

NumPy ExamplesAll

September 5, 2024

1 NumPy Tutorial: Examples

```
[109]: import numpy as np  
  
np.ndarray
```

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

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

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

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

```
[111]: (4,)
```

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

```
[112]: 1
```

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

```
[113]: 1
```

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

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

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

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

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

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

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

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

```
[118]: b.shape
```

[118]: (8,)

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

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

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

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

```
[121]: [0 for i in range(5)]
```

[121]: [0, 0, 0, 0, 0]

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

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

```
[123]: b
```

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

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

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

```
[125]: b
```

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

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

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

```
[6, 7]])
```

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

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

```
[128]: b
```

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

```
[129]: a
```

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

```
[130]: bb
```

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

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

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

```
[132]: bb
```

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

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

```
[134]: bnew
```

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

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

```
[136]: data.shape
```

```
[136]: (2, 4)
```

```
[137]: data
```

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

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

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

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

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

```
[140]: b
```

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

```
[141]: bb
```

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

```
[142]: bnew
```

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

```
[143]: a
```

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

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

```
-----  
ValueError
```

```
Traceback (most recent call last)
```

```
Cell In[144], line 1
```

```
----> 1 bb.dot(a)
```

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

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

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

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

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

```
[147]: a
```

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

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

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

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

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

```
[150]: bnew
```

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

```
[151]: aa.shape
```

```
[151]: (4, 1)
```

```
[152]: bnew.shape
```

```
[152]: (4, 3)
```

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

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

```
[154]: aa
```

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

```
[155]: a
```

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

```
[156]: bnew
```

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

```
[157]: aa
```

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

```
[158]: aa.T
```

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

```
[159]: a
```

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

```
[160]: bnew[:,0]
```

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

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

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

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

```
[162]: np.float64(10.0)
```

```
[163]: bnew * aa
```

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

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

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

```
[165]: a
```

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

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

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

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

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

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

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

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

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

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

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

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

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

```
[6, 7]])
```

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

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

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

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

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

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

```
[175]: (4, 1)
```

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

```
[176]: (1, 4)
```

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

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

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

```
[178]: 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.]])
```

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



```
[179]: 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.]])
```

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

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

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

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

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

```
[182]: np.int64(10)
```

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

```
[183]: np.int64(10)
```

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

```
[184]: np.int64(4)
```

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

```
[185]: np.int64(1)
```

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

```
[186]: np.int64(28)
```

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

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

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

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

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

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

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

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

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

```
[191]: np.int64(7)
```

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

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

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

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

```
[194]: a
```

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

```
[195]: b
```

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

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

```
-----  
ValueError                                Traceback (most recent call last)  
Cell In[196], line 1  
----> 1 np.maximum(a,b)  
  
ValueError: operands could not be broadcast together with shapes (4,) (4,2)
```

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

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

```
[198]: a
```

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

```
[199]: a[:2]
```

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

```
[200]: b
```

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

```
[201]: b.shape
```

```
[201]: (4, 2)
```

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

```
[202]: (2,)
```

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

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

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

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

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

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

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

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

```
[207]: b
```

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

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

```
[208]: np.float64(3.5)
```

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

```
[209]: np.float64(3.5)
```

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

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

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

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

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

```
[212]: np.float64(2.29128784747792)
```

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

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

```
[214]: a
```

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

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

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

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

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

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

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

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

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

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

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

```
[220]: b
```

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

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

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

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

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

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

```
[224]: b
```

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

```
[225]: b1
```

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

```
[226]: b2
```

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

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

```
[228]: b1
```

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

```
[229]: b2
```

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

```
[230]: b
```

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

1.1 Linear algebra module

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

```
[232]: a
```

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

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

```
[233]: np.float64(5.477225575051661)
```

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

```
[234]: np.int64(30)
```

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

```
[235]: np.float64(5.477225575051661)
```

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

```
[236]: np.float64(5.477225575051661)
```

```
[237]: b
```

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

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

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

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

```
[239]: np.float64(11.832159566199232)
```

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

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

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

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

```
[242]: a
```

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

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

```
[243]: (4, 1)
```

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

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

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

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

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

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

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

```
[247]: np.int64(1)
```

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

```
[248]: array([[5, 0, 0, 8],  
            [3, 6, 3, 8],  
            [3, 1, 6, 0],  
            [5, 7, 5, 5]])
```

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

```
[249]: np.int64(4)
```

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

```
[250]: EigResult(eigenvalues=array([1., 1., 1.]), eigenvectors=array([[1., 0., 0.],  
            [0., 1., 0.],  
            [0., 0., 1.])))
```

```
[251]: c
```

```
[251]: array([[5, 0, 0, 8],  
            [3, 6, 3, 8],  
            [3, 1, 6, 0],  
            [5, 7, 5, 5]])
```



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

```
[252]: np.int64(22)
```

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

```
[253]: QRResult(Q=array([[ -0.60633906,  0.65201672,  0.44257509, -0.10655718],
 [ -0.36380344, -0.55887147,  0.14914174, -0.73011402],
 [ -0.36380344,  0.23286311, -0.88424077, -0.1775953 ],
 [ -0.60633906, -0.4564117 , -0.00151567,  0.65118277]]), R=array([[
-8.24621125,  -6.7909975 ,  -6.30592625, -10.79283531],
 [ 0.          , -6.31524765,  -2.56149426,  -1.53689655],
 [ 0.          ,  0.          , -4.86559773,  4.72615627],
 [ 0.          ,  0.          ,  0.          , -3.43745572]]))
```

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

```
[255]: Q
```

```
[255]: array([[ -0.60633906,  0.65201672,  0.44257509, -0.10655718],
 [ -0.36380344, -0.55887147,  0.14914174, -0.73011402],
 [ -0.36380344,  0.23286311, -0.88424077, -0.1775953 ],
 [ -0.60633906, -0.4564117 , -0.00151567,  0.65118277]])
```

```
[256]: R
```

```
[256]: array([[ -8.24621125,  -6.7909975 ,  -6.30592625, -10.79283531],
 [ 0.          , -6.31524765,  -2.56149426,  -1.53689655],
 [ 0.          ,  0.          , -4.86559773,  4.72615627],
 [ 0.          ,  0.          ,  0.          , -3.43745572]])
```

```
[257]: Q @ R
```

```
[257]: array([[5.00000000e+00, 0.00000000e+00, 8.88178420e-16, 8.00000000e+00],
 [3.00000000e+00, 6.00000000e+00, 3.00000000e+00, 8.00000000e+00],
 [3.00000000e+00, 1.00000000e+00, 6.00000000e+00, 5.55111512e-16],
 [5.00000000e+00, 7.00000000e+00, 5.00000000e+00, 5.00000000e+00]])
```

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

```
[258]: array([[5.00000000e+00, 0.00000000e+00, 8.88178420e-16, 8.00000000e+00],
 [3.00000000e+00, 6.00000000e+00, 3.00000000e+00, 8.00000000e+00],
 [3.00000000e+00, 1.00000000e+00, 6.00000000e+00, 5.55111512e-16],
 [5.00000000e+00, 7.00000000e+00, 5.00000000e+00, 5.00000000e+00]])
```

```
[259]: Q[0]
```

```
[259]: array([-0.60633906,  0.65201672,  0.44257509, -0.10655718])
```

```
[260]: Q[0] @ Q[0]
```

```
[260]: np.float64(1.0000000000000004)
```

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

```
[261]: np.float64(-8.326672684688674e-17)
```

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

```
[262]: np.float64(-0.0)
```

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

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

```
[264]: np.float64(-870.9999999999993)
```

```
[265]: la.solve
```

```
[265]: <function solve at 0x1122329b0>
```

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

```
[266]: array([[ 0.15040184, -0.33983927, -0.08266361,  0.30309989],
          [-0.08610792, -0.03444317, -0.1435132 ,  0.19288175],
          [-0.0608496 ,  0.17566016,  0.23191734, -0.1836969 ],
          [ 0.03099885,  0.21239954,  0.05166475, -0.18943743]])
```

```
[267]: # Solve the system of equations  $x_0 + 2 * x_1 = 1$  and  $3 * x_0 + 5 * x_1 = 2$ 
a = np.array([[1, 2], [3, 5]])
b = np.array([1, 2])
x = np.linalg.solve(a, b)
x
```

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

```
[268]: a @ x
```

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