NumPyExamplesAll

January 24, 2023

1 NumPy Tutorial: Examples

```
[1]: import numpy as np

     np.ndarray
```

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

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

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

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

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

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

```
[4]: 1
```

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

```
[5]: 1
```

```
[11]: a = np.array(list(range(4)))
      a
```

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

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

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

```
[14]: a = np.arange(4) + 1
      a
```

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

```
[15]: b = np.arange(8)
      b
```

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

```
[16]: b.shape
```

```
[16]: (8,)
```

```
[17]: b.reshape(4, 2)
```

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

```
[18]: b.reshape(4, 2, order = 'F')
```

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

```
[20]: [0 for i in range(5)]
```

```
[20]: [0, 0, 0, 0, 0]
```

```
[23]: np.zeros(5).astype(int)
```

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

```
[24]: b
```

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

```
[25]: b.astype(float)
```

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

```
[26]: b
```

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

```
[27]: bb = b.reshape(4, 2)
      bb
```

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

```
[28]: bb[-1,-1] = 10
      bb
```

```
[28]: array([[ 0,  1],
              [ 2,  3],
              [ 4,  5],
              [ 6, 10]])
```

```
[29]: b
```

```
[29]: array([ 0,  1,  2,  3,  4,  5,  6, 10])
```

```
[30]: a
```

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

```
[31]: bb
```

```
[31]: array([[ 0,  1],
              [ 2,  3],
              [ 4,  5],
              [ 6, 10]])
```

```
[34]: np.column_stack((a, bb))
```

```
[34]: array([[ 1,  0,  1],
              [ 2,  2,  3],
              [ 3,  4,  5],
              [ 4,  6, 10]])
```

```
[35]: bb
```

```
[35]: array([[ 0,  1],
              [ 2,  3],
              [ 4,  5],
              [ 6, 10]])
```

```
[190]: bnew = np.column_stack((np.ones(bb.shape[0]), bb))
```

```
[191]: bnew
```

```
[191]: array([[ 1.,  0.,  1.],
              [ 1.,  2.,  3.],
              [ 1.,  4.,  5.],
              [ 1.,  6., 10.]])
```

```
[77]: data = np.loadtxt('data.txt')
```

```
[44]: data.shape
```

```
[44]: (2, 4)
```

```
[45]: data
```

```
[45]: array([[ 1. ,  0.5,  2.1, -1.7],
             [-1. ,  0.7,  3.1,  2.7]])
```

```
[46]: y = data[:, 0]
      y
```

```
[46]: array([ 1., -1.])
```

```
[47]: X = data[:,1:]
      X
```

```
[47]: array([[ 0.5,  2.1, -1.7],
             [ 0.7,  3.1,  2.7]])
```

```
[50]: b
```

```
[50]: array([ 0,  1,  2,  3,  4,  5,  6, 10])
```

```
[51]: bb
```

```
[51]: array([[ 0,  1],
             [ 2,  3],
             [ 4,  5],
             [ 6, 10]])
```

```
[52]: bnew
```

```
[52]: array([[ 1.,  0.,  1.],
             [ 1.,  2.,  3.],
             [ 1.,  4.,  5.],
             [ 1.,  6., 10.]])
```

```
[53]: a
```

[53]:
```
array([1, 2, 3, 4])
```

[54]:
```
bb.dot(a)
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Input In [54], in <cell line: 1>()
----> 1 bb.dot(a)

ValueError: shapes (4,2) and (4,) not aligned: 2 (dim 1) != 4 (dim 0)
```

[55]:
```
a.dot(bb)
```

[55]:
```
array([40, 62])
```

[56]:
```
a.reshape(1, 4)
```

[56]:
```
array([[1, 2, 3, 4]])
```

[57]:
```
a
```

[57]:
```
array([1, 2, 3, 4])
```

[58]:
```
a.reshape(4, 1)
```

[58]:
```
array([[1],
       [2],
       [3],
       [4]])
```

[59]:
```
aa = a.reshape(4, -1)
aa
```

[59]:
```
array([[1],
       [2],
       [3],
       [4]])
```

[60]:
```
bnew
```

[60]:
```
array([[ 1.,  0.,  1.],
       [ 1.,  2.,  3.],
       [ 1.,  4.,  5.],
       [ 1.,  6., 10.]])
```

[61]:
```
aa.shape
```

```
[61]: (4, 1)
```

```
[62]: bnew.shape
```

```
[62]: (4, 3)
```

```
[66]: bnew.T @ aa
```

```
[66]: array([[10.],
             [40.],
             [62.]])
```

```
[67]: aa
```

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

```
[68]: a
```

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

```
[69]: bnew
```

```
[69]: array([[ 1.,  0.,  1.],
             [ 1.,  2.,  3.],
             [ 1.,  4.,  5.],
             [ 1.,  6., 10.]])
```

```
[70]: aa
```

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

```
[71]: aa.T
```

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

```
[73]: a
```

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

```
[72]: bnew[:,0]
```

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

```
[74]: bnew[:,0] * a
```

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

```
[75]: bnew[:,0].dot(a)
```

```
[75]: 10.0
```

```
[76]: bnew * aa
```

```
[76]: array([[ 1.,  0.,  1.],
              [ 2.,  4.,  6.],
              [ 3., 12., 15.],
              [ 4., 24., 40.]])
```

```
[81]: b = np.arange(8).reshape(4,-1)
      b
```

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

```
[82]: a
```

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

```
[83]: b.ravel()
```

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

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

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

```
[85]: b.ravel(order = 'C')
```

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

```
[86]: v = b.ravel(order = 'C')
      v
```

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

```
[87]: v.reshape(4, -1)
```

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

```
[88]: v.reshape(-1, 2)
```

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

```
[89]: d = np.array([2, 2, 2, 2])
      d[0] = 1
      d[2] = -1
      d
```

```
[89]: array([ 1,  2, -1,  2])
```

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

```
[91]: d[:,newaxis] # Add a new, second dimension
```

```
[91]: array([[ 1],
             [ 2],
             [-1],
             [ 2]])
```

```
[92]: d[:,newaxis].shape
```

```
[92]: (4, 1)
```

```
[93]: d[newaxis,:].shape # Add a new, first dimension
```

```
[93]: (1, 4)
```

```
[94]: d.reshape(4,-1)
```

```
[94]: array([[ 1],
             [ 2],
             [-1],
             [ 2]])
```

```
[95]: np.eye(10)
```

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

```
[96]: np.eye(10) + 10
```

```
[96]: array([[11., 10., 10., 10., 10., 10., 10., 10., 10., 10.],
             [10., 11., 10., 10., 10., 10., 10., 10., 10., 10.],
             [10., 10., 11., 10., 10., 10., 10., 10., 10., 10.],
             [10., 10., 10., 11., 10., 10., 10., 10., 10., 10.],
             [10., 10., 10., 10., 11., 10., 10., 10., 10., 10.],
             [10., 10., 10., 10., 10., 11., 10., 10., 10., 10.],
             [10., 10., 10., 10., 10., 10., 11., 10., 10., 10.],
             [10., 10., 10., 10., 10., 10., 10., 11., 10., 10.],
             [10., 10., 10., 10., 10., 10., 10., 10., 11., 10.],
             [10., 10., 10., 10., 10., 10., 10., 10., 10., 11.]])
```

```
[97]: np.ones((4, 2))
```

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

```
[98]: np.ones((4, 2)) * 2
```

```
[98]: array([[2., 2.],
             [2., 2.],
             [2., 2.],
             [2., 2.]])
```

```
[99]: np.sum(a)
```

```
[99]: 10
```

```
[100]: a.sum()
```

```
[100]: 10
```

```
[101]: a.max()
```

```
[101]: 4
```

```
[102]: a.min()
```

```
[102]: 1
```

```
[103]: b.sum()
```

```
[103]: 28
```

```
[104]: b.sum(axis = 0)
```

```
[104]: array([12, 16])
```

```
[105]: b.sum(axis = 1)
```

```
[105]: array([ 1,  5,  9, 13])
```

```
[106]: b.max(axis = 1)
```

```
[106]: array([1, 3, 5, 7])
```

```
[108]: b.argmax(axis = 1)
```

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

```
[109]: b.max()
```

```
[109]: 7
```

```
[110]: b.max(axis = 1)
```

```
[110]: array([1, 3, 5, 7])
```

```
[111]: b.argmax(axis = 1)
```

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

```
[112]: a
```

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

```
[113]: b
```

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

[114]:
```
np.maximum(a,b)
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Input In [114], in <cell line: 1>()
----> 1 np.maximum(a,b)

ValueError: operands could not be broadcast together with shapes (4,) (4,2)
```

[115]:
```
np.maximum(a[:,newaxis], b)
```

[115]:
```
array([[1, 1],
       [2, 3],
       [4, 5],
       [6, 7]])
```

[116]:
```
a
```

[116]:
```
array([1, 2, 3, 4])
```

[117]:
```
a[:2]
```

[117]:
```
array([1, 2])
```

[118]:
```
b
```

[118]:
```
array([[0, 1],
       [2, 3],
       [4, 5],
       [6, 7]])
```

[119]:
```
b.shape
```

[119]:
```
(4, 2)
```

[120]:
```
a[:2].shape
```

[120]:
```
(2,)
```

[121]:
```
np.maximum(a[:2], b)
```

[121]:
```
array([[1, 2],
       [2, 3],
       [4, 5],
       [6, 7]])
```

```
[122]: a[:2] * b
```

```
[122]: array([[ 0,  2],
              [ 2,  6],
              [ 4, 10],
              [ 6, 14]])
```

```
[123]: a[:2] + b
```

```
[123]: array([[1, 3],
              [3, 5],
              [5, 7],
              [7, 9]])
```

```
[124]: b / a[:2]
```

```
[124]: array([[0. , 0.5],
              [2. , 1.5],
              [4. , 2.5],
              [6. , 3.5]])
```

```
[125]: b
```

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

```
[126]: b.mean()
```

```
[126]: 3.5
```

```
[130]: b.sum() / b.size
```

```
[130]: 3.5
```

```
[131]: b.mean(axis = 0)
```

```
[131]: array([3., 4.])
```

```
[132]: b.mean(axis = 1)
```

```
[132]: array([0.5, 2.5, 4.5, 6.5])
```

```
[133]: b.std()
```

```
[133]: 2.29128784747792
```

```
[134]: b.std(axis = 0)
```

```
[134]: array([2.23606798, 2.23606798])
```

```
[135]: a
```

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

```
[136]: np.column_stack((a, a))
```

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

```
[137]: np.vstack((a, a))
```

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

```
[138]: np.stack((a, a))
```

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

```
[139]: np.stack((a, a), axis = 0)
```

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

```
[140]: np.stack((a, a), axis = 1)
```

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

```
[141]: b
```

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

```
[142]: np.split(b, 2)
```

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

```
[143]: np.split(b, 2, axis = 1)
```

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

```
[144]: [b1, b2] = np.split(b, 2)
```

```
[145]: b
```

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

```
[146]: b1
```

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

```
[147]: b2
```

```
[147]: array([[4, 5],
              [6, 7]])
```

```
[148]: b1[0,0] = -1
```

```
[149]: b1
```

```
[149]: array([[-1,  1],
              [ 2,  3]])
```

```
[150]: b2
```

```
[150]: array([[4, 5],
              [6, 7]])
```

[151]:
```
b
```

[151]:
```
array([[-1,  1],
       [ 2,  3],
       [ 4,  5],
       [ 6,  7]])
```

1.1 Linear algebra module

[152]:
```
from numpy import linalg as la
```

[153]:
```
a
```

[153]:
```
array([1, 2, 3, 4])
```

[154]:
```
la.norm(a)
```

[154]:
```
5.477225575051661
```

[155]:
```
np.sum(a*a)
```

[155]:
```
30
```

[156]:
```
np.sqrt(np.sum(a*a))
```

[156]:
```
5.477225575051661
```

[157]:
```
np.sqrt(a.dot(a))
```

[157]:
```
5.477225575051661
```

[158]:
```
b
```

[158]:
```
array([[-1,  1],
       [ 2,  3],
       [ 4,  5],
       [ 6,  7]])
```

[159]:
```
b[0,0] = 0
b
```

[159]:
```
array([[0, 1],
       [2, 3],
       [4, 5],
       [6, 7]])
```

[160]:
```
la.norm(b)
```

```
[160]: 11.832159566199232
```

```
[161]: la.norm(b, axis = 0)
```

```
[161]: array([7.48331477, 9.16515139])
```

```
[162]: la.norm(b, axis = 1)
```

```
[162]: array([1.        , 3.60555128, 6.40312424, 9.21954446])
```

```
[163]: a
```

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

```
[164]: a[:, newaxis].shape
```

```
[164]: (4, 1)
```

```
[165]: aa = a[:, newaxis]
       aa
```

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

```
[166]: aa @ aa.T
```

```
[166]: array([[ 1,  2,  3,  4],
              [ 2,  4,  6,  8],
              [ 3,  6,  9, 12],
              [ 4,  8, 12, 16]])
```

```
[167]: np.outer(a, a)
```

```
[167]: array([[ 1,  2,  3,  4],
              [ 2,  4,  6,  8],
              [ 3,  6,  9, 12],
              [ 4,  8, 12, 16]])
```

```
[168]: la.matrix_rank(np.outer(a,a))
```

```
[168]: 1
```

```
[169]: c = np.random.randint(0, 10, (4, 4))
       c
```

```
[169]: array([[8, 6, 4, 1],
              [7, 9, 1, 7],
              [3, 9, 0, 5],
              [7, 3, 8, 4]])
```

```
[170]: la.matrix_rank(c)
```

```
[170]: 4
```

```
[171]: la.eig(np.eye(3))
```

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

```
[172]: c
```

```
[172]: array([[8, 6, 4, 1],
              [7, 9, 1, 7],
              [3, 9, 0, 5],
              [7, 3, 8, 4]])
```

```
[173]: np.trace(c)
```

```
[173]: 21
```

```
[174]: qr = la.qr(c)
       qr
```

```
[174]: (array([[-0.61177529,  0.1870392 ,  0.35381152, -0.68232305],
               [-0.53530338, -0.32389716,  0.46026809,  0.62983666],
               [-0.22941573, -0.80746192, -0.47592346, -0.26243194],
               [-0.53530338,  0.45619318, -0.66065691,  0.26243194]]),
        array([[-13.07669683, -12.1590339 ,  -7.26483157,  -7.64719113],
               [  0.        ,  -7.69141695,   4.07380506,  -4.29277779],
               [  0.        ,   0.        ,  -3.40974112,  -1.44655684],
               [  0.        ,   0.        ,   0.        ,   3.46410162]]))
```

```
[175]: Q = qr[0]
       R = qr[1]
```

```
[176]: Q
```

```
[176]: array([[-0.61177529,  0.1870392 ,  0.35381152, -0.68232305],
              [-0.53530338, -0.32389716,  0.46026809,  0.62983666],
              [-0.22941573, -0.80746192, -0.47592346, -0.26243194],
          [-0.53530338,  0.45619318, -0.66065691,  0.26243194]])
```

```
[177]: R
```

```
[177]: array([[-13.07669683, -12.1590339 ,  -7.26483157,  -7.64719113],
              [  0.        ,  -7.69141695,   4.07380506,  -4.29277779],
              [  0.        ,   0.        ,  -3.40974112,  -1.44655684],
              [  0.        ,   0.        ,   0.        ,   3.46410162]])
```

```
[178]: Q @ R
```

```
[178]: array([[ 8.00000000e+00,  6.00000000e+00,  4.00000000e+00,
                1.00000000e+00],
              [ 7.00000000e+00,  9.00000000e+00,  1.00000000e+00,
                7.00000000e+00],
              [ 3.00000000e+00,  9.00000000e+00, -1.55431223e-15,
                5.00000000e+00],
              [ 7.00000000e+00,  3.00000000e+00,  8.00000000e+00,
                4.00000000e+00]])
```

```
[179]: Q.dot(R)
```

```
[179]: array([[ 8.00000000e+00,  6.00000000e+00,  4.00000000e+00,
                1.00000000e+00],
              [ 7.00000000e+00,  9.00000000e+00,  1.00000000e+00,
                7.00000000e+00],
              [ 3.00000000e+00,  9.00000000e+00, -1.55431223e-15,
                5.00000000e+00],
              [ 7.00000000e+00,  3.00000000e+00,  8.00000000e+00,
                4.00000000e+00]])
```

```
[180]: Q[0]
```

```
[180]: array([-0.61177529,  0.1870392 ,  0.35381152, -0.68232305])
```

```
[181]: Q[0] @ Q[0]
```

```
[181]: 1.0
```

```
[182]: Q[0] @ Q[1]
```

```
[182]: -3.3306690738754696e-16
```

```
[183]: np.round(Q[0] @ Q[1])
```

```
[183]: -0.0
```

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

```
[185]: np.linalg.det(c)
```

```
[185]: 1188.0000000000005
```

```
[186]: la.solve
```

```
[186]: <function numpy.linalg.solve(a, b)>
```

```
[187]: la.inv(c)
```

```
[187]: array([[ 0.10353535,  0.21212121, -0.25505051, -0.07828283],
              [ 0.07491582, -0.17171717,  0.23821549, -0.01599327],
              [-0.02020202, -0.21212121,  0.17171717,  0.16161616],
              [-0.1969697 ,  0.18181818, -0.07575758,  0.07575758]])
```

```
[188]: # Solve the system of equations x0 + 2 * x1 = 1 and 3 * x0 + 5 * x1 = 2
       a = np.array([[1, 2], [3, 5]])
       b = np.array([1, 2])
       x = np.linalg.solve(a, b)
       x
```

```
[188]: array([-1.,  1.])
```

```
[189]: a @ x
```

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