Numpy cont.

In [1]:
```
import numpy as np

    #create an array
arr = np.array ([[1,2,3,12],[4,5,6,11],[7,8,9,10]])
```

In [2]:
```
arr.shape
```

Out[2]:
```
(3, 4)
```

In [3]:
```
arr
```

Out[3]:
```
array([[ 1,  2,  3, 12],
       [ 4,  5,  6, 11],
       [ 7,  8,  9, 10]])
```

In [4]:
```
#reverse the rows

arr2 = arr[::-1,]
```

In [5]:
```
arr2
```

Out[5]:
```
array([[ 7,  8,  9, 10],
       [ 4,  5,  6, 11],
       [ 1,  2,  3, 12]])
```

In [6]:
```
#reverse the rows and columns

arr2 = arr[::-1,::-1]
```

In [7]:
```
arr2
```

Out[7]:
```
array([[10,  9,  8,  7],
       [11,  6,  5,  4],
       [12,  3,  2,  1]])
```

displaying the items of an array

In [8]:
```
arr [0:1,]
```

Out[8]:
```
array([[ 1,  2,  3, 12]])
```

In [9]:
```
arr [0:2,]
```

Out[9]:
```
array([[ 1,  2,  3, 12],
       [ 4,  5,  6, 11]])
```

In [10]:
```
arr [:]
```

Out[10]:
```
array([[ 1,  2,  3, 12],
       [ 4,  5,  6, 11],
       [ 7,  8,  9, 10]])
```

In [11]:
```
arr [::]
```

Out[11]:
```
array([[ 1,  2,  3, 12],
       [ 4,  5,  6, 11],
       [ 7,  8,  9, 10]])
```

In [12]:
```
#reversing only columns
arr [:,::-1]
```

Out[12]:
```
array([[12,  3,  2,  1],
       [11,  6,  5,  4],
       [10,  9,  8,  7]])
```

In [13]:
```
#extracting specific elements in an array
arr [0,2]
```

Out[13]:
```
3
```

In [14]:
```
arr [0,0]
```

Out[14]:
```
1
```

In [15]:
```
arr [0,::-1]
```

Out[15]:
```
array([12,  3,  2,  1])
```

In [16]:
```
arr [1,::-1]
```

Out[16]:
```
array([11,  6,  5,  4])
```

In [17]:
```
arr [2,::-1]
```

Out[17]:
```
array([10,  9,  8,  7])
```

In [18]:
```
arr [::-1,0]
```

Out[18]:
```
array([7, 4, 1])
```

In [19]:
```
arr [::-1,0]
```

Out[19]:
```
array([7, 4, 1])
```

In [20]:
```
#for all rows, etract column
arr [:,0:2]
```

Out[20]:
```
array([[1, 2],
       [4, 5],
       [7, 8]])
```

In [21]:
```
#for columns, etract rows
arr [0:2,:]
```

Out[21]:
```
array([[ 1,  2,  3, 12],
       [ 4,  5,  6, 11]])
```

In [22]:
```
arr [:,0:-1]
```

Out[22]:
```
array([[1, 2, 3],
       [4, 5, 6],
       [7, 8, 9]])
```

In [23]:
```
#first 2 rows, first 2 columns
arr [0:2,0:2]
```

Out[23]:
```
array([[1, 2],
       [4, 5]])
```

In [24]:
```
#replacing specfic array elements
arr [1,2] = 12
```

In [25]:
```
arr
```

Out[25]:
```
array([[ 1,  2,  3, 12],
       [ 4,  5, 12, 11],
       [ 7,  8,  9, 10]])
```

In [26]:
```
arr [2,2] = arr[1,1]
```

In [27]:
```
arr
```

Out[27]:
```
array([[ 1,  2,  3, 12],
       [ 4,  5, 12, 11],
       [ 7,  8,  5, 10]])
```

In [28]:
```
arr
```

Out[28]:
```
array([[ 1,  2,  3, 12],
       [ 4,  5, 12, 11],
       [ 7,  8,  5, 10]])
```

In [29]:
```
#swap 5 with 9 in the above array
a = arr [1,1]
b = arr [2,2]
arr [2,2]= a
arr [1,1] = b
```

In [30]:
```
arr
```

Out[30]:
```
array([[ 1,  2,  3, 12],
       [ 4,  5, 12, 11],
       [ 7,  8,  5, 10]])
```

In [31]:
```
arr.mean()
```

Out[31]:
```
6.666666666666667
```

In [32]:
```
arr.min()
```

Out[32]:
```
1
```

In [33]:
```
arr.cumsum ()
```

Out[33]:
```
array([ 1,  3,  6, 18, 22, 27, 39, 50, 57, 65, 70, 80], dtype=int32)
```

In [34]:
```
np.mean (arr)
```

Out[34]:
```
6.666666666666667
```

In [35]:
```
arr.std()
```

Out[35]:
```
3.7490739597339973
```

In [36]:
```
arr.shape
```

Out[36]:
```
(3, 4)
```

In [37]:
```
arr
```

Out[37]:
```
array([[ 1,  2,  3, 12],
       [ 4,  5, 12, 11],
       [ 7,  8,  5, 10]])
```

In [38]:
```
arr.reshape (4,3)
```

Out[38]:
```
array([[ 1,  2,  3],
       [12,  4,  5],
       [12, 11,  7],
       [ 8,  5, 10]])
```

In [39]:
```
#repeating sequences
a = [1,2,3]
np.repeat(a,4)
```

Out[39]:
```
array([1, 1, 1, 1, 2, 2, 2, 2, 3, 3, 3, 3])
```

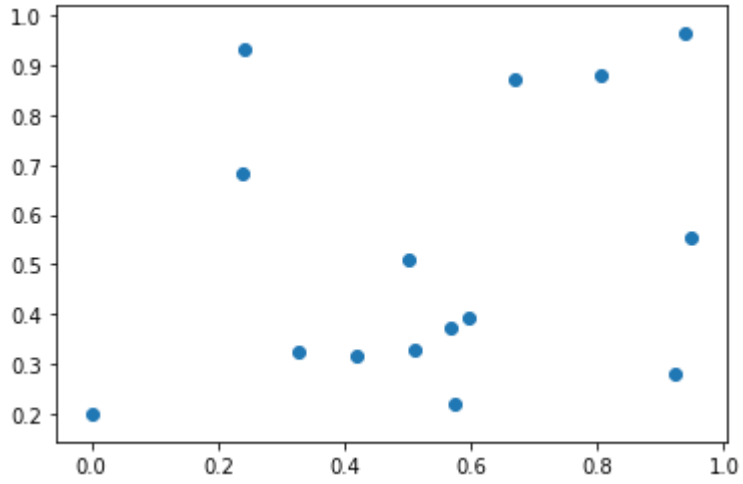
In [40]:
```
#tile repeats lists
np.tile(a,5)
```

Out[40]:
```
array([1, 2, 3, 1, 2, 3, 1, 2, 3, 1, 2, 3, 1, 2, 3])
```

In [84]:
```
import matplotlib.pyplot as plt
x = np.random.rand(15)
y = np.random.rand(15)
```

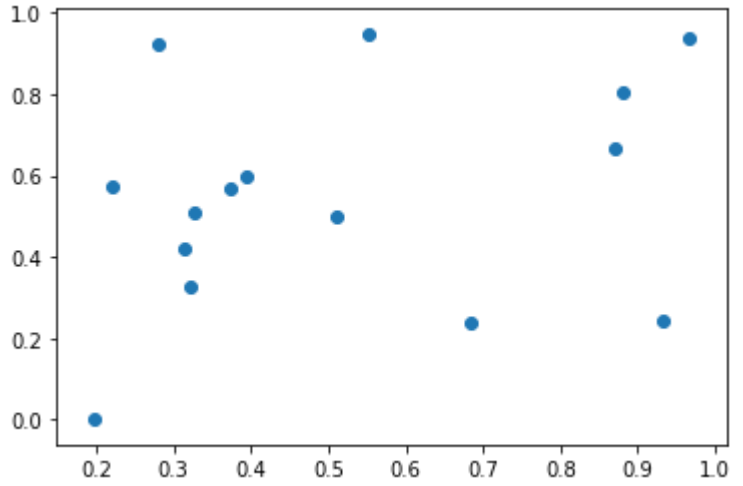
In [76]:
```
plt.scatter(x,y)
```

Out[76]: <matplotlib.collections.PathCollection at 0x19aa4170630>



In [77]:
```
plt.scatter(y,x)
```

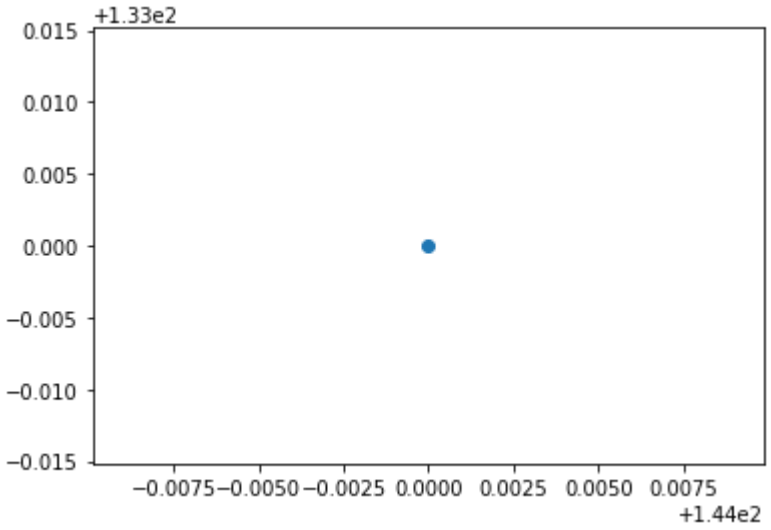
Out[77]: <matplotlib.collections.PathCollection at 0x19aa41c95c0>



In [81]:
```
x = np.random.randint(150)
y = np.random.randint(150)
```

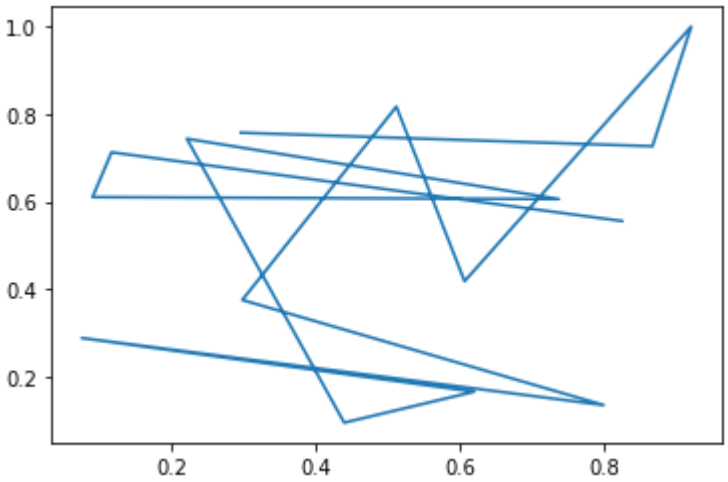
In [82]:
```
plt.scatter(x,y)
```

Out[82]:
```
<matplotlib.collections.PathCollection at 0x19aa42f2240>
```



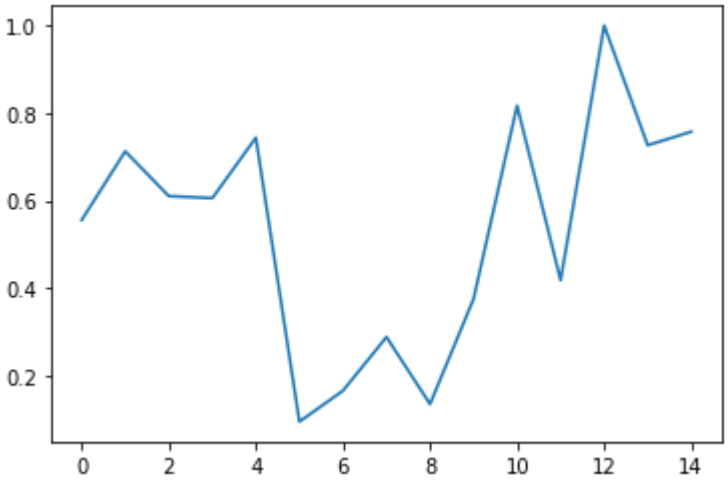
In [85]:
```
plt.plot(x,y)
```

Out[85]:
```
[<matplotlib.lines.Line2D at 0x19aa43bd0b8>]
```



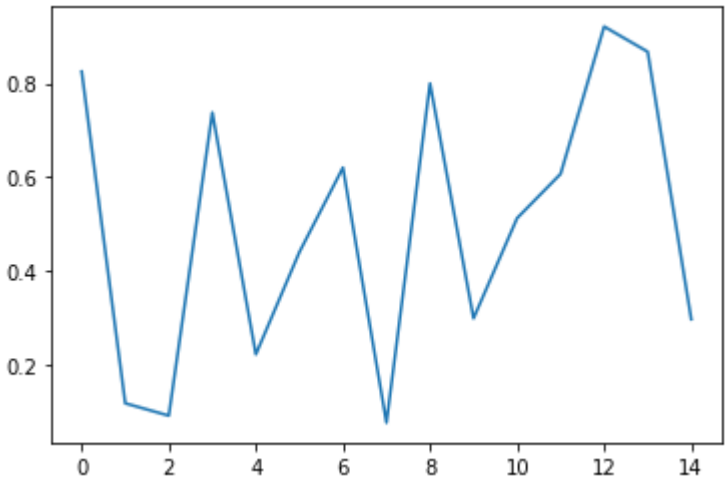
In [86]:
```
plt.plot(y)
```

Out[86]:
```
[<matplotlib.lines.Line2D at 0x19aa440d400>]
```



In [87]:
```
plt.plot(x)
```

Out[87]:
```
[<matplotlib.lines.Line2D at 0x19aa446ab00>]
```



In [88]:
```
plt.pie(x,y)
```

Out[88]:
```
([<matplotlib.patches.Wedge at 0x19aa44c5d68>,
  <matplotlib.patches.Wedge at 0x19aa44d2278>,
  <matplotlib.patches.Wedge at 0x19aa44d2710>,
  <matplotlib.patches.Wedge at 0x19aa44d2ba8>,
  <matplotlib.patches.Wedge at 0x19aa44df080>,
  <matplotlib.patches.Wedge at 0x19aa44df518>,
  <matplotlib.patches.Wedge at 0x19aa44df9b0>,
  <matplotlib.patches.Wedge at 0x19aa44dfe48>,
  <matplotlib.patches.Wedge at 0x19aa44ec320>,
  <matplotlib.patches.Wedge at 0x19aa44ec7b8>,
  <matplotlib.patches.Wedge at 0x19aa44a6d68>,
  <matplotlib.patches.Wedge at 0x19aa44fa0f0>,
  <matplotlib.patches.Wedge at 0x19aa44fa588>,
  <matplotlib.patches.Wedge at 0x19aa44faa20>,
  <matplotlib.patches.Wedge at 0x19aa44faeb8>],
 [Text(1.5557302880778323, 0.5657160560657132, ''),
  Text(1.3293790583206306, 1.2315700256132296, ''),
  Text(1.1471465169470538, 1.268168140766178, ''),
  Text(0.6405199486460003, 1.580612966354375, ''),
  Text(-0.03790654891103108, 1.8427223503783154, ''),
  Text(-0.35377339393206364, 1.1421706269346856, ''),
  Text(-0.8617640233058783, 0.9278609809260453, ''),
  Text(-1.1995998645262034, 0.6996344029615921, ''),
  Text(-1.2204642670654042, 0.19421908380093403, ''),
  Text(-1.4067056468973897, -0.44511118117842663, ''),
  Text(-1.5263307896426337, -1.1586306958374142, ''),
  Text(-0.6582044492924908, -1.3680746446685743, ''),
  Text(0.4116163142442058, -2.058083610696971, ''),
  Text(1.4887585620924662, -1.0572225042039223, ''),
  Text(1.84228432065418, -0.23233946161774344, '')])
```

In [ ]:
```

```