import numpy as np

b=[1,2,3]

a=np.array(b)

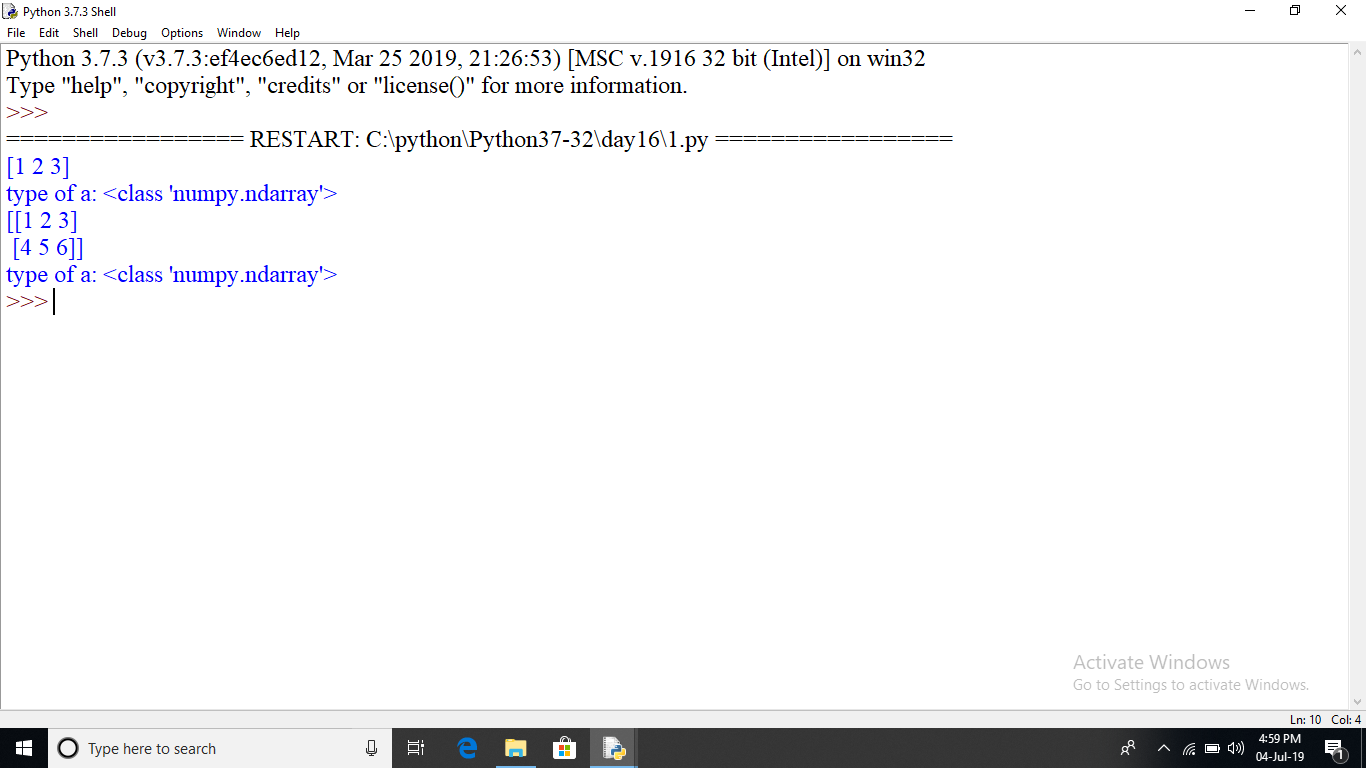
print(a)

print("type of a:",type(a))

a=np.array([(1,2,3),(4,5,6)])

print(a)

print("type of a:",type(a))



#Numpy v/s List

#less memory,fast,convenient

import numpy as np

import sys

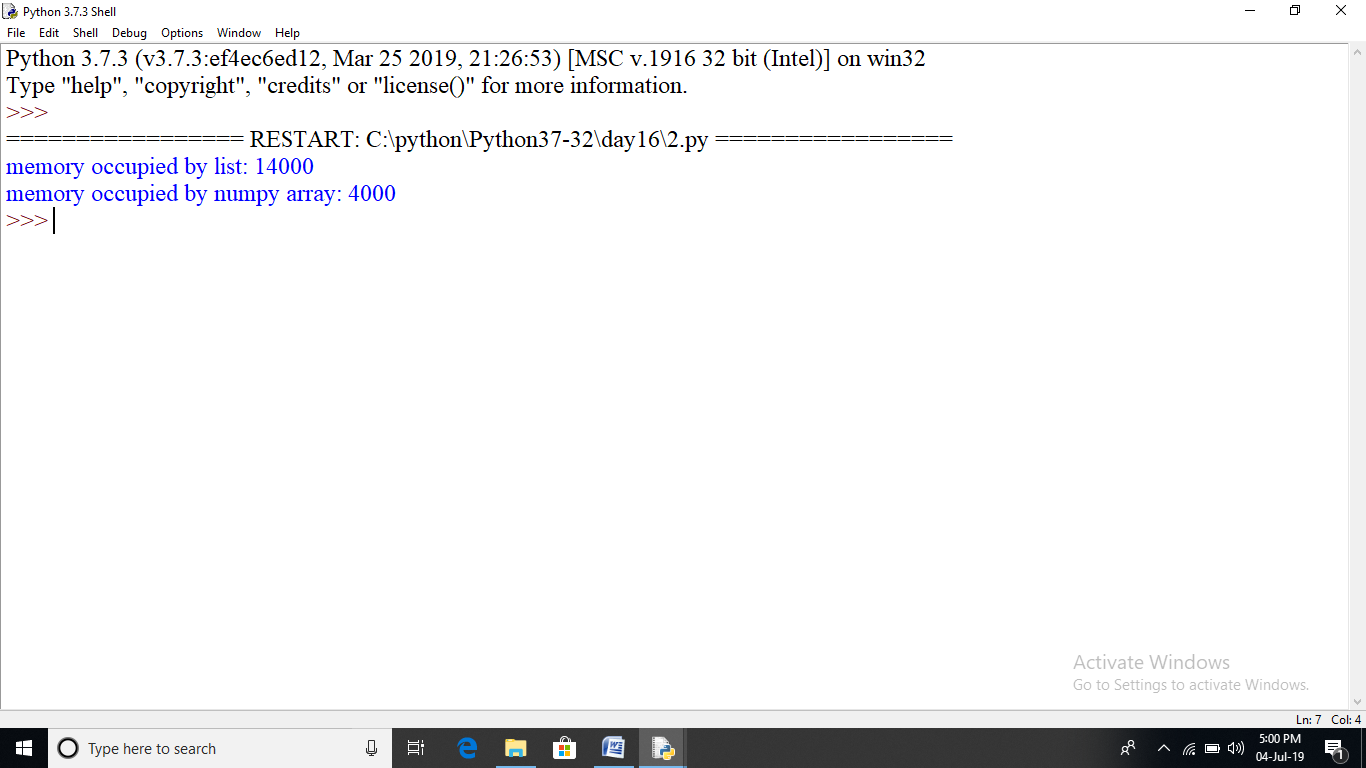
S=list(range(1000))

#memory occupied by one element\*size of list

print("memory occupied by list:",sys.getsizeof(999)\*len(S))

D=np.arange(1000)

print("memory occupied by numpy array:",D.itemsize\*D.size)



#Numpy v/s List

#less memory,fast,convenient

import numpy as np

import time

import sys

SIZE=10000000

L1=list(range(SIZE))

L2=list(range(SIZE))

start=time.time()

result=[(x+y)for x,y in zip(L1,L2)]

print("time taken by the list operation:",(time.time()-start)\*100000)

#numpy array

A1=np.arange(SIZE)

A2=np.arange(SIZE)

start=time.time()

result=A1+A2

print("time taken by numpy array operation:", (time.time()-start)\*100000)



import numpy as np

a=np.array([1,2,3])

print(a.ndim)

a=np.array([(1,2,3),(2,3,4)])

print(a.ndim)

print(a.itemsize)

print(a.dtype)

a=np.array([1,2,3])

print(a.size)

a=np.array([1,2,3,4,5,6,7])

print(a.size)

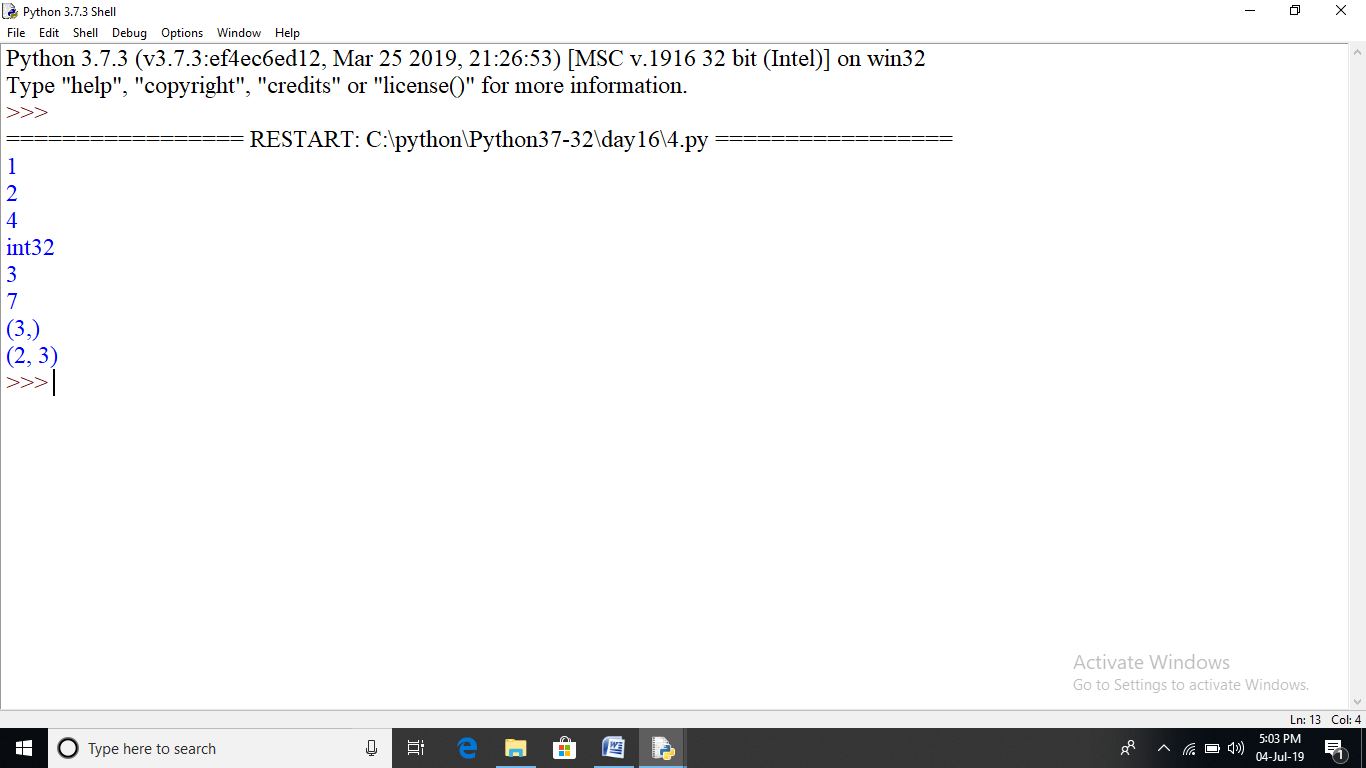
#shape of array

a=np.array([1,2,3])

print(a.shape)

a=np.array([(1,2,3),(2,3,4)])

print(a.shape)



import numpy as np

a=np.array([(1,2,3,4),(3,4,5,6)])

print(a)

a=a.reshape(4,2)

print(a)

a=np.array([(1,2,3,4),(13,14,15,16)])

print(a[0,2])

print(a[1,2])

print(a[0:,3])

a=np.array([(1,2,3,4),(13,14,15,16),(21,22,23,24)])

print('out:',a[0:3])

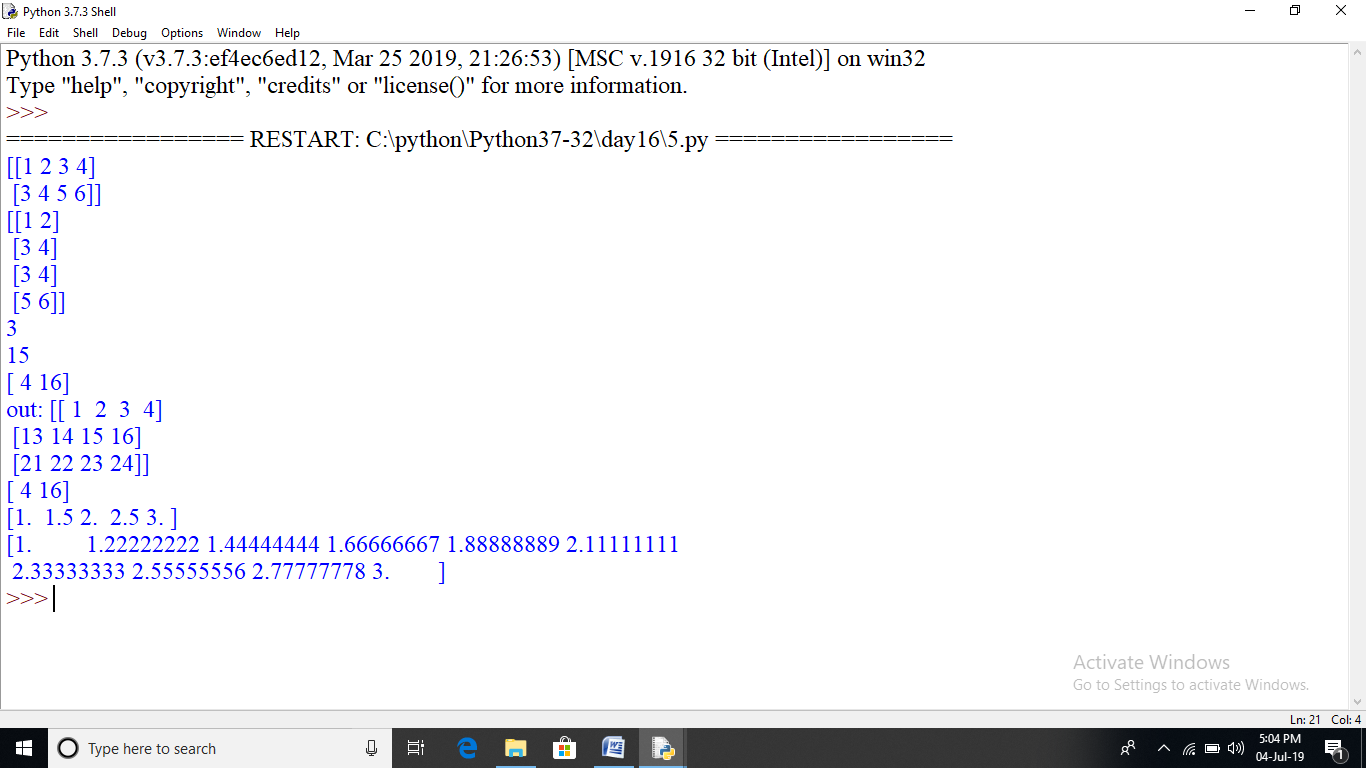
print(a[0:2,3])

a=np.linspace(1,3,5)

print(a)

a=np.linspace(1,3,10)

print(a)



#numpy operation

#max,min,sum

import numpy as np

a=np.array([1,2,3,4])

print("min:",a.min())

print("max:",a.max())

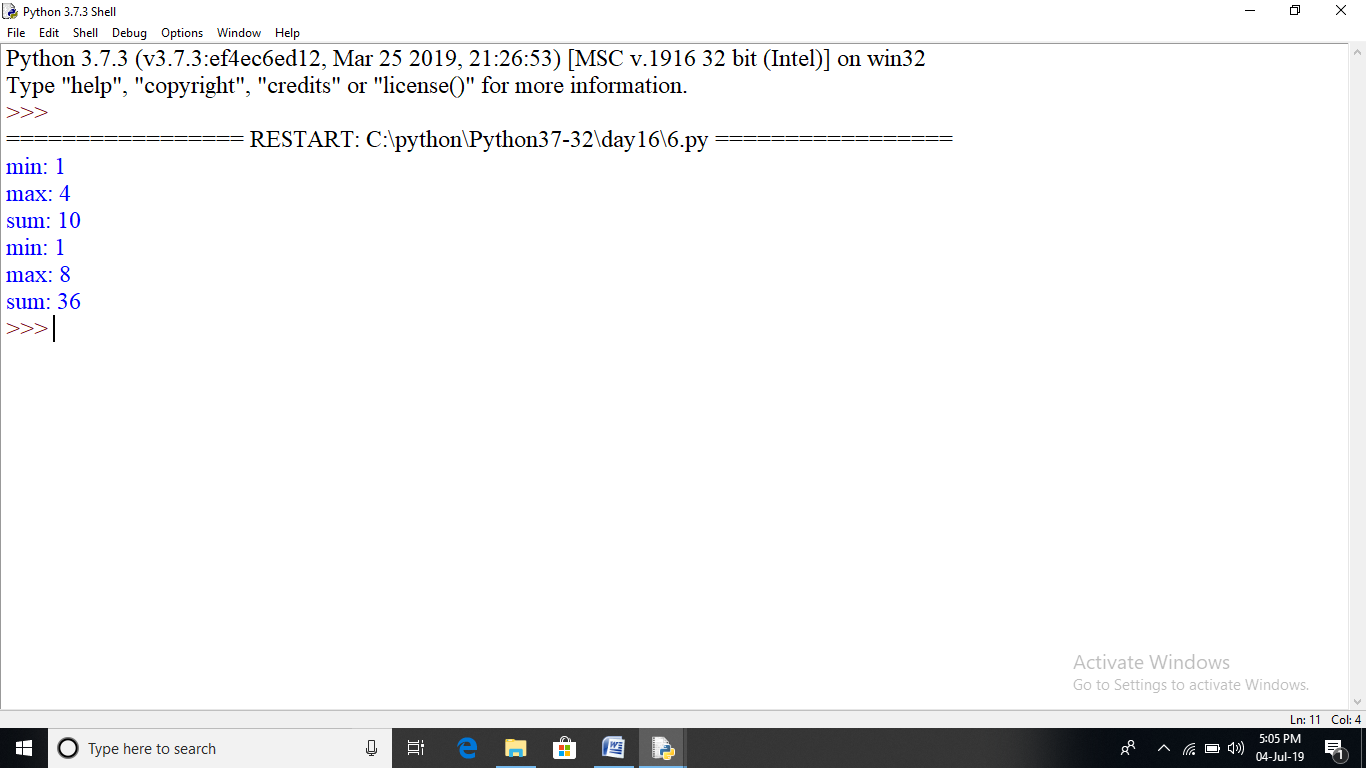
print("sum:",a.sum())

a=np.array([(1,2,3,4),(5,6,7,8)])

print("min:",a.min())

print("max:",a.max())

print("sum:",a.sum())



#numpy operation

#axis

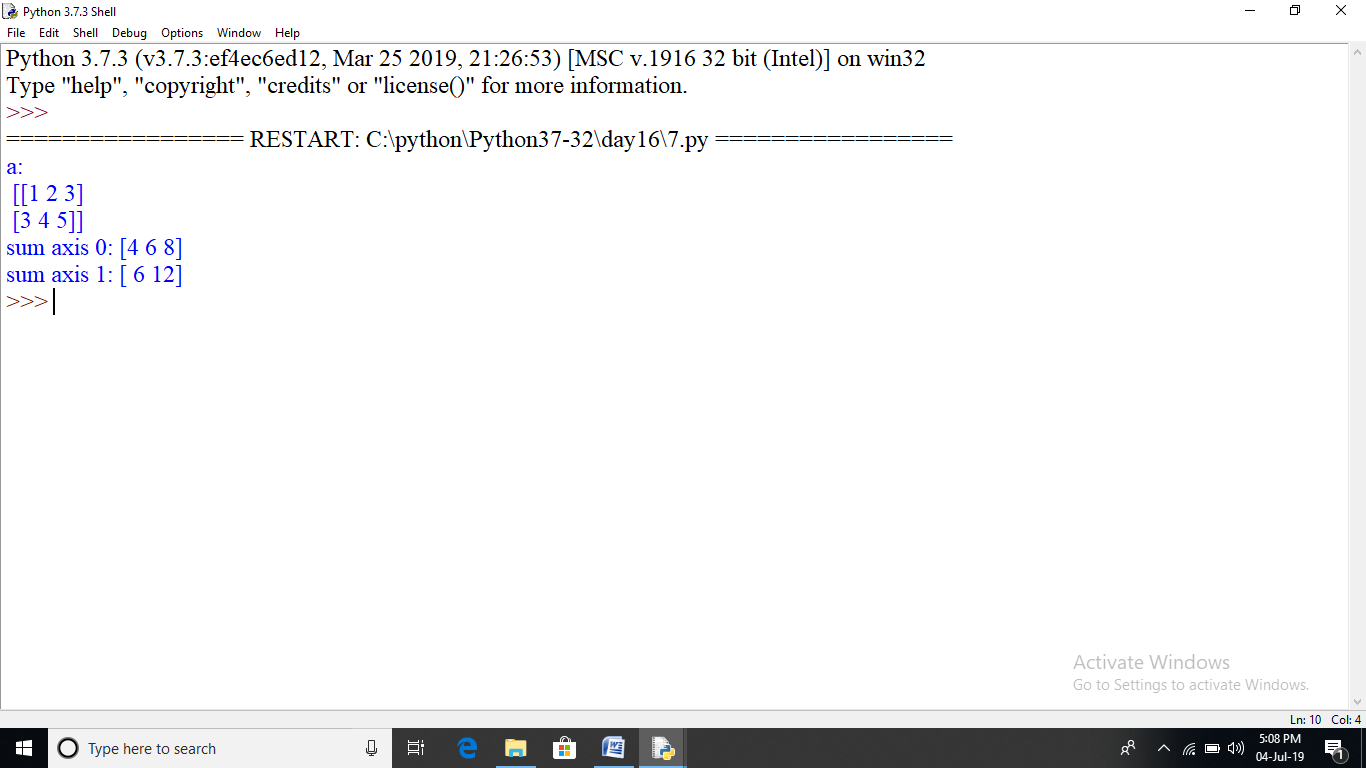
import numpy as np

a=np.array([(1,2,3),(3,4,5)])

print("a:\n",a)

print("sum axis 0:",a.sum(axis=0))

print("sum axis 1:",a.sum(axis=1))



#numpy operation

#squareroot,standard deviation

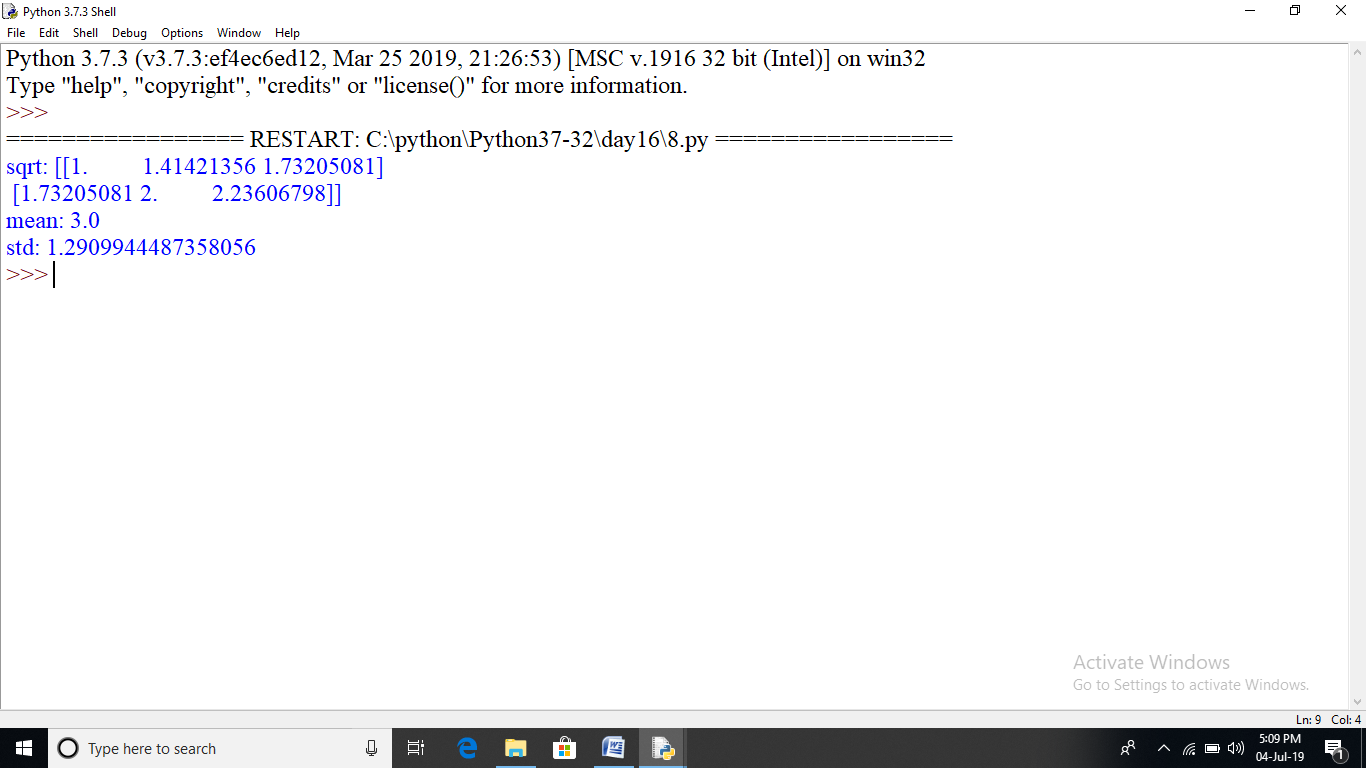
import numpy as np

a=np.array([(1,2,3),(3,4,5)])

print("sqrt:",np.sqrt(a))

print("mean:",np.mean(a))

print("std:",np.std(a))



#numpy operation

#matrix operation

import numpy as np

a=np.array([(1,2,3),(3,4,5)])

b=np.array([(1,2,3),(3,4,5)])

print(a+b)

print(a-b)

print(a\*b)

print(a/b)

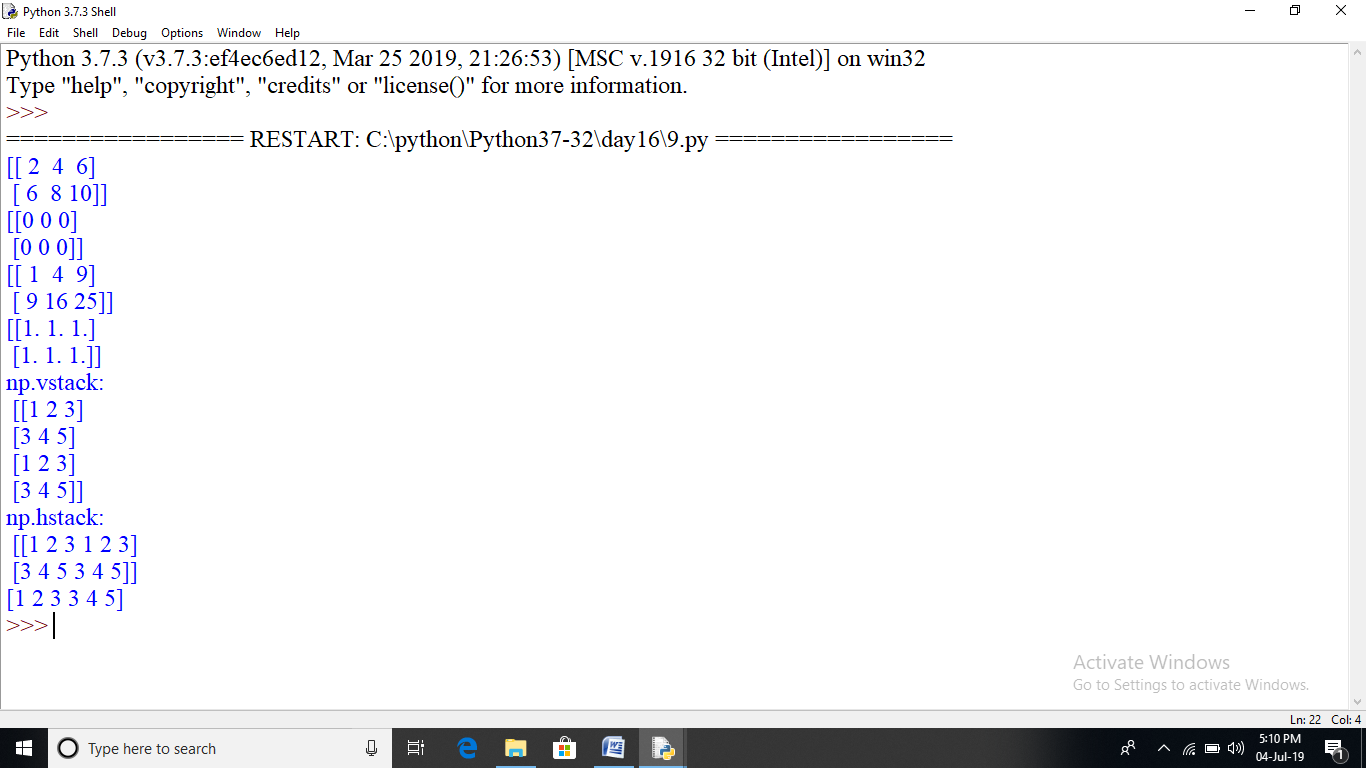
#concatenate

#vertical stacking

print("np.vstack:\n",np.vstack((a,b)))

print("np.hstack:\n",np.hstack((a,b)))

print(a.ravel())



#numpy operation

#sin,cos,tan

import numpy as np

import matplotlib.pyplot as plt

##x=np.arange(-1,3\*np.pi,0.1)

x=([1,2,3,4,5,6,7,8,9,10])

print("X:\n",x)

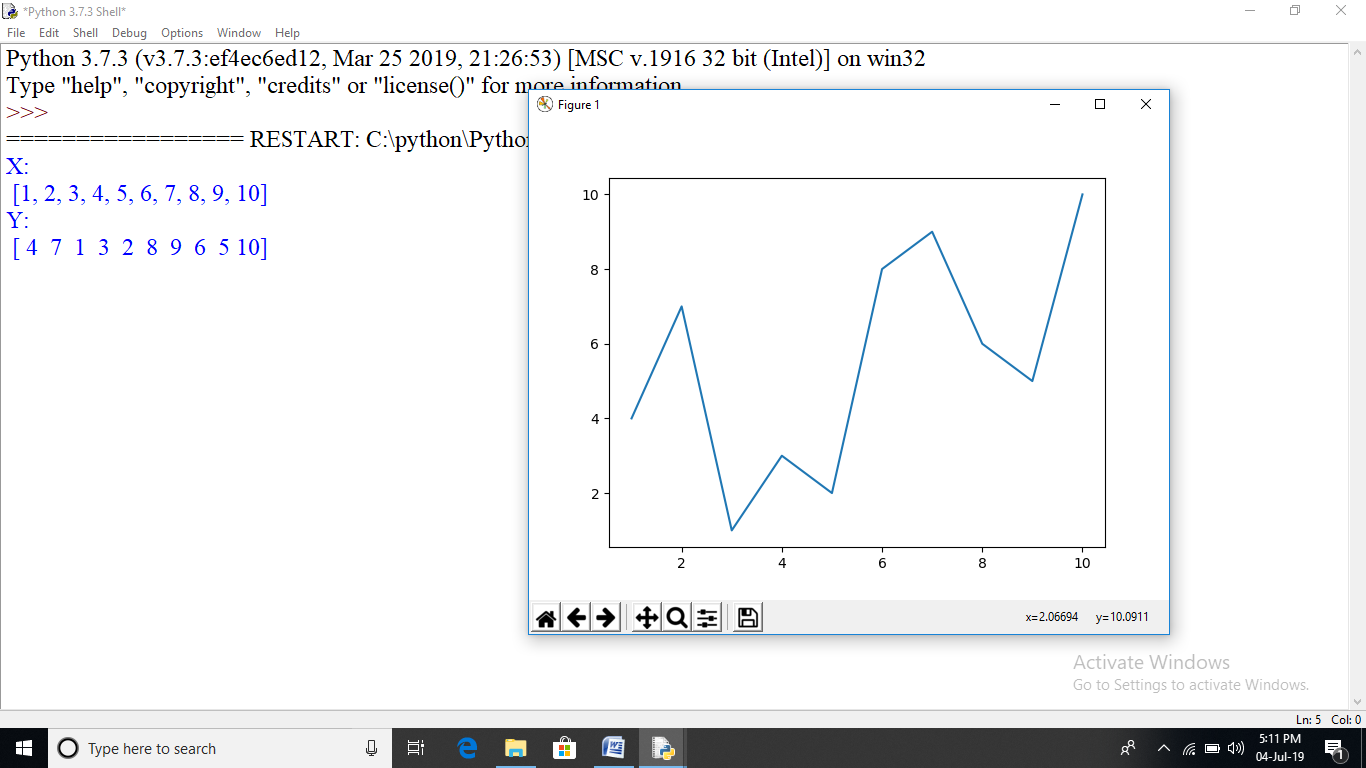
y=np.array([4,7,1,3,2,8,9,6,5,10])

##y=np.sin(x)

print("Y:\n",y)

plt.plot(x,y)

plt.show()



#numpy operation

import numpy as np

import matplotlib.pyplot as plt

objects=('python','c++','java','perl','scala','lisp')

x\_pos=np.arange(len(objects))

performance=[10,3,6,8,2,7]

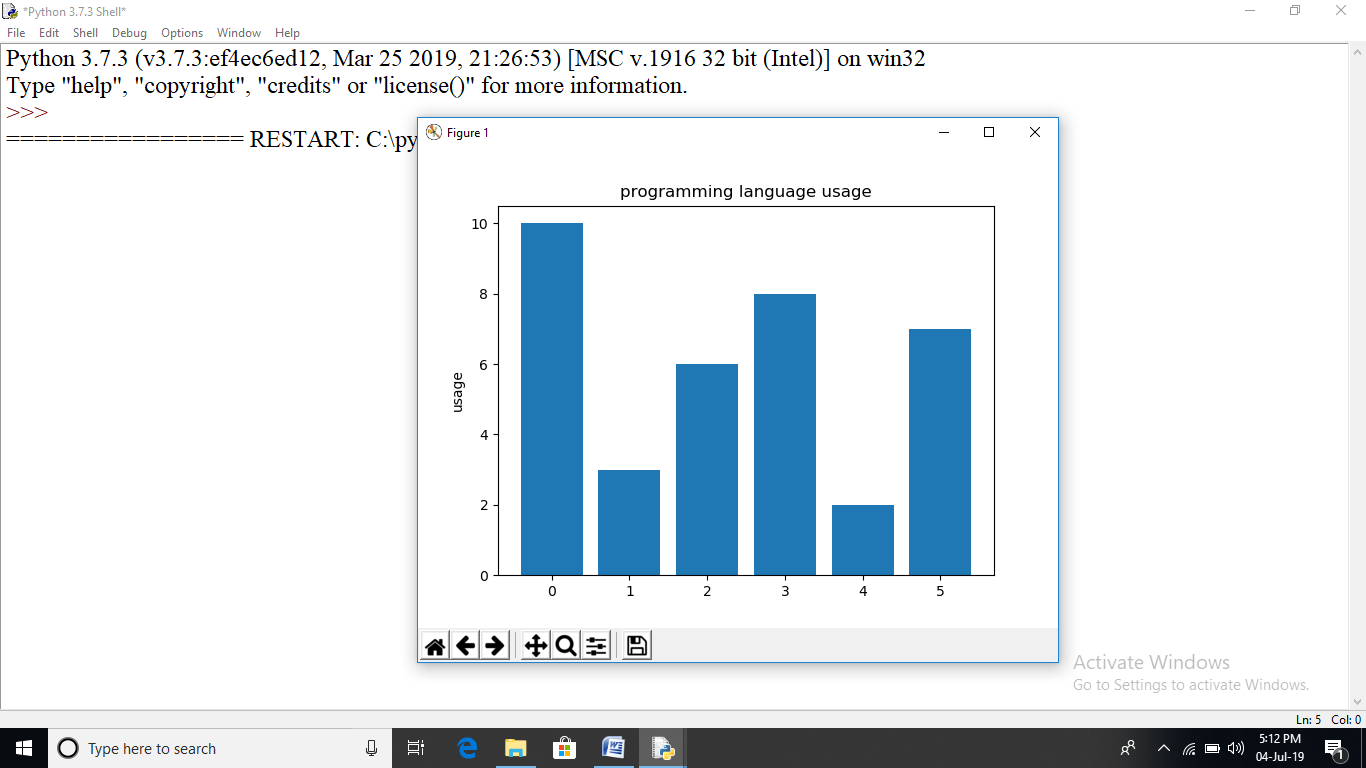
plt.bar(x\_pos,performance,align='center')

##plt.xticks(x\_pos,objects)

plt.ylabel('usage')

plt.title('programming language usage')

plt.show()



#numpy operation

import numpy as np

import matplotlib.pyplot as plt

objects=('python','c++','java','perl','scala','lisp')

y\_pos=np.arange(len(objects))

performance=[10,8,6,4,2,1]

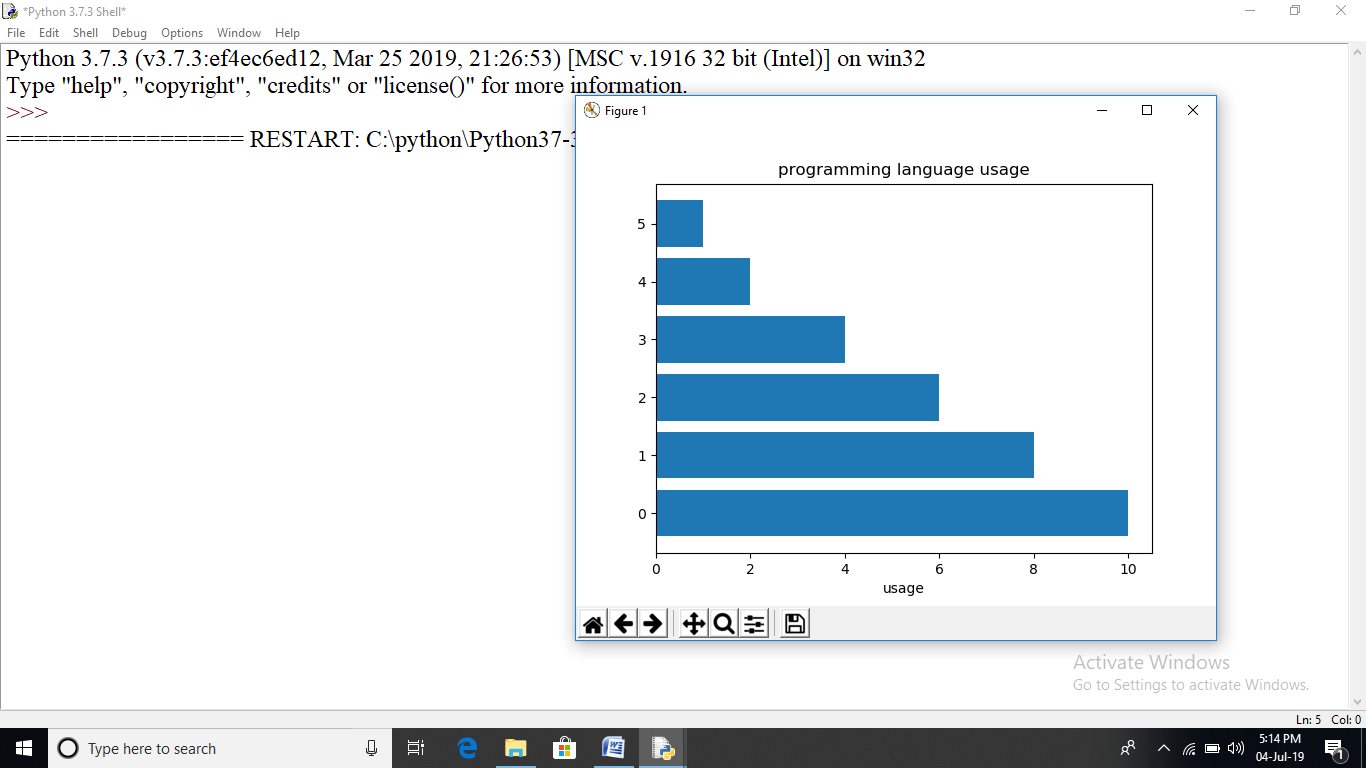
plt.barh(y\_pos,performance,align='center')

##plt.xticks(x\_pos,objects)

plt.xlabel('usage')

plt.title('programming language usage')

plt.show()



import matplotlib.pyplot as plt

#data to plot

labels=['python','c++','ruby','java']

sizes=[215,130,245,210]

colors=['gold','yellowgreen','lightcoral','lightskyblue']

explode=(0.1,0,0,0)

#plot

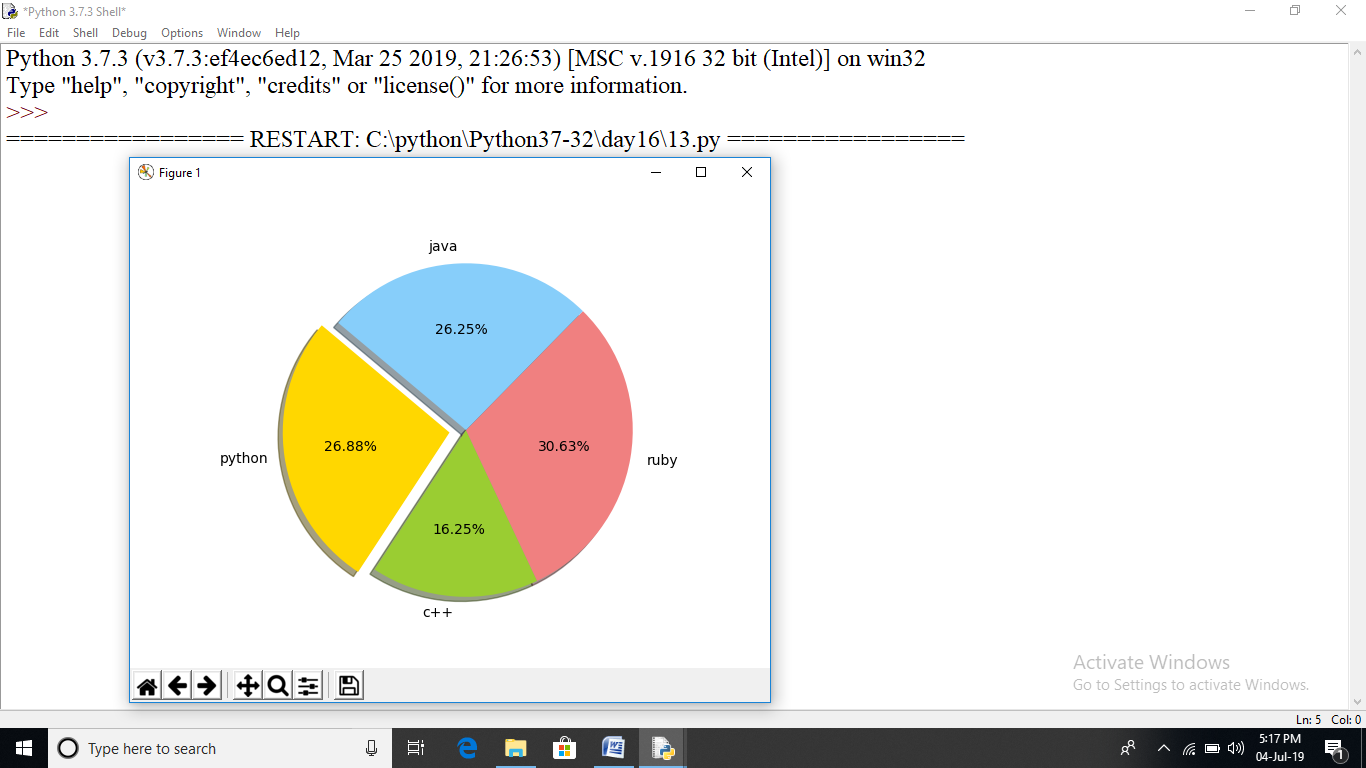
plt.pie(sizes,explode=explode,labels=labels,

colors=colors,

autopct='%1.2f%%',shadow=True,startangle=140)

plt.axis('equal')

plt.show()



import numpy as np

import matplotlib.pyplot as plt

#create data

N=500

x=np.random.rand(N)

y=np.random.rand(N)

colors=(0,1,0)

area=np.pi\*3

#plot

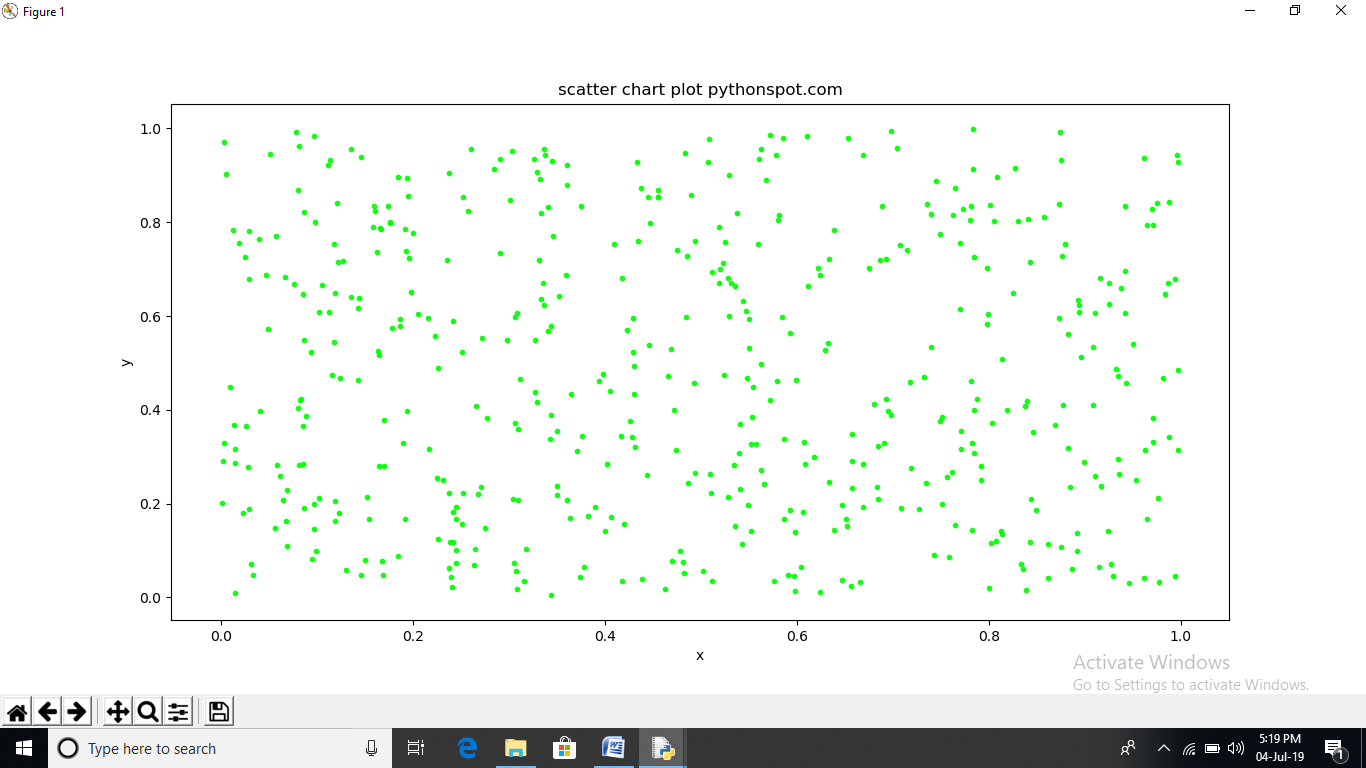
plt.scatter(x,y,s=area,c=colors,alpha=0.9)

plt.title("scatter chart plot pythonspot.com")

plt.xlabel('x')

plt.ylabel('y')

plt.show()



import numpy as np

import matplotlib.pyplot as plt

y=[2,4,6,8,10,12,14,16,18,20]

y2=[10,11,12,13,14,15,16,17,18,19]

x=np.arange(10)

fig=plt.figure()

ax=plt.subplot(111)

ax.plot(x,y,label='$y=numbers')

ax.plot(x,y2,label='$y2=other numbers')

plt.title('legend outside')

chartBox=ax.get\_position()

ax.set\_position([chartBox.x0,chartBox.y0,chartBox.width\*0.6,chartBox.height])

ax.legend(loc='upper center',bbox\_to\_anchor=(1.45,0.8),shadow=True,ncol=1)

plt.show()

