
                [  73,   69,   61,   32,   64],
                [  62,   50,   96,   33,   39],
                [  29,   31,   73,   31,   50]])

In [107]: arr1 = np.random.randint(1,3 , (3,3))
          arr2 = np.random.randint(1,3 , (3,3))

In [108]: arr1

Out[108]: array([[1, 2, 2],
                [1, 1, 1],
                [2, 1, 2]])

In [109]: arr2

Out[109]: array([[1, 2, 2],
                [2, 2, 1],
                [1, 1, 1]])

In [110]: arr1+arr2

Out[110]: array([[2, 4, 4],
                [3, 3, 2],
                [3, 2, 3]])

In [111]: arr1-arr2

Out[111]: array([[ 0,  0,  0],
                [-1, -1,  0],
                [ 1,  0,  1]])

In [112]: arr1/arr2

Out[112]: array([[1. , 1. , 1. ],
                [0.5, 0.5, 1. ],
                [2. , 1. , 2. ]])

In [113]: arr1 * arr2

Out[113]: array([[1, 4, 4],
                [2, 2, 1],
                [2, 1, 2]])

In [114]: arr1@arr2

Out[114]: array([[7, 8, 6],
                [4, 5, 4],
                [6, 8, 7]])
```

```
In [115]: arr1/0

          /tmp/ipykernel_276/1510032488.py:1: RuntimeWarning: divide by zero encountered in divide
            arr1/0

Out[115]: array([[inf, inf, inf],
                 [inf, inf, inf],
                 [inf, inf, inf]])

In [116]: arr1+100

Out[116]: array([[101, 102, 102],
                 [101, 101, 101],
                 [102, 101, 102]])

In [117]: arr1**2

Out[117]: array([[1, 4, 4],
                 [1, 1, 1],
                 [4, 1, 4]])
```

Numpy - Broadcasting

```
In [119]: np.zeros((4,4))

Out[119]: array([[0., 0., 0., 0.],
                 [0., 0., 0., 0.],
                 [0., 0., 0., 0.],
                 [0., 0., 0., 0.]])

In [120]: arr = np.zeros((4,4))

In [121]: arr

Out[121]: array([[0., 0., 0., 0.],
                 [0., 0., 0., 0.],
                 [0., 0., 0., 0.],
                 [0., 0., 0., 0.]])

In [122]: row = np.array([1,2,3,4])

In [123]: row

Out[123]: array([1, 2, 3, 4])

In [124]: arr + row

Out[124]: array([[1., 2., 3., 4.],
                 [1., 2., 3., 4.],
                 [1., 2., 3., 4.],
                 [1., 2., 3., 4.]])

In [125]: col =  np.array([[1,2,3,4]])

In [126]: col

Out[126]: array([[1, 2, 3, 4]])

In [127]: col.T

Out[127]: array([[1],
                 [2],
                 [3],
                 [4]])

In [128]: col.T + arr

Out[128]: array([[1., 1., 1., 1.],
                 [2., 2., 2., 2.],
                 [3., 3., 3., 3.],
                 [4., 4., 4., 4.]])

In [129]: arr1 = np.random.randint(1,4 , (3,4))

In [130]: arr1

Out[130]: array([[1, 1, 3, 1],
                 [2, 2, 3, 2],
                 [3, 1, 2, 2]])

In [131]: np.sqrt(arr1)

Out[131]: array([[1.        , 1.        , 1.73205081, 1.        ],
                 [1.41421356, 1.41421356, 1.73205081, 1.41421356],
                 [1.73205081, 1.        , 1.41421356, 1.41421356]])

In [132]: np.exp(arr1)

Out[132]: array([[ 2.71828183,  2.71828183, 20.08553692,  2.71828183],
                 [ 7.3890561 ,  7.3890561 , 20.08553692,  7.3890561 ],
                 [20.08553692,  2.71828183,  7.3890561 ,  7.3890561 ]])

In [133]: np.log10(arr1)

Out[133]: array([[0.        , 0.        , 0.47712125, 0.        ],
                 [0.30103   , 0.30103   , 0.47712125, 0.30103   ],
                 [0.47712125, 0.        , 0.30103   , 0.30103   ]])

  In [ ]:
```