numpy, math

In [3]:
```
import math as m
```

In [4]:
```
q = m.sqrt(9)
```

In [5]:
```
print(round(q))
```

```
3
```

In [6]:
```
import numpy as np
```

In [7]:
```
np.__version__
```

```
'1.24.3'
```

In [8]:
```
import pandas as pd
pd.__version__
```

```
'2.0.3'
```

In [106]:
```
import random as r
# function for otp generation
def otpgen():
    otp = ""
    for i in range(5):
        otp += str(r.randint(0,8))
    print ("Your One Time Password is ")
    print (otp)
otpgen()
```

```
Your One Time Password is
40015
```

In [109]:
```
import random as r
# function for otp generation
def otpgen():
    otp = 0
    for i in range(5):
        otp += r.randint(10,89)
    print ("Your One Time Password is ")
    print (otp)
otpgen()
```

```
Your One Time Password is
223
```

In [10]:
```
z = list(range(1,8))
z
```

```
[1, 2, 3, 4, 5, 6, 7]
```

In [11]:
```
z.append(3)
z
```

```
[1, 2, 3, 4, 5, 6, 7, 3]
```

Array

arange(1 arg | 2arg | 3 arg) - (1st < 2nd)

In [12]:
```
a = np.array(z)
print(type(a))
print(type(z))
```

```
<class 'numpy.ndarray'>
<class 'list'>
```

In [18]:
```
# what is the main difference between list & array
# you cannot create an array of different data types
b = np.array(z)
b
```

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

In [21]:
```
b = np.arange(9)
b
```

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

In [28]:
```
np.arange(10,20)
```

```
array([10, 11, 12, 13, 14, 15, 16, 17, 18, 19])
```

In [29]:
```
np.arange(20,10)
```

```
array([], dtype=int32)
```

In [30]:
```
np.arange(10,10)
```

```
array([], dtype=int32)
```

In [31]:
```
np.arange(10,30,5)
```

```
array([10, 15, 20, 25])
```

In [36]:
```
np.arange(40,30,5)
```

```
array([], dtype=int32)
```

Zeros(), ones() - n args and no condition 1st - rows, 2nd - column

In [35]:
```
np.zeros((3,2))
```

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

In [42]:
```
np.zeros((3,2))
```

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

In [33]:
```
np.zeros((5,4))
```

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

In [37]:
```
np.ones((3,2))
```

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

rand, randint in random function

In [43]:
```
np.random.randint(10,30)
```

```
10
```

In [47]:
```
#constant number
print(np.random.randint(1))
np.random.randint(10,11)
```

```
0
```

```
10
```

In [49]:
```
np.random.randint(1,3,5)  #1st - start, 2nd - end, 3rd - no of terms
```

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

In [52]:
```
r = np.random.randint(10,20,(4,5)) #generre the element 10 -30 with 4*4 mat
r
```

```
array([[12, 15, 11, 17, 19],
       [12, 10, 13, 16, 18],
       [16, 13, 15, 17, 12],
       [19, 19, 17, 12, 17]])
```

In [53]:
```
r.max()
```

```
19
```

In [54]:
```
r.min()
```

```
10
```

In [55]:
```
r.mean()
```

```
15.0
```

In [60]:
```
from numpy import *
median(r)
```

```
15.5
```

In [64]:
```
from numpy import *
a = array([1,2,3,4,9,7])
median(a)
```

```
3.5
```

In [65]:
```
a.reshape(3,2)
```

```
array([[1, 2],
       [3, 4],
       [9, 7]])
```

In [68]:
```
a.reshape(1,4)  # no of elements = limit multiple (6 != 1*4)
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Cell In[68], line 1
----> 1 a.reshape(1,4)

ValueError: cannot reshape array of size 6 into shape (1,4)
```

Indexing

In [69]:
```
mat = np.arange(0,100).reshape(10,10)
mat
```

```
array([[ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9],
       [10, 11, 12, 13, 14, 15, 16, 17, 18, 19],
       [20, 21, 22, 23, 24, 25, 26, 27, 28, 29],
       [30, 31, 32, 33, 34, 35, 36, 37, 38, 39],
       [40, 41, 42, 43, 44, 45, 46, 47, 48, 49],
       [50, 51, 52, 53, 54, 55, 56, 57, 58, 59],
       [60, 61, 62, 63, 64, 65, 66, 67, 68, 69],
       [70, 71, 72, 73, 74, 75, 76, 77, 78, 79],
       [80, 81, 82, 83, 84, 85, 86, 87, 88, 89],
       [90, 91, 92, 93, 94, 95, 96, 97, 98, 99]])
```

In [70]:
```
row = 2
col = 5
mat[row,col]
```

```
25
```

Slicing

In [71]:
```
1 mat[:]
```

```
array([[ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9],
       [10, 11, 12, 13, 14, 15, 16, 17, 18, 19],
       [20, 21, 22, 23, 24, 25, 26, 27, 28, 29],
       [30, 31, 32, 33, 34, 35, 36, 37, 38, 39],
       [40, 41, 42, 43, 44, 45, 46, 47, 48, 49],
       [50, 51, 52, 53, 54, 55, 56, 57, 58, 59],
       [60, 61, 62, 63, 64, 65, 66, 67, 68, 69],
       [70, 71, 72, 73, 74, 75, 76, 77, 78, 79],
       [80, 81, 82, 83, 84, 85, 86, 87, 88, 89],
       [90, 91, 92, 93, 94, 95, 96, 97, 98, 99]])
```

Only particular row is printed

In [75]:
```
1 mat[3]
```

```
array([30, 31, 32, 33, 34, 35, 36, 37, 38, 39])
```

In [76]:
```
1 mat[3,:]
```

```
array([30, 31, 32, 33, 34, 35, 36, 37, 38, 39])
```

Only particular column is printed

In [78]:
```
1 mat[:,4]
```

```
array([ 4, 14, 24, 34, 44, 54, 64, 74, 84, 94])
```

Particular element

In [80]:
```
1 mat[0,4]
```

```
4
```

In [83]:
```
1 mat[7,7]
```

```
77
```

In [85]:
```
mat[2:6,2:4] # 1:5 --> only row part /// 1:3 -- it indicates only column pa
```

```
array([[22, 23],
       [32, 33],
       [42, 43],
       [52, 53]])
```

In [86]:
```
mat[1:2,2:4]
```

```
array([[12, 13]])
```

In [87]:
```
mat[3:5,2:4]
```

```
array([[32, 33],
       [42, 43]])
```

In [88]:
```
mat[2:3,4:5]
```

```
array([[24]])
```

Masking

In [95]:
```
k = np.arange(0,16).reshape(4,4)
k
```

```
array([[ 0,  1,  2,  3],
       [ 4,  5,  6,  7],
       [ 8,  9, 10, 11],
       [12, 13, 14, 15]])
```

In [96]:
```
k < 10
```

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

In [101]:
```
a = k[k < 10]
a
```

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

In [98]:
```
k > 10
```

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

In [102]:
```
b = k [k > 10]
b
```

```
array([11, 12, 13, 14, 15])
```

In [ ]:
```

```