```
import numpy as np
```

```
x=np.zeros((2,3)) #2d array
print(x)
print("---------------------------")
y=np.ones((2,3,2)) #3d array
print(y)
```

```
[[0. 0. 0.]
 [0. 0. 0.]]
---------------------------
[[[1. 1.]
  [1. 1.]
  [1. 1.]]

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

```
7*y
```

```
array([[[7., 7.],
        [7., 7.],
        [7., 7.]],

       [[7., 7.],
        [7., 7.],
        [7., 7.]]])
```

```
np.random.rand(2,3) #0-1
```

```
array([[0.30023249, 0.20008134, 0.11240225],
       [0.80631065, 0.31349979, 0.42699784]])
```

```
#generate random
np.random.randn(2,3)
```

```
array([[-0.87883443, -0.23482361,  0.0447543 ],
       [-0.64630004, -1.34078847, -0.4041143 ]])
```

```
np.random.randint(0,100,(2,3))
```

```
array([[43, 89, 83],
       [77, 99, 75]])
```

```
np.arange(7,70,10) #step size at end
```

```
array([ 7, 17, 27, 37, 47, 57, 67])
```

```
np.linspace(1,100,5) #no of equally spaced at end
```

```
array([  1.  ,  25.75,  50.5 ,  75.25, 100.  ])
```

```
x=np.array([[True,False],[False,True]])
print(x)
```

```
[[ True False]
 [False  True]]
```

```
str_array = np.array(['1.4','6.8','9.6','10.1'])
str_array
```

```
array(['1.4', '6.8', '9.6', '10.1'], dtype='<U4')
```

```
str_array=np.array(str_array,dtype=float)
str_array
```

```
array([ 1.4,  6.8,  9.6, 10.1])
```

```
str_array=np.array(str_array,dtype=int)
str_array
```

```
array([ 1,  6,  9, 10])
```

```
#indexing an array and fancy indexing
arr4=np.random.randint(0,100,(2,5,5))
arr4
```

```
array([[[83, 64, 76, 74, 43],
        [78, 26, 13,  1, 38],
        [40, 47, 43, 78, 32],
        [ 4, 87, 98, 57, 70],
        [ 8, 45, 96,  2, 49]],

       [[35,  3, 80, 35, 78],
        [80, 53, 41, 62, 49],
        [90, 31, 48, 49, 26],
        [98, 21, 11, 83, 65],
        [96, 17, 37, 82, 94]]])
```

```
arr4.flatten()
```

```
array([83, 64, 76, 74, 43, 78, 26, 13,  1, 38, 40, 47, 43, 78, 32,  4, 87,
       98, 57, 70,  8, 45, 96,  2, 49, 35,  3, 80, 35, 78, 80, 53, 41, 62,
       49, 90, 31, 48, 49, 26, 98, 21, 11, 83, 65, 96, 17, 37, 82, 94])
```

```
arr4[1,2:4,2:5]
#or
arr4[1,2:4,2:]
```

```
array([[48, 49, 26],
       [11, 83, 65]])
```

```
arr4[1,3:5,1:4]
```

```
array([[21, 11, 83],
       [17, 37, 82]])
```

```
arr4[:,4,:]
```

```
array([[ 8, 45, 96,  2, 49],
       [96, 17, 37, 82, 94]])
```

```
arr4[:,2:4,2:4]
```

```
array([[[43, 78],
        [98, 57]],

       [[48, 49],
        [11, 83]]])
```

```
i=int(input("i "))
j=int(input("j "))
k=int(input("k "))
arr4[i,j,k]
```

```
i 1
j 1
k 1
53
```

```
arr4%2==0
```

even no. T/f

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

       [[False, False,  True, False,  True],
        [ True, False, False,  True, False],
        [ True, False,  True, False,  True],
        [ True, False, False, False, False],
        [ True, False, False,  True,  True]]])
```

```
str_array=np.array(arr4%2==0,dtype=int)
str_array
```

even no. 1/0

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

       [[0, 0, 1, 0, 1],
        [1, 0, 0, 1, 0],
        [1, 0, 1, 0, 1],
```

```
            [1, 0, 0, 0, 0],
            [1, 0, 0, 1, 1]]])
```

even no. in 1D array

```
arre=arr4[arr4 % 2 == 0]
arre
```

```
array([64, 76, 74, 78, 26, 38, 40, 78, 32,  4, 98, 70,  8, 96,  2, 80, 78,
       80, 62, 90, 48, 26, 98, 96, 82, 94])
```

same

```
arre=np.array(arr4[arr4 % 2 == 0])
arre
```

```
array([64, 76, 74, 78, 26, 38, 40, 78, 32,  4, 98, 70,  8, 96,  2, 80, 78,
       80, 62, 90, 48, 26, 98, 96, 82, 94])
```

odd no

```
arro=arr4[arr4 % 2 == 1]
arro
```

```
array([83, 43, 13,  1, 47, 43, 87, 57, 45, 49, 35,  3, 35, 53, 41, 49, 31,
       49, 21, 11, 83, 65, 17, 37])
```

odd no & no >5

```
arro=arr4[(arr4 % 2 == 1)&(arr4 > 5)] # dont use and
arro
```

```
array([83, 43, 13, 47, 43, 87, 57, 45, 49, 35, 35, 53, 41, 49, 31, 49, 21,
       11, 83, 65, 17, 37])
```

✓ 0s completed at 1:53 PM