

```
import numpy as np
```

zad.1

```
a = np.array([1, 4, 3, 5, 3])
b1 = np.array([3.14, 4, 2, 3])
b2 = np.array([3.14, 4, 2, 3], dtype=np.int32)
c = np.array([[2, 4, 6], [1, 2, 3]])
d = np.array([[-3.0, 2.3],
              [0.1, 5.0],
              [8.0, 11.0]])
e = np.array([[[[2, 4], [1, 2], [8, 9]],
               [[2, 4], [1, 2], [8, 9]],
               [[2, 4], [1, 2], [8, 9]],
               [[7, 6], [3, 4], [0, 8]]]])
```

zad.2

```
print("ndim= ", e.ndim)
print("shape= ", e.shape)
print("size= ", e.size)
print("dtype= ", e.dtype)
print("itemsize= ", e.itemsize)
print("nbytes= ", e.nbytes)
```

```
ndim= 4
shape= (1, 4, 3, 2)
size= 24
dtype= int64
itemsize= 8
nbytes= 192
```

zad.3

```
X1 = np.random.random((4,3))
X2 = np.zeros((2,1))
X3 = np.full((6,2,1),2)
X4 = np.random.randint(0,7,(5,8))
X5 = np.eye(5)
```

```
print(X4)
print("")
print(X4[0])
print("")
print(X4[0,2])
print("")
print(X4[:,2])
print("")
print(X4[1:3,2:5])
```

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

```
[5 2 4 3 4 6 6 4]
```

```
4
```

```
[4 6 3 3 1]
```

```
[[6 0 6]
 [3 1 6]]
```

zad.4

```
A = np.random.randint(0,6,(4,5))
A
```

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

```
B = A[:,1:3]
```

B

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

```
C = A[1:3, 1:].copy()
C1 = C.copy()
C1[0] = [11,11,11,11]
```

A

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

C

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

C1

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

zad.5

```
D = np.array(np.arange(0,12))
D
```

```
array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11])
```

```
print(D.reshape(2,6))
```

```
[[ 0  1  2  3  4  5]
 [ 6  7  8  9 10 11]]
```

```
print(D.reshape(3
                ,4))
```

```
[[ 0  1  2  3]
 [ 4  5  6  7]
 [ 8  9 10 11]]
```

```
print(D.reshape(3,4,1))
```

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

```
[[ 4]
 [ 5]
 [ 6]
 [ 7]]
```

```
[[ 8]
 [ 9]
 [10]
 [11]]]
```

```
print(D.reshape(1,3,1,4,1,1))
```

```
[[[[[ 0]]
```

```
[[ 1]]
```

```
[[ 2]]
```

```
[[ 3]]]]
```

```
[[[ 4]]
```

```
[[ 5]]
```

```
[[ 6]]
```

```

[[ 7]]]

[[[ 8]]

[[ 9]]

[[10]]

[[11]]]]]]

```

```

print(D.reshape(3,2,2))

[[[ 0  1]
  [ 2  3]]

 [[ 4  5]
  [ 6  7]]

 [[ 8  9]
  [10 11]]]

```

```

print(D.reshape(6,2,1))

[[[ 0]
  [ 1]]

 [[ 2]
  [ 3]]

 [[ 4]
  [ 5]]

 [[ 6]
  [ 7]]

 [[ 8]
  [ 9]]

 [[10]
  [11]]]

```

```

print(D.reshape(3,3))

```

```

-----
ValueError                                Traceback (most recent call last)
<ipython-input-88-e076888ce4fb> in <module>()
----> 1 print(D.reshape(3,3))

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

```

SEARCH STACK OVERFLOW

zad.6

przykład)

```

p1 = np.array([[2,4],[1,2]])
p2 = np.array([[-3,5],[-7,8]])

print("p1 =\n", p1)
print("p2 =\n", p2)
print("p1.shape =", p1.shape)
print("p2.shape =", p2.shape)
print(np.concatenate([p1,p2], axis = 0))
print(np.concatenate([p1,p2], axis = 1))
print(np.concatenate([p2,p1], axis = 0))
print(np.concatenate([p2,p1], axis = 1))

p1 =
[[2 4]
 [1 2]]
p2 =
[[-3  5]
 [-7  8]]
p1.shape = (2, 2)
p2.shape = (2, 2)
[[ 2  4]
 [ 1  2]
 [-3  5]
 [-7  8]]
[[ 2  4 -3  5]

```

```
[ 1  2 -7  8]]
[[-3  5]
 [-7  8]
 [ 2  4]
 [ 1  2]]
[[-3  5  2  4]
 [-7  8  1  2]]
```

6.1)

```
np.concatenate((c, d), axis=None)
```

```
array([ 2. ,  4. ,  6. ,  1. ,  2. ,  3. , -3. ,  2.3,  0.1,  5. ,  8. ,
        11. ])
```

```
np.concatenate((d, c), axis=None)
```

```
array([-3. ,  2.3,  0.1,  5. ,  8. , 11. ,  2. ,  4. ,  6. ,  1. ,  2. ,
        3. ])
```

6.2)

```
np.concatenate((c, X2), axis=1)
```

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

```
np.concatenate((X2, c), axis=1)
```

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

```
np.concatenate((c, X2), axis=None)
```

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

```
np.concatenate((X2, c), axis=None)
```

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

6.3)

```
np.concatenate((X1, X2), axis=None)
```

```
array([0.3035459 , 0.92731896, 0.38682981, 0.97819047, 0.21710461,
        0.52868867, 0.85765635, 0.11342225, 0.67924504, 0.36125249,
        0.4724143  , 0.39803007, 0.          , 0.          ])
```

```
np.concatenate((X2, X1), axis=None)
```

```
array([0.          , 0.          , 0.3035459 , 0.92731896, 0.38682981,
        0.97819047, 0.21710461, 0.52868867, 0.85765635, 0.11342225,
        0.67924504, 0.36125249, 0.4724143 , 0.39803007])
```

6.4)

```
Z5T1 = D.reshape(3, 4)
```

```
Z5T2 = D.reshape(4, 3)
```

```
print(Z5T1)
```

```
print("")
```

```
print(Z5T2)
```

```
[[ 0  1  2  3]
 [ 4  5  6  7]
 [ 8  9 10 11]]
```

```
[[ 0  1  2]
 [ 3  4  5]
 [ 6  7  8]
 [ 9 10 11]]
```

```
np.concatenate((Z5T1, Z5T2), axis=None)
```

```
array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11,  0,  1,  2,  3,  4,
        5,  6,  7,  8,  9, 10, 11])
```

```
np.concatenate((Z5T2, Z5T1), axis=None)
```

```
array([-3. ,  2.3,  0.1,  5. ,  8. , 11. ,  2. ,  4. ,  6. ,  1. ,  2. ,
        3. ])
```

```
np.concatenate((Z512, Z511), axis=None)
```

```
array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11,  0,  1,  2,  3,  4,
        5,  6,  7,  8,  9, 10, 11])
```

zad.7

7.1)

```
print(d+6)
print("")
print(d-8)
print("")
print(d*8)
print("")
print(d/8)
```

```
[[ 3.  8.3]
 [ 6.1 11. ]
 [14. 17. ]]
```

```
[[ -11.  -5.7]
 [ -7.9  -3. ]
 [  0.   3. ]]
```

```
[[ -24.  18.4]
 [  0.8  40. ]
 [ 64.  88. ]]
```

```
[[ -0.375  0.2875]
 [ 0.0125  0.625 ]
 [ 1.      1.375 ]]
```

7.2)

```
print(abs(8))
print("")
print(np.exp(d))
print("")
print(np.power(d,4))
print("")
print(np.log(d))
```

```
8
```

```
[[4.97870684e-02 9.97418245e+00]
 [1.10517092e+00 1.48413159e+02]
 [2.98095799e+03 5.98741417e+04]]
```

```
[[8.10000e+01 2.79841e+01]
 [1.00000e-04 6.25000e+02]
 [4.09600e+03 1.46410e+04]]
```

```
[[          nan  0.83290912]
 [-2.30258509  1.60943791]
 [ 2.07944154  2.39789527]]
```

```
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:7: RuntimeWarning: invalid value encountered in log
import sys
```

7.3)

```
A = [[1],[2]];
print(X2)
print("")
print(A+X2)
print("")
print(A-X2)
print("")
print(A*X2)
```

```
[[0.]
 [0.]]
```

```
[[1.]
 [2.]]
```

```
[[1.]
 [2.]]
```

```
[[0.]
 [0.]]
```

zad.8

8.1)

```
A = np.array([[1, 1, 1], [1, 1, 1]])
B = np.array([[0, 1, 2]])
print(A+B)
```

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

8.2)

```
C = np.array([[0]], [[1]], [[2]])
print(C+B)
```

```
[[0 1 2]]
```

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

```
[[2 3 4]]
```