MatplotLib

Matplot lib is a plotting library for the python programming language

and its numerical mathematics extension Numpy.

It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits like Tkinter,WXPython, QT, or GTK+.

Some of the major pros of Matplotlib are:

***. Generally easy to get started for simple plots . Support for Custom labels and texts . Great control pf every element in a figure . High - quality ouput in many formats . very customizable in general ***

In [1]:
```
import matplotlib.pyplot as plt
%matplotlib inline

# instead of % we always use plt and we have to write plot.show directly w
```

In [2]:
```
# print(matplotlib.__version__)
```

In [3]:
```
import numpy as np
```

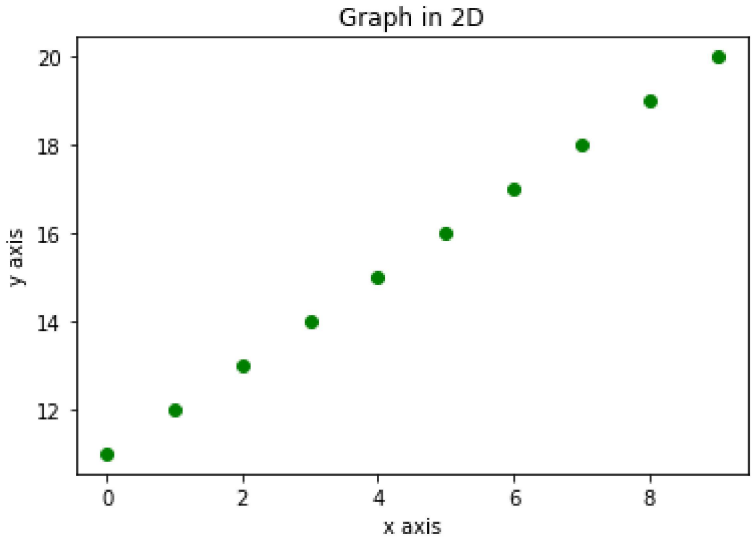
In [7]:
```
## simple Examples
x=np.arange(0,10)
y=np.arange(11,21) # i need to create 2D an  visualization diagram
print(x)
print(y)
```

```
[0 1 2 3 4 5 6 7 8 9]
[11 12 13 14 15 16 17 18 19 20]
```

In [8]:
```
a=np.arange(40,50)
b=np.arange(50,60)
print(a)
print(b)
```

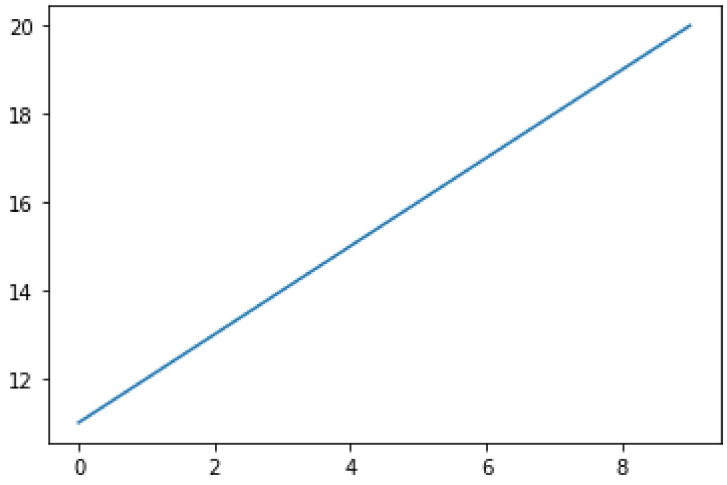
```
[40 41 42 43 44 45 46 47 48 49]
[50 51 52 53 54 55 56 57 58 59]
```

In [9]:
```
## plotting using matplotlib

## plt scatter ( use to scatter the values of x and y)

plt.scatter(x,y,c='g') # r=red, b=blue
plt.xlabel('x axis')
plt.ylabel('y axis')
plt.title('Graph in 2D')
plt.savefig('Test.png')
```



In [26]:
```
## plt plot

plt.plot(x,y)
```

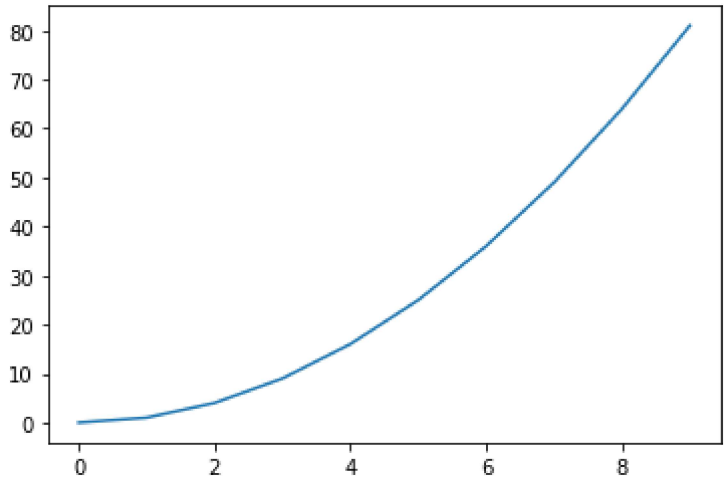
Out[26]: [<matplotlib.lines.Line2D at 0x1cc779e5b20>]



In [27]:
```
y=x*x
```

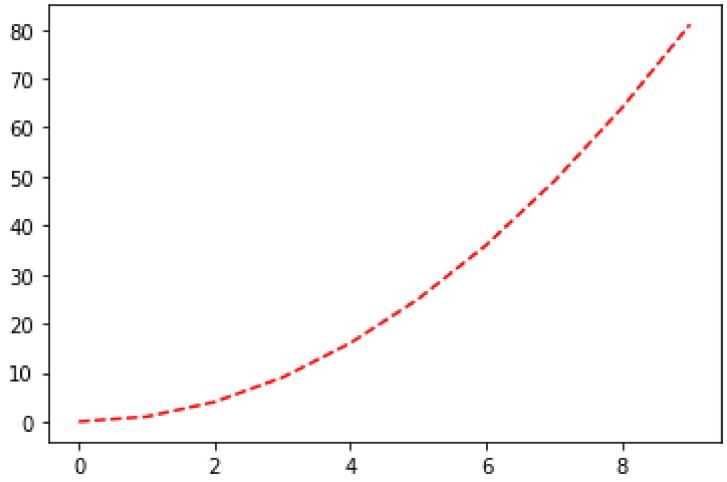
In [28]:
```
1 plt.plot(x,y)
```

Out[28]:
```
[<matplotlib.lines.Line2D at 0x1cc778df7c0>]
```



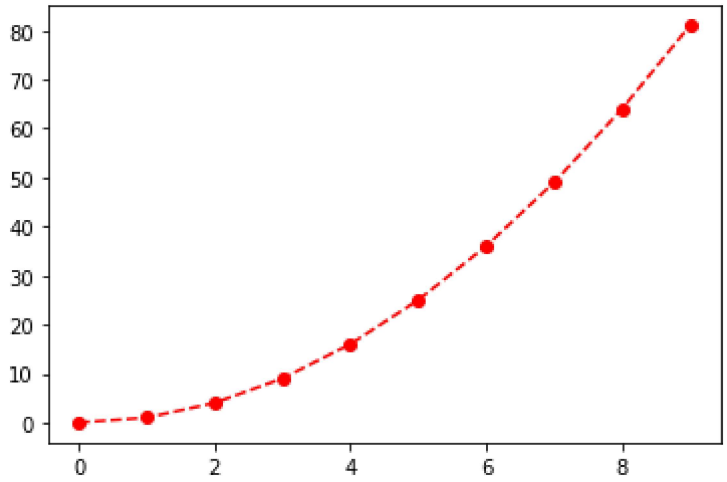
In [31]:
```
1 plt.plot(x,y,'r--')
```

Out[31]:
```
[<matplotlib.lines.Line2D at 0x1cc77a46e20>]
```



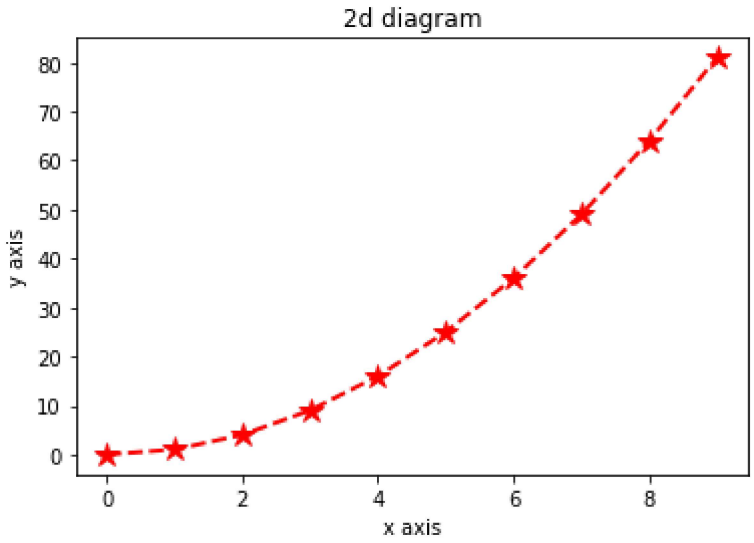
In [33]:
```
1 plt.plot(x,y,'ro--')
```

Out[33]:
```
[<matplotlib.lines.Line2D at 0x1cc774fc070>]
```



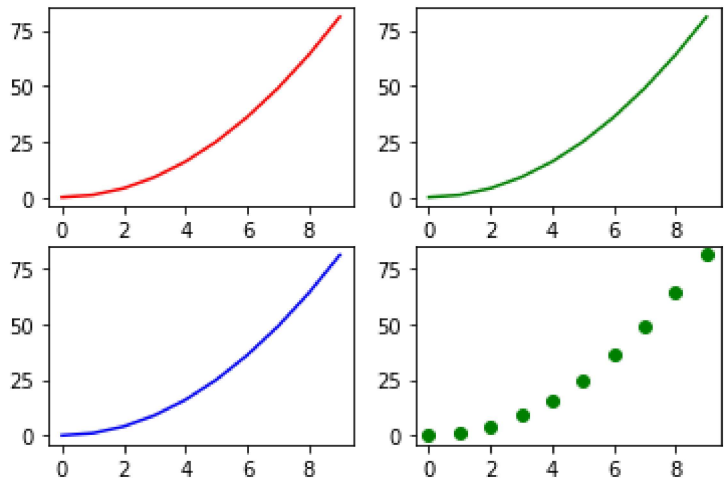
In [35]:
```
1 plt.plot(x,y,'r*',linestyle='dashed',linewidth=2,markersize=12)
2 plt.xlabel('x axis')
3 plt.ylabel('y axis')
4 plt.title('2d diagram')
```

Out[35]:
```
Text(0.5, 1.0, '2d diagram')
```

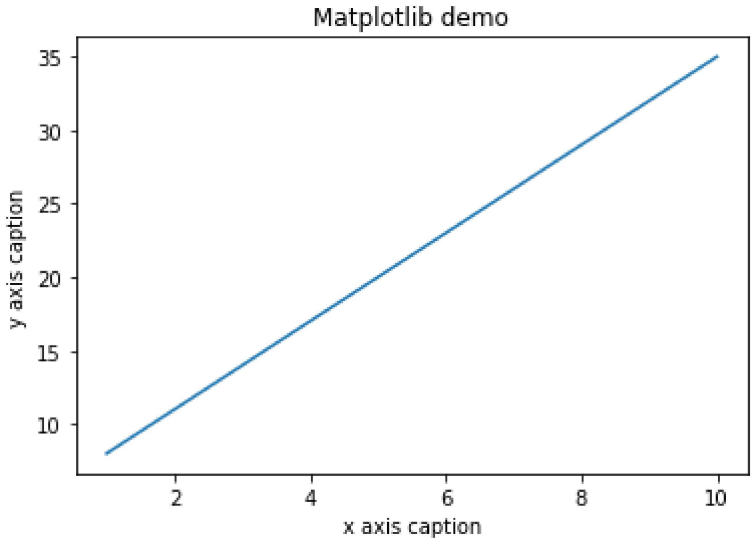


In [40]:
```
# creating sub plots
plt.subplot(2,2,1)
plt.plot(x,y,'r')
plt.subplot(2,2,2)
plt.plot(x,y,'g')
plt.subplot(2,2,3)
plt.plot(x,y,'b')
plt.subplot(2,2,4)
plt.plot(x,y,'go')
```

Out[40]:
```
[<matplotlib.lines.Line2D at 0x1cc792f1040>]
```



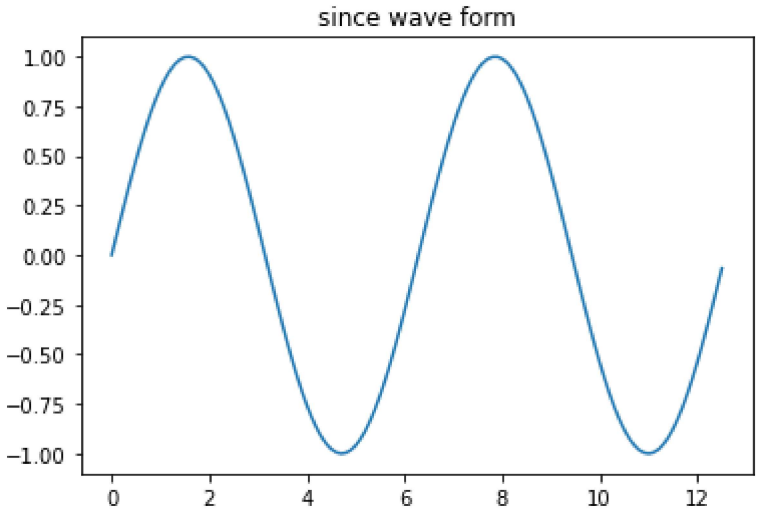
In [42]:
```
x = np.arange(1,11)
y = 3 * x + 5
plt.title("Matplotlib demo")
plt.xlabel("x axis caption")
plt.ylabel("y axis caption")
plt.plot(x,y)
plt.show()
```



In [43]:
```
1 np.pi # 22/7
```

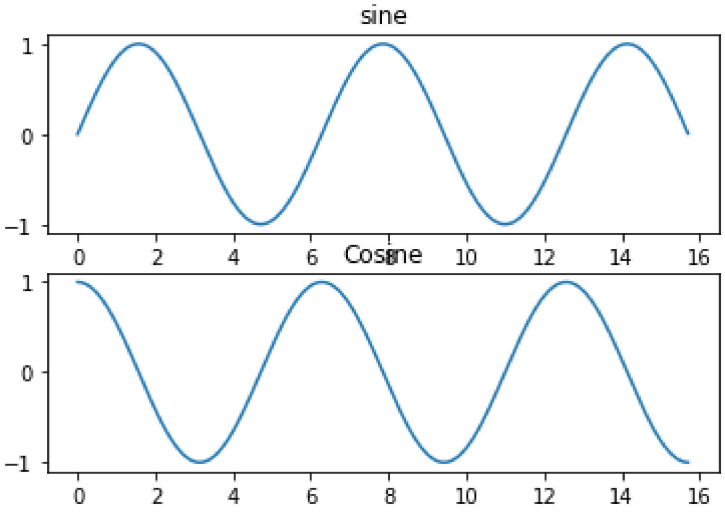
Out[43]: 3.141592653589793

In [44]:
```
1 # compute the x and Y coordinates for points on a sine curve
2 x= np.arange(0,4 * np.pi,0.1)
3 y = np.sin(x)
4 plt.title("since wave form")
5
6 # plot the points using matplot lib
7
8 plt.plot(x,y)
9 plt.show()
```

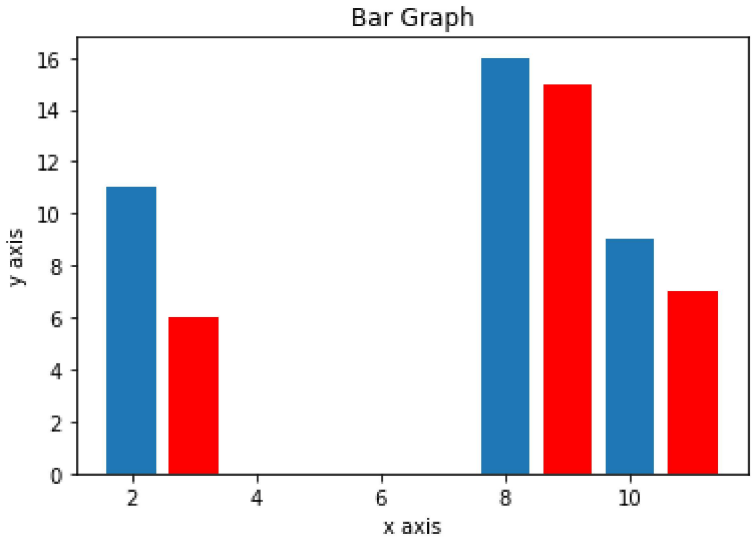


In [49]:

```
# subplot()
# Compute the x and y coordinates for points on sine and cosine curves
x = np.arange(0, 5 * np.pi, 0.1)
y_sin = np.sin(x)
y_cos = np.cos(x)

# set up a sub plot grid that has height 2 and width ,
# and set the first such subplot as active

plt.subplot(2,1,1)
# make the first plot
plt.plot(x,y_sin) # 'b--'
plt.title('sine')

# Set the second subplot as active, and make the second plot.
plt.subplot(2,1,2)
plt.plot(x,y_cos) # 'g--
plt.title('Cosine')

# show the fig
plt.show()
```

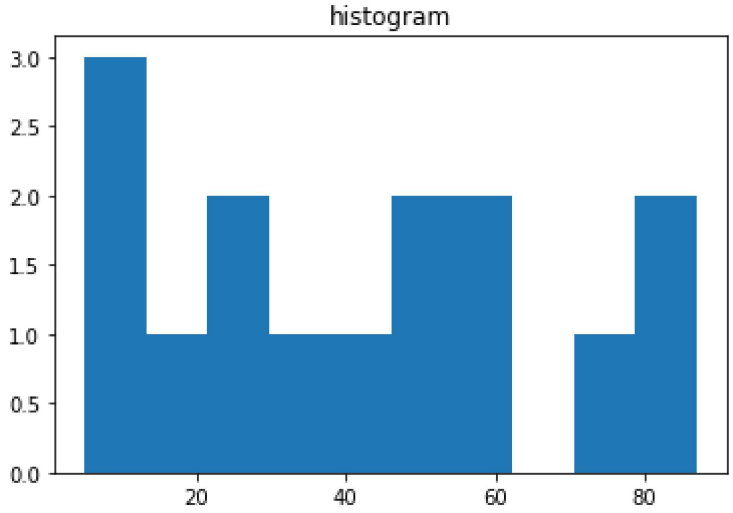


In [54]:
```
# Bar plot

x=[2,8,10]
y=[11,16,9]
x2 = [3,9,11]
y2 = [6,15,7]
plt.bar(x,y)
plt.bar(x2,y2,color='r')
plt.title('Bar Graph')
plt.ylabel('y axis')
plt.xlabel('x axis')
plt.show()
```

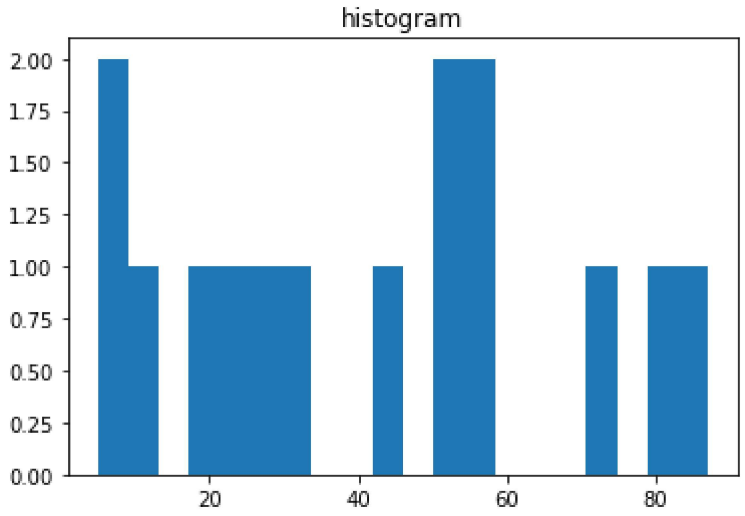


Histogram

In [4]:
```
a = np.array([22,87,5,43,56,73,55,54,11,20,51,5,79,31,27])
plt.hist(a)
plt.title("histogram")
plt.show()
```



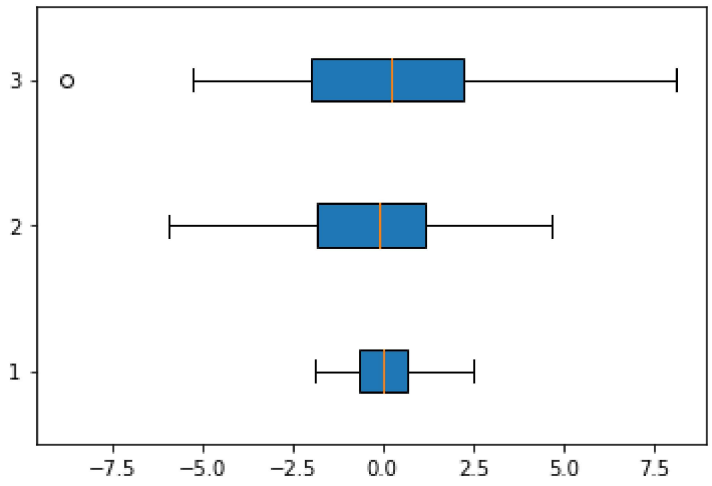
In [5]:
```
a = np.array([22,87,5,43,56,73,55,54,11,20,51,5,79,31,27])
plt.hist(a,bins=20)
plt.title("histogram")
plt.show()
```



Box Plot using Matplotlib

A Box plot is to created to display the set of data values having properties like minimum,first quartile,median,third quartile and maximum.

In [8]:
```
data = [np.random.normal(0, std, 100) for std in range(1,4)]

# rectangle box plot

plt.boxplot(data,vert=False,patch_artist=True);
```



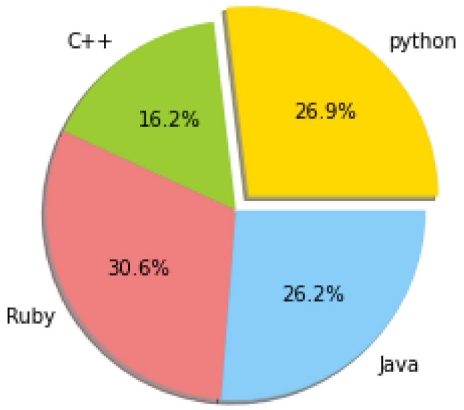
In [7]:
```
1 data
```

Out[7]:
```
[array([-0.4755428 , -1.64365065,  0.07670247, -0.72569885, -0.39623313,
        -1.18677333, -0.66964339,  0.2543023 , -0.79920123, -0.21059098,
         1.12391626, -0.33113633, -0.7232409 ,  0.4861032 ,  2.40145613,
         0.68474607,  0.95412983,  0.14967977,  0.91647046,  0.54144577,
        -1.07027336,  0.88019941,  0.97617979, -0.26605712, -0.69697324,
         0.43531049, -0.15660533,  0.06234313, -1.64502124,  0.31278744,
         0.05293133,  0.98070207, -1.92060028, -0.90493676, -0.7095216 ,
         0.15148847,  0.91796555,  0.78691744, -0.13789225, -0.02614623,
         0.15972292, -1.58464849, -0.61092913,  1.21011854,  0.55293963,
         0.03797287,  0.77469877, -1.70714303,  0.56132552,  0.43578146,
         0.08569558,  0.14760292, -0.14068482,  0.21216604, -0.85111989,
        -0.68500413,  0.51137404, -0.91135965,  0.57393162,  1.41100399,
        -0.28663456, -0.89591314,  1.08001789, -0.57116831, -0.3817778 ,
         1.35619307,  0.12896623,  0.44042485, -0.51159419, -2.78249696,
         0.363208  , -0.3196117 , -1.03380218, -0.36322418, -0.88233447,
        -0.44042014,  0.85808901, -0.72445472, -1.04219988, -2.39400842,
        -0.38126329, -1.5554564 , -0.27964083, -0.07817333,  0.2164768 ,
         0.14372843,  0.09657282,  1.95259547,  0.21084029, -0.63003963,
        -1.74778344, -0.80599749,  0.68021581, -0.22403971, -0.62431036,
        -0.72328433, -0.50470891,  0.24735229,  1.066252  ,  2.01927226]),
 array([-1.35318187, -1.39608562,  2.70898139, -1.87472805, -1.27479738,
        -0.58979927, -0.03056212,  0.95410149, -0.167283  ,  1.75973685,
         1.15782917, -2.13390467,  0.52100556,  0.68681421, -2.70367634,
         0.93063262, -1.20437112,  2.49553808, -0.6569862 , -1.37269374,
        -3.26348472, -2.17651166, -4.22013242, -1.53928813,  3.04052049,
        -1.58696419,  3.55775359,  4.47299928, -2.83579763,  0.43996655,
        -1.38439278,  1.26505439,  3.90755113, -1.10721825, -0.46446172,
        -2.25741797,  0.88354279, -0.55614129,  0.55436966, -1.29705394,
        -0.20868796, -1.16994499,  2.57600656, -1.68407255,  0.9292838 ,
         2.99880877, -0.59415261,  0.1010363 , -0.14580206,  2.79686748,
        -0.93532726, -1.32045663,  1.21625168,  1.24344084, -1.39994533,
         0.9167393 , -0.04287487,  0.8403024 , -0.09114053,  1.57147718,
         1.67948895,  3.3436428 , -1.06075926, -2.29217582, -1.40664654,
        -0.36525506,  1.92097124,  3.33347149, -1.83659529,  1.87947232,
         0.43793552,  1.22201898, -0.58025357,  1.12685236, -0.17235064,
        -1.3001744 ,  0.65190006, -0.33624832, -0.10272956, -0.20300155,
         3.1343291 , -0.04913706, -0.11258112,  3.62371993, -1.27952074,
        -3.40826762,  0.26924857,  4.85827872, -1.88043821, -1.37385783,
         0.08983494,  2.1330363 ,  0.1871799 , -0.3884085 , -1.33657255,
         0.92301016,  0.49300517, -5.57096787, -4.14493145, -1.46570078]),
 array([-3.51294205e+00,  3.07870226e+00, -5.87533813e+00,  4.21904134e-01,
        -2.44632419e-01,  8.70084412e-01,  2.54771353e+00,  8.60856206e-01,
         3.56476914e+00, -3.91938649e+00,  4.29329981e+00,  8.50188363e-01,
        -3.56064092e+00, -1.16633870e-01, -9.56603794e-01,  1.39886721e+00,
        -3.68866426e+00,  1.25195803e+00,  6.10180664e+00, -2.48653290e+00,
         1.49791124e+00,  6.08643861e+00, -1.28144706e+00,  7.03788841e-01,
        -3.95807348e+00,  1.33576557e+00,  1.96486183e-01,  4.23970177e+00,
        -1.33872095e+00,  2.60725106e+00,  7.93118598e-01, -3.21459255e+00,
        -2.30181394e+00, -4.88197344e+00,  1.63048516e+00,  1.60393726e+00,
         1.37971932e-01, -4.60441302e+00, -4.79700305e+00, -1.99210624e+00,
         6.54809319e+00,  4.41990321e+00,  2.02672876e+00,  6.33908805e-01,
         6.89163142e-01,  3.89376344e+00,  1.06259460e+00, -2.33780436e-01,
         9.17568512e-01, -6.18552611e-01, -4.20815412e+00,  5.30806619e+00,
        -5.95959520e+00,  9.90970828e-01,  5.92666218e-01,  3.06716224e+00,
         3.96314049e+00,  1.42461174e+00, -7.56534302e+00, -2.40401813e+00,
         1.56062096e+00,  1.50724004e+00, -3.69208330e+00,  7.96522597e-01,
        -1.06959143e+00,  1.87392061e+00,  2.58766298e+00,  3.06329943e+00,
```

```
-5.31606751e+00,  6.15005494e+00,  1.76054328e+00,  4.86814265e+00,
 1.66616964e+00, -2.04497892e+00,  3.51361053e+00,  3.58543382e-02,
-1.35555498e+00, -2.38118865e+00,  7.51335503e+00, -6.25427297e+00,
 7.04325532e-01,  2.45443564e-01, -7.50378894e-03,  5.22271284e-01,
 3.55246628e-02, -2.21957178e+00,  1.58159453e-01, -4.01622037e+00,
-9.22557412e-01,  2.10467648e+00,  1.04463168e+00,  2.53462068e+00,
 1.80783038e+00,  9.30664864e-03, -1.61807446e+00, -5.97074012e-02,
 4.95663157e+00,  6.53451188e-01,  6.84286009e+00,  3.24630056e+00])]
```

Pie Chart

In [14]:
```
# 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='%1.1f%%',shadow=True)

plt.axis('equal')
plt.show()
```



In [ ]:
```
1
```

In [ ]:
```
1
```