Entrée [7]:

```
import numpy as np
```

Entrée [8]:

```
list1 = [0,1,2,3,4]
```

Entrée [10]:

```
list1
```

Out[10]:

```
[0, 1, 2, 3, 4]
```

Entrée [11]:

```
arr1d = np.array(list1)
```

Entrée [12]:

```
arr1d
```

Out[12]:

```
array([0, 1, 2, 3, 4])
```

Entrée [13]:

```
list1.append(5)
```

Entrée [14]:

```
list1
```

Out[14]:

```
[0, 1, 2, 3, 4, 5]
```

Entrée [15]:

```
list1 + 2
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-15-a66ed14c8eee> in <module>
----> 1 list1 + 2

TypeError: can only concatenate list (not "int") to list
```

Entrée [16]:

```
arr1d + 2
```

Out[16]:

```
array([2, 3, 4, 5, 6])
```

Entrée [17]:

```
list2 = [[1, 1, 1], [2, 2 ,2], [3, 3, 3]]
arr2d = np.array(list2)
```

Entrée [19]:

```
type(arr2d)
```

Out[19]:

```
numpy.ndarray
```

Entrée [20]:

```
arr2d.dtype
```

Out[20]:

```
dtype('int32')
```

Entrée [21]:

```
arr2d = np.array(list2, dtype = 'float')
```

Entrée [22]:

```
arr2d
```

Out[22]:

```
array([[1., 1., 1.],
       [2., 2., 2.],
       [3., 3., 3.]])
```

Entrée [25]:

```
arr2d = arr2d.astype('int')
```

Entrée [26]:

```
arr2d.astype('str')
```

Out[26]:

```
array([['1', '1', '1'],
       ['2', '2', '2'],
       ['3', '3', '3']], dtype='<U11')
```

Entrée [30]:

```
list1.append('6')
```

Entrée [31]:

```
list1
```

Out[31]:

```
[0, 1, 2, 3, 4, 5, '6']
```

Entrée [ ]:

```
```

Entrée [32]:

```
np2list = arr2d.tolist()
```

Entrée [33]:

```
np2list
```

Out[33]:

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

Entrée [34]:

```
arr2d.tostring()
```

Out[34]:

```
b'\x01\x00\x00\x00\x01\x00\x00\x00\x01\x00\x00\x00\x02\x00\x00\x00\x02\x00\x
00\x00\x02\x00\x00\x00\x03\x00\x00\x00\x03\x00\x00\x00\x03\x00\x00\x00'
```

Entrée [35]:

```
arr2d.tobytes()
```

Out[35]:

```
b'\x01\x00\x00\x00\x01\x00\x00\x00\x01\x00\x00\x00\x02\x00\x00\x00\x02\x00\x
00\x00\x02\x00\x00\x00\x03\x00\x00\x00\x03\x00\x00\x00\x03\x00\x00\x00'
```

Entrée [ ]:

```
```

dtype and shape

Entrée [36]:

```
list2
```

Out[36]:

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

Entrée [38]:

```
arr2d = arr2d.astype('float' )
```

Entrée [39]:

```
arr2d
```

Out[39]:

```
array([[1., 1., 1.],
       [2., 2., 2.],
       [3., 3., 3.]])
```

Entrée [40]:

```
print('Shape: ', arr2d.shape) # python3
```

```
Shape:  (3, 3)
```

Entrée [41]:

```
arr2d.dtype
```

Out[41]:

```
dtype('float64')
```

Entrée [43]:

```
arr2d.size
```

Out[43]:

```
9
```

Entrée [44]:

```
arr1d.size
```

Out[44]:

```
5
```

Entrée [45]:

```
arr2d.ndim
```

Out[45]:

```
2
```

Entrée [46]:

```
arr1d.ndim
```

Out[46]:

```
1
```

Entrée [ ]:

```

```

Entrée [50]:

```
arr1d = arr1d * arr1d
```

Entrée [51]:

```
arr1d
```

Out[51]:

```
array([ 4,  9,  0,  1, 36], dtype=int32)
```

Entrée [55]:

```
arr1d[1]
```

Out[55]:

```
9
```

Entrée [60]:

```
arr2d[1][0] #[R][C]
```

Out[60]:

```
2.0
```

Entrée [59]:

```
arr2d
```

Out[59]:

```
array([[1., 1., 1.],
       [2., 2., 2.],
       [3., 3., 3.]])
```

Entrée [ ]:

```

```

Entrée [61]:

```
boolarr = arr2d<3
```

Entrée [62]:

```
boolarr
```

Out[62]:

```
array([[ True,  True,  True],
       [ True,  True,  True],
       [False, False, False]])
```

Entrée [63]:

```
arr2d[boolarr]
```

Out[63]:

```
array([1., 1., 1., 2., 2., 2.])
```

Entrée [ ]:

```

```

Entrée [64]:

```
arr2d
```

Out[64]:

```
array([[1., 1., 1.],
       [2., 2., 2.],
       [3., 3., 3.]])
```

Entrée [65]:

```
arr2d[::-1, ]
```

Out[65]:

```
array([[3., 3., 3.],
       [2., 2., 2.],
       [1., 1., 1.]])
```

Entrée [ ]:

```

```

Entrée [66]:

```
arr2d[::-1, ::-1]
```

Out[66]:

```
array([[3., 3., 3.],
       [2., 2., 2.],
       [1., 1., 1.]])
```

np.nan, np.inf

Entrée [67]:

```
np.nan
```

Out[67]:

```
nan
```

Entrée [68]:

```
np.inf
```

Out[68]:

```
inf
```

Entrée [69]:

```
arr2d
```

Out[69]:

```
array([[1., 1., 1.],
       [2., 2., 2.],
       [3., 3., 3.]])
```

Entrée [70]:

```
arr2d[0][0] = np.nan
arr2d[0][1] = np.inf
arr2d
```

Out[70]:

```
array([[nan, inf,  1.],
       [ 2.,  2.,  2.],
       [ 3.,  3.,  3.]])
```

Entrée [71]:

```
np.isnan(arr2d)
```

Out[71]:

```
array([[ True, False, False],
       [False, False, False],
       [False, False, False]])
```

Entrée [72]:

```
np.isinf(arr2d)
```

Out[72]:

```
array([[False,  True, False],
       [False, False, False],
       [False, False, False]])
```

Entrée [73]:

```
missing_flag = np.isnan(arr2d) | np.isinf(arr2d)
missing_flag
```

Out[73]:

```
array([[ True,  True, False],
       [False, False, False],
       [False, False, False]])
```

Entrée [74]:

```
#replace inf and nan with 0
```

Entrée [75]:

```
arr2d[missing_flag]
```

Out[75]:

```
array([nan, inf])
```

Entrée [76]:

```
arr2d[missing_flag] = 0
```

Entrée [77]:

```
arr2d
```

Out[77]:

```
array([[0., 0., 1.],
       [2., 2., 2.],
       [3., 3., 3.]])
```

Entrée [ ]:

```

```

Statistical operations

Entrée [78]:

```
#mean, std, var
```

Entrée [79]:

```
arr2d.mean()
```

Out[79]:

```
1.7777777777777777
```

Entrée [80]:

```
arr2d.max()
```

Out[80]:

```
3.0
```

Entrée [81]:

```
arr2d.min()
```

Out[81]:

```
0.0
```

Entrée [82]:

```
arr2d.std()
```

Out[82]:

```
1.1331154474650633
```

Entrée [83]:

```
arr2d.var()
```

Out[83]:

```
1.2839506172839505
```

Entrée [85]:

```
arr2d.squeeze()
```

Out[85]:

```
array([[0., 0., 1.],
       [2., 2., 2.],
       [3., 3., 3.]])
```

Entrée [86]:

```
arr2d.cumsum()
```

Out[86]:

```
array([ 0.,  0.,  1.,  3.,  5.,  7., 10., 13., 16.])
```

Entrée [ ]:

```

```

Entrée [87]:

```
arr2d
```

Out[87]:

```
array([[0., 0., 1.],
       [2., 2., 2.],
       [3., 3., 3.]])
```

Entrée [ ]:

```

```

Entrée [88]:

```
arr = arr2d[:2, :2]
```

Entrée [89]:

```
arr
```

Out[89]:

```
array([[0., 0.],
       [2., 2.]])
```

Entrée [ ]:

```

```

Entrée [91]:

```
arr2d[1:3, 1:2]
```

Out[91]:

```
array([[2.],
       [3.]])
```

Entrée [ ]:

```

```

Entrée [92]:

```
arr2d
```

Out[92]:

```
array([[0., 0., 1.],
       [2., 2., 2.],
       [3., 3., 3.]])
```

```

```

Entrée [101]:

```
a = arr2d.reshape(1,9)
a
```

Out[101]:

```
array([[0., 0., 1., 2., 2., 2., 3., 3., 3.]])
```

Entrée [100]:

```
a.ndim
```

Out[100]:

```
2
```

Entrée [95]:

```
arr2d.reshape(9,1)
```

Out[95]:

```
array([[0.],
       [0.],
       [1.],
       [2.],
       [2.],
       [2.],
       [3.],
       [3.],
       [3.]])
```

Entrée [ ]:

```

```

Entrée [105]:

```
a = arr2d.flatten()
a #copy
```

Out[105]:

```
array([0., 0., 1., 2., 2., 2., 3., 3., 3.])
```

Entrée [106]:

```
b = arr2d.ravel()
b #reference
```

Out[106]:

```
array([0., 0., 1., 2., 2., 2., 3., 3., 3.])
```

Entrée [107]:

```
arr2d
```

Out[107]:

```
array([[0., 0., 1.],
       [2., 2., 2.],
       [3., 3., 3.]])
```

Entrée [109]:

```
b[0] = -1
```

Entrée [110]:

```
arr2d
```

Out[110]:

```
array([[-1.,  0.,  1.],
       [ 2.,  2.,  2.],
       [ 3.,  3.,  3.]])
```

Entrée [ ]:

```
```

sequence, repetitions, and random numbers

Entrée [116]:

```
np.arange(1, 5, dtype = 'int')
```

Out[116]:

```
array([1, 2, 3, 4])
```

Entrée [118]:

```
np.arange(1, 50, 2)
```

Out[118]:

```
array([ 1,  3,  5,  7,  9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 29, 31, 33,
       35, 37, 39, 41, 43, 45, 47, 49])
```

Entrée [ ]:

```
```

Entrée [123]:

```
np.linspace(1, 50, 9)
```

Out[123]:

```
array([ 1.   ,  7.125, 13.25 , 19.375, 25.5  , 31.625, 37.75 , 43.875,
       50.   ])
```

Entrée [ ]:

```
```

Entrée [124]:

```
np.logspace(1, 50, 10)
```

Out[124]:

```
array([1.00000000e+01, 2.78255940e+06, 7.74263683e+11, 2.15443469e+17,
       5.99484250e+22, 1.66810054e+28, 4.64158883e+33, 1.29154967e+39,
       3.59381366e+44, 1.00000000e+50])
```

Entrée [ ]:

```
```

Entrée [126]:

```
np.zeros([2,2])
```

Out[126]:

```
array([[0., 0.],
       [0., 0.]])
```

Entrée [127]:

```
np.ones([2, 2])
```

Out[127]:

```
array([[1., 1.],
       [1., 1.]])
```

Entrée [ ]:

```

```

Entrée [128]:

```
a = [1, 2, 3]
```

Entrée [129]:

```
a
```

Out[129]:

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

Entrée [130]:

```
np.tile(a, 3)
```

Out[130]:

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

Entrée [131]:

```
np.repeat(a, 3)
```

Out[131]:

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

Entrée [132]:

```
np.repeat(arr2d, 3)
```

Out[132]:

```
array([-1., -1., -1.,  0.,  0.,  0.,  1.,  1.,  1.,  2.,  2.,  2.,  2.,
        2.,  2.,  2.,  2.,  2.,  3.,  3.,  3.,  3.,  3.,  3.,  3.,  3.,
        3.])
```

Entrée [ ]:

```
```

Entrée [133]:

```
arr2d
```

Out[133]:

```
array([[-1.,  0.,  1.],
       [ 2.,  2.,  2.],
       [ 3.,  3.,  3.]])
```

Entrée [ ]:

```
```

Entrée [139]:

```
np.random.rand(3, 3)
```

Out[139]:

```
array([[0.03022941, 0.4897838 , 0.48808364],
       [0.06658789, 0.25569082, 0.30411541],
       [0.65788261, 0.16300932, 0.61885932]])
```

Entrée [138]:

```
np.random.randn(3, 3)
```

Out[138]:

```
array([[-0.95832052, -0.33374875,  0.46566206],
       [-1.04891141,  0.5784687 , -0.1718599 ],
       [ 0.27468945, -0.93154267,  0.65645431]])
```

Entrée [141]:

```
np.random.randint(0, 10, [3,3])
```

Out[141]:

```
array([[6, 4, 0],
       [2, 5, 9],
       [7, 3, 0]])
```

Entrée [ ]:

```
```

Entrée [157]:

```
np.random.seed(1)
np.random.randint(0, 10, [3,3])
```

Out[157]:

```
array([[5, 8, 9],
       [5, 0, 0],
       [1, 7, 6]])
```

Entrée [ ]:

```

```

Entrée [161]:

```
np.unique(arr2d)
```

Out[161]:

```
array([-1.,  0.,  1.,  2.,  3.])
```

Entrée [162]:

```
arr2d
```

Out[162]:

```
array([[-1.,  0.,  1.],
       [ 2.,  2.,  2.],
       [ 3.,  3.,  3.]])
```

Entrée [163]:

```
uniques, counts = np.unique(arr2d, return_counts= True)
```

Entrée [164]:

```
uniques
```

Out[164]:

```
array([-1.,  0.,  1.,  2.,  3.])
```

Entrée [165]:

```
counts
```

Out[165]:

```
array([1, 1, 1, 3, 3], dtype=int64)
```

Entrée [ ]:

```

```

Entrée [ ]:

```

```