

# NumPyExamples

January 24, 2023

## 1 NumPy Examples

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

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

```
[ ]: a
```

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

```
[ ]: a.shape
```

```
[ ]: (4,)
```

```
[ ]: a.ndim
```

```
[ ]: 1
```

```
[ ]: len(a.shape)
```

```
[ ]: 1
```

```
[ ]: a = np.arange(4)
```

```
[ ]: a
```

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

```
[ ]: a + 1
```

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

```
[ ]: np.ndarray
```

```
[ ]: numpy.ndarray
```

```
[ ]: type(a)
```

```
[ ]: numpy.ndarray
```

```
[ ]: type(1)
```

```
[ ]: int
```

```
[ ]: b = np.arange(8).reshape(4, 2, order = 'C')
```

```
[ ]: b
```

```
[ ]: array([[0, 1],  
         [2, 3],  
         [4, 5],  
         [6, 7]])
```

```
[ ]: c = np.arange(8).reshape(4, 2, order = 'F')
```

```
[ ]: c
```

```
[ ]: array([[0, 4],  
         [1, 5],  
         [2, 6],  
         [3, 7]])
```

```
[ ]: c.ravel()
```

```
[ ]: array([0, 4, 1, 5, 2, 6, 3, 7])
```

```
[ ]: c.ravel(order = 'F')
```

```
[ ]: array([0, 1, 2, 3, 4, 5, 6, 7])
```

```
[ ]: d = b.reshape(1, 8)  
d
```

```
[ ]: array([[0, 1, 2, 3, 4, 5, 6, 7]])
```

```
[ ]: d.shape
```

```
[ ]: (1, 8)
```

```
[ ]: d = b.reshape(8, 1)  
d
```

```
[ ]: array([[0],  
         [1],  
         [2],
```

```
[3],  
[4],  
[5],  
[6],  
[7]])
```

```
[ ]: from numpy import newaxis
```

```
[ ]: d = b[:, newaxis]
```

```
[ ]: d
```

```
[ ]: array([[0, 1],  
          [2, 3],  
          [4, 5],  
          [6, 7]])
```

```
[ ]: d.shape
```

```
[ ]: (4, 1, 2)
```

```
[ ]: b
```

```
[ ]: array([[0, 1],  
          [2, 3],  
          [4, 5],  
          [6, 7]])
```

```
[ ]: b = np.arange(6)
```

```
[ ]: b
```

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

```
[ ]: d = b[:, newaxis]  
print(d)  
print(d.shape)
```

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

```
[5]  
(6, 1)
```

```
[ ]: d = b[newaxis, :]  
print(d)  
print(d.shape)
```

```
[[0 1 2 3 4 5]]  
(1, 6)
```

```
[ ]: d = b[newaxis, :4]  
print(d)  
print(d.shape)
```

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

```
[ ]: b
```

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

```
[ ]: c = b.reshape(-1, 3)  
c
```

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

```
[ ]: c.T
```

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

```
[ ]: b.T
```

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

```
[ ]: b.dot(b)
```

```
[ ]: 55
```

```
[ ]: b @ b
```

```
[ ]: 55
```

```
[ ]: np.sqrt
```

```
[ ]: <ufunc 'sqrt'>
```

```
[ ]: np.sqrt(b @ b)
```

```
[ ]: 7.416198487095663
```

```
[ ]: from numpy import linalg as la
```

```
[ ]: la.norm
```

```
[ ]: <function numpy.linalg.norm(x, ord=None, axis=None, keepdims=False)>
```

```
[ ]: la.norm(b)
```

```
[ ]: 7.416198487095663
```

```
[ ]: b = b[:3]  
b
```

```
[ ]: array([0, 1, 2])
```

```
[ ]: b = np.arange(6)  
b
```

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

```
[ ]: c = b[:3]  
c
```

```
[ ]: array([0, 1, 2])
```

```
[ ]: c[-1] = 4  
c
```

```
[ ]: array([0, 1, 4])
```

```
[ ]: b
```

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

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

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

```
[ ]: c
```

```
[ ]: array([-1, 1, 4])
```

```
[ ]: c = np.arange(3)
c
```

```
[ ]: array([0, 1, 2])
```

```
[ ]: c.dot(c)
```

```
[ ]: 5
```

```
[ ]: np.outer(c,c)
```

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

```
[ ]: d
```

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

```
[ ]: d = np.arange(6).reshape(2, -1)
d
```

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

```
[ ]: c
```

```
[ ]: array([0, 1, 2])
```

```
[ ]: d.shape
```

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

```
[ ]: c.shape
```

```
[ ]: (3,)
```

```
[ ]: d.T
```

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

```
[ ]: d.T.shape
```

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

```
[ ]: c * c
```

```
[ ]: array([0, 1, 4])
```

```
[ ]: d
```

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

```
[ ]: d * d
```

```
[ ]: array([[ 0,  1,  4],  
          [ 9, 16, 25]])
```

```
[ ]: print(d)
```

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

```
[ ]: print(c)
```

```
[0 1 2]
```

```
[ ]: d * c
```

```
[ ]: array([[ 0,  1,  4],  
          [ 0,  4, 10]])
```

```
[ ]: d.T
```

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

```
[ ]: c
```

```
[ ]: array([0, 1, 2])
```

```
[ ]: d.T * c
```

```
-----  
ValueError                                Traceback (most recent call last)  
<ipython-input-81-6c9888d6b7d3> in <module>  
----> 1 d.T * c  
  
ValueError: operands could not be broadcast together with shapes (3,2) (3,)
```

```
[ ]: d
```

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

```
[ ]: c
```

```
[ ]: array([0, 1, 2])
```

```
[ ]: d.shape
```

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

```
[ ]: c.shape
```

```
[ ]: (3,)
```

```
[ ]: c1 = c.reshape(1, 3)  
c1
```

```
[ ]: array([[0, 1, 2]])
```

```
[ ]: d
```

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

```
[ ]: c1.shape
```

```
[ ]: (1, 3)
```

```
[ ]: d.shape
```

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

```
[ ]: d * c1
```

```
[ ]: array([[ 0,  1,  4],  
          [ 0,  4, 10]])
```

```
[ ]: c2 = np.row_stack((c1, c1))  
c2
```

```
[ ]: array([[0, 1, 2],  
          [0, 1, 2]])
```

```
[ ]: d
```

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

```
[ ]: c2
```

```
[ ]: array([[0, 1, 2],
           [0, 1, 2]])
```

```
[ ]: d.shape
```

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

```
[ ]: c2.shape
```

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

```
[ ]: d * c2
```

```
[ ]: array([[ 0,  1,  4],
           [ 0,  4, 10]])
```

```
[ ]: d * c
```

```
[ ]: array([[ 0,  1,  4],
           [ 0,  4, 10]])
```

```
[ ]: np.column_stack((d, np.ones(d.shape[0])))
```

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

```
[ ]: d.dtype
```

```
[ ]: dtype('int64')
```

```
[ ]: np.ones((2,3), dtype = 'int64')
```

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

```
[ ]: np.column_stack((np.ones(d.shape[0], dtype = 'int64'), d))
```

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

```
[ ]: d
```

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

```
[ ]: np.ones_like(d)
```

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

```
[ ]: np.zeros_like(d)
```

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

```
[ ]: np.zeros_like(d) + 1
```

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

```
[ ]: np.ones_like(d) - 1
```

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

```
[ ]: np.eye(3)
```

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

```
[ ]: c
```

```
[ ]: array([0, 1, 2])
```

```
[ ]: np.min(c), np.max(c)
```

```
[ ]: (0, 2)
```

```
[ ]: d
```

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

```
[ ]: np.min(d), np.max(d)
```

```
[ ]: (0, 5)
```

```
[ ]: np.max(d, axis = 0)
```

```
[ ]: array([3, 4, 5])
```

```
[ ]: np.max(d, axis = 1)
```

```
[ ]: array([2, 5])
```

```
[ ]: np.sum(d)
```

```
[ ]: 15
```

```
[ ]: np.sum(d, axis = 1)
```

```
[ ]: array([ 3, 12])
```

```
[ ]: np.mean(d, axis = 1)
```

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

```
[ ]: np.mean(d, axis = 1, keepdims = True)
```

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

```
[ ]:
```