

```
In [1]: import math
a=16
print(math.sqrt(a))

4.0
```

```
In [2]: from math import sin, sqrt
```

```
In [3]: c=25
d=3.10
```

```
In [4]: print(sqrt(c))

5.0
```

```
In [5]: print(sqrt(d))

1.760681686165901
```

```
In [14]: distance = [10,15,17,26]
time = [.30,47,55,1]
```

```
In [15]: import numpy as np
```

```
In [16]: np_distance=np.array(distance)
```

```
In [17]: np_time=np.array(time)
```

```
In [18]: speed= np_distance/np_time
```

```
In [19]: print(speed)

[33.33333333  0.31914894  0.30909091 26.          ]
```

```
In [20]: import numpy as np
```

```
In [22]: x=np.arange (12)
```

```
In [24]: y=np.reshape(x, (4,3))
x
y
```

```
Out[24]: array([[ 0,  1,  2],
               [ 3,  4,  5],
               [ 6,  7,  8],
               [ 9, 10, 11]])
```

```
In [25]: import numpy as np
```

```
In [26]: a= np.array([[1,2],[3,4]])
```

```
In [28]: print (a)

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

```
In [29]: b= np.array([[5,6],[7,8]])
```

```
In [30]: print (b)
```

```
[[5 6]
 [7 8]]
```

In [31]:

```
[[ 6  8]
 [10 12]]
```

In [37]:

```
import numpy as np
print (np.char.add(['hello'], [' world']))
print('\n')
print (np.char.add(['hello', 'hi'], ['world', 'there']))
```

Input In [37]

```
print (np.char.add(['hello', 'hi'], ['world', 'there']))
```

**SyntaxError:** unexpected EOF while parsing

In [35]:

```
#This function performs elementwise string concatenation.
import numpy as np
print ('Concatenate two strings:')
print (np.char.add(['hello'], [' world']))
print ('\n')
print ('Concatenation elementwise:')
print (np.char.add(['hello', 'hi'], [' world', ' there']))
```

Concatenate two strings:  
['hello world']

Concatenation elementwise:  
['hello world' 'hi there']

In [39]:

```
#numpy.mod()
#This function returns the remainder of division of the corresponding elements in
#The function numpy.remainder() also produces the same result.
import numpy as np
a = np.array([10,20,30])
b = np.array([3,5,7])
print( 'First array:')
print( a)
print( '\n')
print( 'Second array:')
print( b)
print( '\n')
print( 'Applying mod() function:')
print( np.mod(a,b))
print( '\n')
print( 'Applying remainder() function:')
print( np.remainder(a,b))
```

First array:  
[10 20 30]

Second array:  
[3 5 7]

Applying mod() function:  
[1 0 2]

Applying remainder() function:  
[1 0 2]

```
In [40]: #This function treats elements in the first input array as base
#and returns it raised to the power of the corresponding element in the second input array
import numpy as np
a = np.array([10,100,1000])
print( 'Our array is:')
print(a)
print('\n')
print('Applying power function:')
print(np.power(a,2))
print('\n')
print('Second array:')
b = np.array([1,2,3])
print(b)
print('\n')
print('Applying power function again:')
print(np.power(a,b))
```

Our array is:  
[ 10 100 1000]

Applying power function:  
[ 100 10000 1000000]

Second array:  
[1 2 3]

Applying power function again:  
[ 10 10000 1000000000]

```
In [41]: #numpy.median()
#Median is defined as the value separating the higher half of a data sample from the lower half
#The numpy.median() function is used as shown in the following program.
import numpy as np
a = np.array([[30,65,70],[80,95,10],[50,90,60]])
print('Our array is:')
print(a)
print('\n')
print('Applying median() function:')
print(np.median(a))
print('\n')
print('Applying median() function along axis 0:')
print(np.median(a, axis = 0))
print('\n')

print('Applying median() function along axis 1:')
print(np.median(a, axis = 1))
```

```
Our array is:  
[[30 65 70]  
 [80 95 10]  
 [50 90 60]]
```

```
Applying median() function:  
65.0
```

```
Applying median() function along axis 0:  
[50. 90. 60.]
```

```
Applying median() function along axis 1:  
[65. 80. 60.]
```

```
In [42]: import numpy as np  
a = np.array([[1,2,3],[3,4,5],[4,5,6]])  
print('Our array is:')  
print(a)  
print('\n')  
print('Applying mean() function:')  
print(np.mean(a))  
print('\n')
```

```
Our array is:  
[[1 2 3]  
 [3 4 5]  
 [4 5 6]]
```

```
Applying mean() function:  
3.6666666666666665
```

```
In [43]: import numpy as np  
a = np.array([1,2,3,4])  
print('Our array is:')  
print(a)  
print('\n')  
print('Applying average() function:')  
print(np.average(a))  
print('\n')
```

```
Our array is:  
[1 2 3 4]
```

```
Applying average() function:  
2.5
```

```
In [44]: import numpy as np  
import pandas as pd
```

```
-----
ModuleNotFoundError                                Traceback (most recent call last)
Input In [44], in <cell line: 2>()
      1 import numpy as np
----> 2 import panda as pd

ModuleNotFoundError: No module named 'panda'
```

```
In [45]: import numpy as np
```

```
In [46]: import pandas as pd
```

```
In [48]: #Create Dataframe from a Dict with equal length lists
#we create a dataframe with last five olympics data which has place,year,number of
olympic_data_list = {'City':['London','Beijing','Athens','Sydney','Atlanta'],
                     'Year':[2012,2008,2004,2000,1996],
                     'No of Participating Countries':[205,204,201,200,197]}
}
```

```
In [53]: import numpy as np
```

```
In [54]: import pandas as pd
```

```
In [55]: df_olympic_data = pd.DataFrame(olympic_data_list)
```

```
In [57]: olympic_data_dict= {'London':{2012:205},'Beijing': { 2008:204}}
```

```
In [52]: df.olympic_data=pd.DataFrame(olympic_data_dict)
```

```
In [ ]:
```

```
In [ ]:
```

```
In [60]: import pandas as pd
import numpy as np
```

```
In [61]: df = pd.read_csv()
df.head()
```

```
-----
TypeError                                Traceback (most recent call last)
Input In [61], in <cell line: 1>()
----> 1 df = pd.read_csv()
      2 df.head()

File ~\Anaconda3\lib\site-packages\pandas\util\_decorators.py:311, in deprecate_no
nkeyword_arguments.<locals>.decorate.<locals>.wrapper(*args, **kwargs)
    305 if len(args) > num_allow_args:
    306     warnings.warn(
    307         msg.format(arguments=arguments),
    308         FutureWarning,
    309         stacklevel=stacklevel,
    310     )
--> 311 return func(*args, **kwargs)

TypeError: read_csv() missing 1 required positional argument: 'filepath_or_buffer'
```

```
In [63]: import numpy as np
import matplotlib.pyplot as plt
```

```
from matplotlib import style
%matplotlib inline
```

```
In [65]: randomnumber = np.random.rand(10)
```

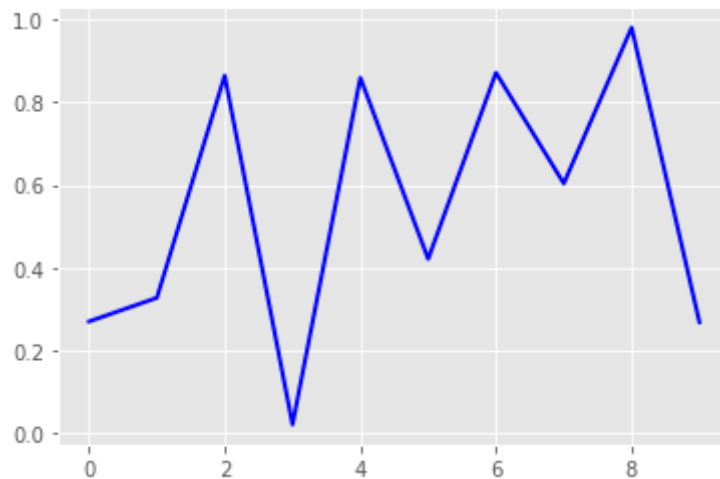
```
In [66]: randomnumber
```

```
Out[66]: array([0.27060835, 0.32778016, 0.86416482, 0.02176171, 0.85913182,
0.42172273, 0.87122369, 0.60389822, 0.98016679, 0.26909165])
```

```
In [71]: style.use('ggplot')
```

```
In [73]: plt.plot(randomnumber,'b',label= 'line one', linewidth=2 )
```

```
Out[73]: [matplotlib.lines.Line2D at 0x20780062dc0>]
```



```
In [1]: x,y = 2, 22
```

```
In [3]: x
```

```
Out[3]: 2
```

```
In [4]: y
```

```
Out[4]: 22
```

```
In [5]: word = " Encyclopedia "
```

```
In [6]: "t" in word
```

```
Out[6]: False
```

```
In [7]: num1 = (45,73,56)
num2 = (34,55,77)
num1+num2
```

```
Out[7]: (45, 73, 56, 34, 55, 77)
```

```
In [10]: def function1():
    #print("Hello you are in function1")

    function1()
```

```
In [11]: num_list = range(15)
```

```
In [12]: list(reversed(num_list))
```

```
Out[12]: [14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0]
```

```
In [13]: import numpy as np
         from numpy.random import randn
```

```
In [15]: randn()
```

```
Out[15]: -0.7880054379847926
```

```
In [42]: x=randn()
         if x>1:
             answer="greater than 1"
         print(x)
         print(answer)
```

```
2.8437079970262196
greater than 1
```

```
In [51]: answer = None
         x=randn()
         if x>1:
             answer="greater than 1"
         else:
             answer="less than 1"
         print(x)
         print(answer)
```

```
-1.148423426866844
less than 1
```

```
In [52]: for i in range (5):
         print("Hello Python")
```

```
Hello Python
Hello Python
Hello Python
Hello Python
Hello Python
```

```
In [55]: for j in range(6):
         print(j)
```

```
0
1
2
3
4
5
```

```
In [56]: for i in range(5):
         print(" Hello Python: ",i)
```

```
Hello Python: 0
Hello Python: 1
Hello Python: 2
Hello Python: 3
Hello Python: 4
```

```
In [57]: mylist=[10,100,1000]
mylist
```

```
Out[57]: [10, 100, 1000]
```

```
In [58]: for i in 'simplilearn':
          print (i)
```

```
s
i
m
p
l
i
l
e
a
r
n
```

```
In [63]: subjects = ["History", "Geography", "Chemistry", "Mathematics","Economics"]
for i in subjects:
    print(i)
    if i == "Mathematics":
        break
```

```
History
Geography
Chemistry
Mathematics
```

```
In [65]: subjects = ["History", "Geography", "Chemistry", "Mathematics","Economics"]
for i in subjects:
    #print(i)
    if i == "Mathematics":
        break
    print(i)
```

```
History
Geography
Chemistry
```

```
In [66]: subjects = ["History", "Geography", "Chemistry", "Mathematics","Economics"]
for i in subjects:
    if i == "Mathematics":
        continue
    print (i)
```

```
History
Geography
Chemistry
Economics
```

```
In [68]: for x in range (8):
          print (x)
else:
    print("Loop exit")
```



```
0
1
2
3
4
5
6
7
Loop exit
```

```
In [69]: for x in range (8):
        if x==5:break
        print(x)
    else:
        print("done")
```

```
0
1
2
3
4
```

```
In [76]: subjects=["His","geo","sci","math","eco"]
        for i in subjects:
            print(i)
            if i == "math":
                break
```

```
His
geo
sci
math
```

```
In [74]: subjects = ["His","Geo","Sci","Math","Eco"]
        for i in subjects:
            print(i)
            if i=="Math":
                break
```

```
His
Geo
Sci
Math
```

```
In [78]: while False:
        print("hello")
```

```
In [3]: counter=0
        while counter<12:
            print(counter)
            counter=counter+1
```

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

```
In [4]: x=0
while (x<6):
    print("The count is: ",x)
    x=x+1
print("looping done")
```

```
The count is: 0
The count is: 1
The count is: 2
The count is: 3
The count is: 4
The count is: 5
looping done
```

```
In [5]: count=0
while count<5:
    print(count, " is less than 5")
    count=count+1
else:
    print(count, " is not less than 5")
```

```
0 is less than 5
1 is less than 5
2 is less than 5
3 is less than 5
4 is less than 5
5 is not less than 5
```

```
In [5]: import numpy as np
first_trial_cyclist = [ 10,15,17,26]
```

```
In [6]: second_trial_cyclist = [12,11,21,24]
```

```
In [7]: np_first_trial_cyclist = np.array(first_trial_cyclist)
np_second_trial_cyclist = np.array(second_trial_cyclist)
```

```
In [8]: np_first_trial_cyclist+np_second_trial_cyclist
```

```
Out[8]: array([22, 26, 38, 50])
```

```
In [ ]:
```