2nd july (import math module)

```
x=sqrt(25)        #sqrt is imbuild function
```

```
---------------------------------------------------------------------------
-----
NameError                                 Traceback (most recent call
last)
Cell In[1], line 1
----> 1 x=sqrt(25)

NameError: name 'sqrt' is not defined
```

```
import math    #math is module

x=math.sqrt(25)
x

x1=math.sqrt(15)
x1

print(math.floor(2.9))   ## floor -min or least value

print(math.ceil(2.9))   #ceil-max or hughest value

print(math.pow(3,2))

print(math.pi)   #these are constant

print(math.e)   #these are constant

import math as m
m.sqrt(10)

from math import sqrt,pow   #math has many function if you want to
call specific function then you use from
pow(2,3)

round(pow(2,3))

#help(math)

#pycharm run debug
# how to install python idle
#how to install pycharm & starts working on pycharm

x=input()
y=input()
z=x+y
print(z)
```

```
x1=input('Enter the 1st number')
y1=input('Enter the 2nd number')
z1=x1+y1
print(z1)

type(x1)
type(y1)

x1=input('Enter the 1st number')
a1=int(x1)
y1=input('Enter the 2nd number')
b1=int(y1)
z1=a1+b1
print(z1)

x2=int(input('Enter the 1st number'))
y2=int(input('Enter the 2nd number'))    #by using less memory than
compared to the previous code
z2=x2+y2
z2

ch=input('enter a char')
print(ch)

ch=input('enter a char')
print(ch)

print(ch[0])

print(ch[1])

print(ch[-1])

ch=input('enter a char')[0]
print(ch)

ch=input('enter a char')[1:3]
print(ch)

ch=input('enter a char')
print(ch)

ch=input('enter a char')
print(ch)

result=eval(input('enter an expr'))
print(result)
```

3rd july(class)

```
import numpy as np
```

```
np.__version__
```

```
'1.26.4'
```

##Creating arrays

```
my_list=[0,1,2,3,4,5]
my_list
```

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

```
type(my_list)
```

```
list
```

```
arr=np.array(my_list)
arr
```

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

```
type(arr)
```

```
numpy.ndarray
```

```
type(my_list)
```

```
list
```

```
print(type(arr))
print(type(my_list))
```

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

```
np.arange(10)
```

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

```
np.arange(10,20)
```

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

```
np.arange(10,50,5)
```

```
array([10, 15, 20, 25, 30, 35, 40, 45])
```

```
np.arange(10,30,3)
```

```
array([10, 13, 16, 19, 22, 25, 28])
```

```
np.arange(10,30,30,3)
```

```
-----------------------------------------------------------------------
-----
```

```
TypeError                                 Traceback (most recent call
last)
Cell In[16], line 1
----> 1 np.arange(10,30,30,3)

TypeError: Cannot interpret '3' as a data type
```

```
np.arange(8,20)
```

```
array([ 8,  9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19])
```

```
np.arange(20,8)
```

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

```
np.arange(-20,8)   #1st arg < 2nd arg
```

```
array([-20, -19, -18, -17, -16, -15, -14, -13, -12, -11, -10,  -9,  -
8,
        -7,  -6,  -5,  -4,  -3,  -2,  -1,   0,   1,   2,   3,   4,
5,
         6,   7])
```

```
n=np.arange(-20,8)
n
```

```
array([-20, -19, -18, -17, -16, -15, -14, -13, -12, -11, -10,  -9,  -
8,
        -7,  -6,  -5,  -4,  -3,  -2,  -1,   0,   1,   2,   3,   4,
5,
         6,   7])
```

```
np.zeros(3)
```

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

```
np.zeros(3,dtype=int)
```

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

```
z=np.zeros(5)
z
```

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

```
z=np.zeros(5,dtype=int)
z
```

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

```
z=np.zeros((2,2))    ##2d array
z
```

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

z=np.zeros((3,3),dtype=int)
z

array([[0, 0, 0],
       [0, 0, 0],
       [0, 0, 0]])

z=np.zeros((5,9),dtype=int)
z

array([[0, 0, 0, 0, 0, 0, 0, 0, 0],
       [0, 0, 0, 0, 0, 0, 0, 0, 0],
       [0, 0, 0, 0, 0, 0, 0, 0, 0],
       [0, 0, 0, 0, 0, 0, 0, 0, 0],
       [0, 0, 0, 0, 0, 0, 0, 0, 0]])

z=np.ones((5,9),dtype=int)
z

array([[1, 1, 1, 1, 1, 1, 1, 1, 1],
       [1, 1, 1, 1, 1, 1, 1, 1, 1],
       [1, 1, 1, 1, 1, 1, 1, 1, 1],
       [1, 1, 1, 1, 1, 1, 1, 1, 1],
       [1, 1, 1, 1, 1, 1, 1, 1, 1]])

z=np.ones(3)
z

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

z=np.ones((3,3),dtype=int)
z

array([[1, 1, 1],
       [1, 1, 1],
       [1, 1, 1]])

nd1=np.ones((10,10),dtype=int)
nd1

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

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

arr

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

4th

```
random.rand(2)

---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[33], line 1
----> 1 random.rand(2)

NameError: name 'random' is not defined

np.random.rand(2)

array([0.89416197, 0.97947215])

np.random.rand(3)

array([0.54322352, 0.61346882, 0.56633927])

np.random.rand(2,3)

array([[0.92168749, 0.30218734, 0.33974685],
       [0.34905877, 0.09676112, 0.98625311]])

np.random.rand(4,6)

array([[0.63505312, 0.64846398, 0.01510994, 0.10671652, 0.90166924,
        0.90136016],
       [0.55702141, 0.49493337, 0.01386852, 0.89041848, 0.78848032,
        0.43360333],
       [0.81022745, 0.74111541, 0.88367638, 0.43558094, 0.86108721,
        0.44628085],
       [0.16303457, 0.59649499, 0.00310971, 0.02291458, 0.33436904,
        0.62409376]])

np.random.randint(3)

2

np.random.randint(2,10)

4

np.random.randint(2,10,4)
```

```
array([7, 3, 4, 8])
```

```
m=np.random.randint(10,40,(10,10))   #with variable
m
```

```
array([[39, 39, 36, 18, 13, 25, 28, 20, 34, 38],
       [31, 28, 10, 13, 22, 14, 11, 28, 18, 39],
       [27, 29, 16, 16, 24, 31, 39, 20, 14, 17],
       [26, 14, 12, 15, 38, 35, 30, 33, 13, 35],
       [32, 13, 23, 22, 15, 17, 38, 19, 25, 39],
       [33, 20, 35, 10, 32, 38, 16, 23, 18, 10],
       [37, 28, 15, 19, 32, 29, 11, 26, 27, 24],
       [34, 24, 15, 29, 16, 35, 33, 13, 31, 26],
       [32, 27, 38, 21, 29, 23, 34, 33, 25, 13],
       [19, 36, 39, 29, 13, 26, 26, 19, 12, 13]])
```

```
np.random.randint(10,40,(10,10))      #without variable
```

```
array([[16, 32, 36, 13, 20, 35, 27, 10, 17, 29],
       [28, 21, 14, 28, 29, 17, 10, 19, 26, 29],
       [21, 16, 17, 26, 20, 16, 25, 34, 14, 16],
       [17, 29, 16, 28, 16, 30, 20, 35, 28, 16],
       [37, 32, 25, 18, 37, 17, 16, 20, 22, 15],
       [28, 24, 38, 28, 25, 14, 18, 35, 24, 11],
       [16, 28, 37, 30, 12, 26, 26, 18, 30, 34],
       [18, 19, 21, 11, 27, 33, 23, 26, 25, 27],
       [22, 35, 38, 35, 36, 28, 38, 31, 36, 24],
       [24, 32, 36, 30, 33, 38, 15, 33, 24, 33]])
```

```
arr
```

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

```
arr.reshape(2,3)
```

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

```
arr.reshape(3,3)
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Cell In[45], line 1
----> 1 arr.reshape(3,3)

ValueError: cannot reshape array of size 6 into shape (3,3)
```

```
arr.reshape(6,1)
```

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

```
arr.reshape(1,6)
```

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

```
b=np.random.randint(10,40,(5,4))
b
```

```
array([[28, 34, 25, 29],
       [10, 24, 32, 22],
       [21, 25, 28, 35],
       [31, 37, 11, 34],
       [34, 34, 19, 32]])
```

##slicing in matrix

```
b[:]
```

```
array([[28, 34, 25, 29],
       [10, 24, 32, 22],
       [21, 25, 28, 35],
       [31, 37, 11, 34],
       [34, 34, 19, 32]])
```

```
b[1:4]
```

```
array([[10, 24, 32, 22],
       [21, 25, 28, 35],
       [31, 37, 11, 34]])
```

```
b
```

```
array([[28, 34, 25, 29],
       [10, 24, 32, 22],
       [21, 25, 28, 35],
       [31, 37, 11, 34],
       [34, 34, 19, 32]])
```

```
b[-1:]
```

```
array([[34, 34, 19, 32]])
```

```
b[:-1]
```

```
array([[28, 34, 25, 29],
       [10, 24, 32, 22],
```

```
       [21, 25, 28, 35],
       [31, 37, 11, 34]])
```

```
b[1:4]
```

```
array([[10, 24, 32, 22],
       [21, 25, 28, 35],
       [31, 37, 11, 34]])
```

```
b[1,2]
```

```
32
```

```
b[1,3]
```

```
22
```

```
b[1,-1]
```

```
22
```

##Numoy operations

```
arr
```

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

```
arr.max()
```

```
5
```

```
arr.min()
```

```
0
```

```
arr.mean()
```

```
2.5
```

##Indexing

```
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]])

row=4
col=5

row

4

col

5

mat[row,col]   #it gives 4th row nd 5th col value

45

row=4
col=6

mat[row,col]

46

mat[1]   #printing rows

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

mat[:,col]    #how to print col from the matrix

array([ 6, 16, 26, 36, 46, 56, 66, 76, 86, 96])

mat[:,3]     #for print the 3rd col

array([ 3, 13, 23, 33, 43, 53, 63, 73, 83, 93])

mat[3]     3for print the 3rd row

<>:1: SyntaxWarning: invalid decimal literal
<>:1: SyntaxWarning: invalid decimal literal
C:\Users\Admin\AppData\Local\Temp\ipykernel_37380\557096412.py:1:
SyntaxWarning: invalid decimal literal
  mat[3]     3for print the 3rd row

  Cell In[73], line 1
    mat[3]     3for print the 3rd row
                          ^
SyntaxError: invalid decimal literal


mat[::-1]    #reverse the matrix
```

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

```
mat[::-2]   #reverse the matrix with step count 2
```

```
array([[90, 91, 92, 93, 94, 95, 96, 97, 98, 99],
       [70, 71, 72, 73, 74, 75, 76, 77, 78, 79],
       [50, 51, 52, 53, 54, 55, 56, 57, 58, 59],
       [30, 31, 32, 33, 34, 35, 36, 37, 38, 39],
       [10, 11, 12, 13, 14, 15, 16, 17, 18, 19]])
```

```
mat[2:6,2:4]
```

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

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

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

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

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

##Masking or filter

```
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]])
```

```
mat>50
```

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

```
mat[mat>50]
```

```
array([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])
```

```
mat[mat>=50]
```

```
array([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])
```

```
mat[mat==50]
```

```
array([50])
```

```
mat[mat<50]
```

```
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])
```

```
mat[mat<=50]
```

```
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])
```

```
mat[mat!=50]
```

```
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,
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])
```

```
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]])
```

```
mat==50
```

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

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

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]])

a1=mat[mat<50]
a1

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])

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]])

a2=mat[mat>50]
a2

array([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])

a3=mat[mat<=50]
a3

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])

a4=mat[mat==50]
a4

array([50])
```

##python program to generate otp

```
import random
def generate_otp(length=4):    #generate a numeric otp of a specified
length
    digits= '012345'
    otp=''.join(random.choice(digits) for _ in range(length))
    return otp
#Example usage
otp_length=4     #you can change this to any length you prefer
otp=generate_otp(otp_length)
print(f"Your OTP id: {otp}")

Your OTP id: 0042

def wish():
    print('good even')
wish()
def wish():
    print('good even')
wish()
def wish():
    print('good even')
wish()

good even
good even
good even

def wish():
    print('good even')
wish()
```

```
wish()
wish()
```

```
good even
good even
good even
```

```
list1=['a','b','g',1,5]
print(list1.pop)
```

```
<built-in method pop of list object at 0x000001AC4B5C3E40>
```

```
x=[1,2,3]
y=x.copy()
x.append(4)
print(x)
```

```
[1, 2, 3, 4]
```