matplotlib

July 30, 2023

1 MATPLOTLIB Tutorial

[1]:
```
from matplotlib import pylab
print(pylab.__version__)
```

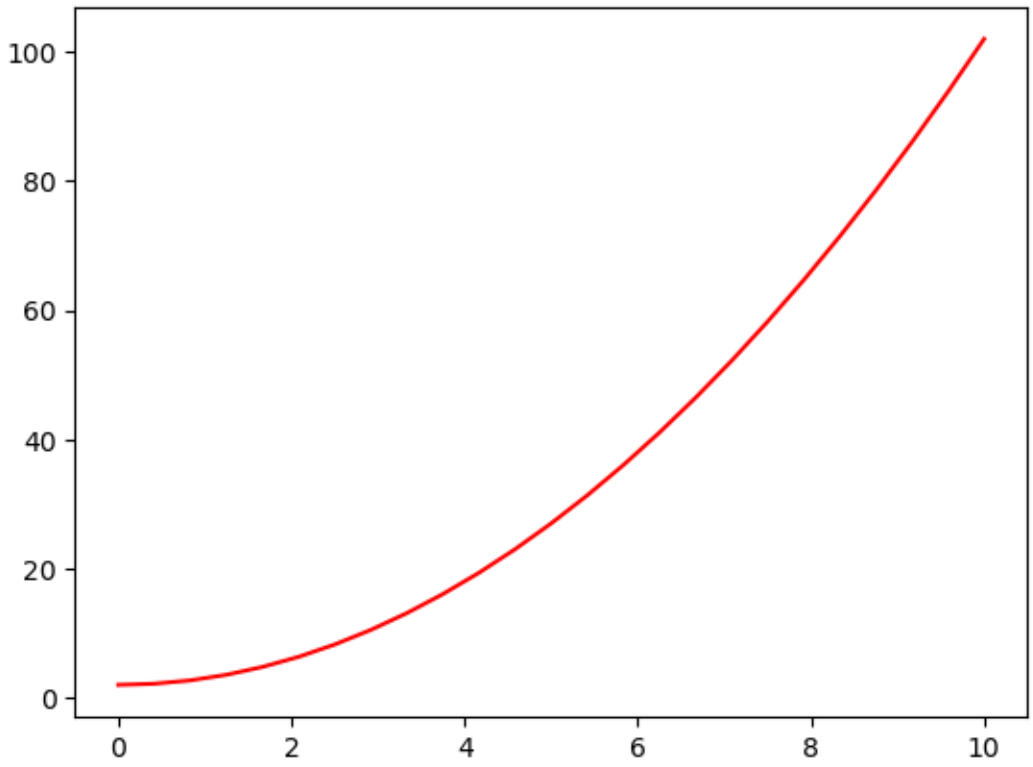
```
1.23.5
```

[2]:
```
# use Numpy to generate random data
import numpy as np
x = np.linspace(0,10,25)
y = x*x+2
print(x)
print(y)
print(np.array([x,y]).reshape(2,25))
```

```
[ 0.          0.41666667  0.83333333  1.25        1.66666667  2.08333333
  2.5         2.91666667  3.33333333  3.75        4.16666667  4.58333333
  5.          5.41666667  5.83333333  6.25        6.66666667  7.08333333
  7.5         7.91666667  8.33333333  8.75        9.16666667  9.58333333
 10.        ]
[  2.           2.17361111   2.69444444   3.5625       4.77777778
   6.34027778   8.25        10.50694444  13.11111111  16.0625
  19.36111111  23.00694444  27.          31.34027778  36.02777778
  41.0625      46.44444444  52.17361111  58.25        64.67361111
  71.44444444  78.5625      86.02777778  93.84027778 102.        ]
[[  0.           0.41666667   0.83333333   1.25         1.66666667
    2.08333333   2.5          2.91666667   3.33333333   3.75
    4.16666667   4.58333333   5.           5.41666667   5.83333333
    6.25         6.66666667   7.08333333   7.5          7.91666667
    8.33333333   8.75         9.16666667   9.58333333  10.        ]
 [  2.           2.17361111   2.69444444   3.5625       4.77777778
    6.34027778   8.25        10.50694444  13.11111111  16.0625
   19.36111111  23.00694444  27.          31.34027778  36.02777778
   41.0625      46.44444444  52.17361111  58.25        64.67361111
   71.44444444  78.5625      86.02777778  93.84027778 102.        ]]
```

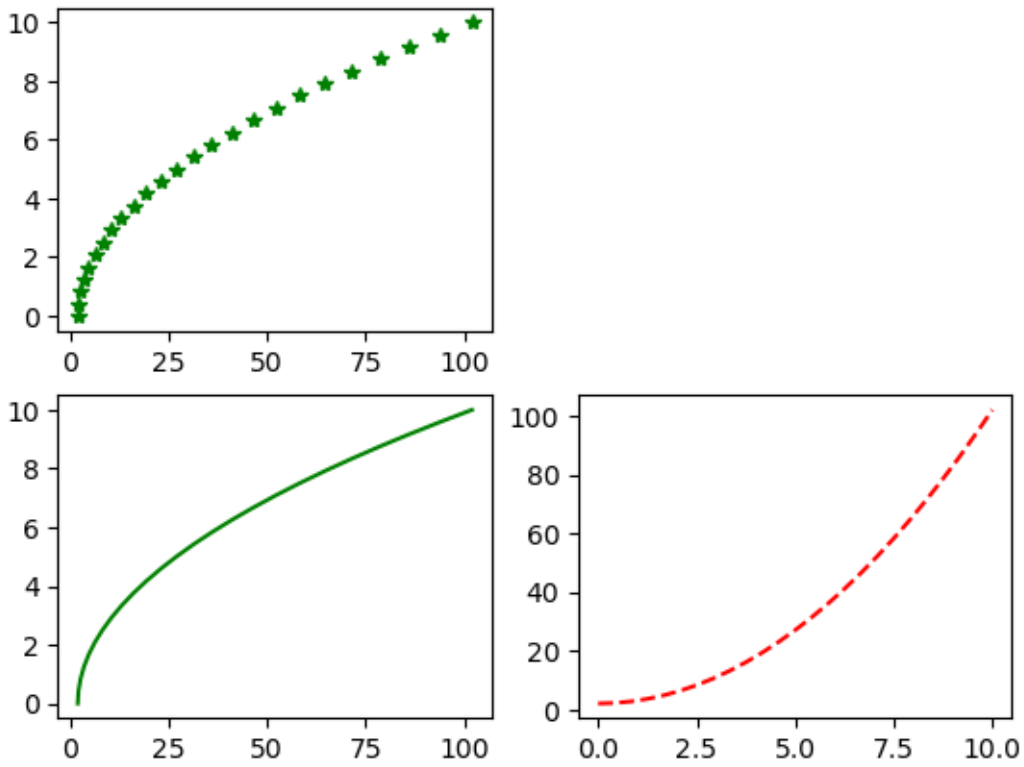
[3]:
```
# It only takes 1 command to draw
pylab.plot(x,y,'r') #'r' stands for red
```

[3]: [<matplotlib.lines.Line2D at 0x1856180bdc0>]



[4]:
```
# Drawing a subgraph
pylab.subplot(2,2,4)#The contents of the branchets␣
 ↪represent(rows,columns,indexes)
pylab.plot(x,y,'r--')#The third parameter here determines color and line style
pylab.subplot(2,2,1)
pylab.plot(y,x,'g*')
pylab.subplot(2,2,3)
pylab.plot(y,x,'g-')
```

[4]: [<matplotlib.lines.Line2D at 0x185618be7d0>]



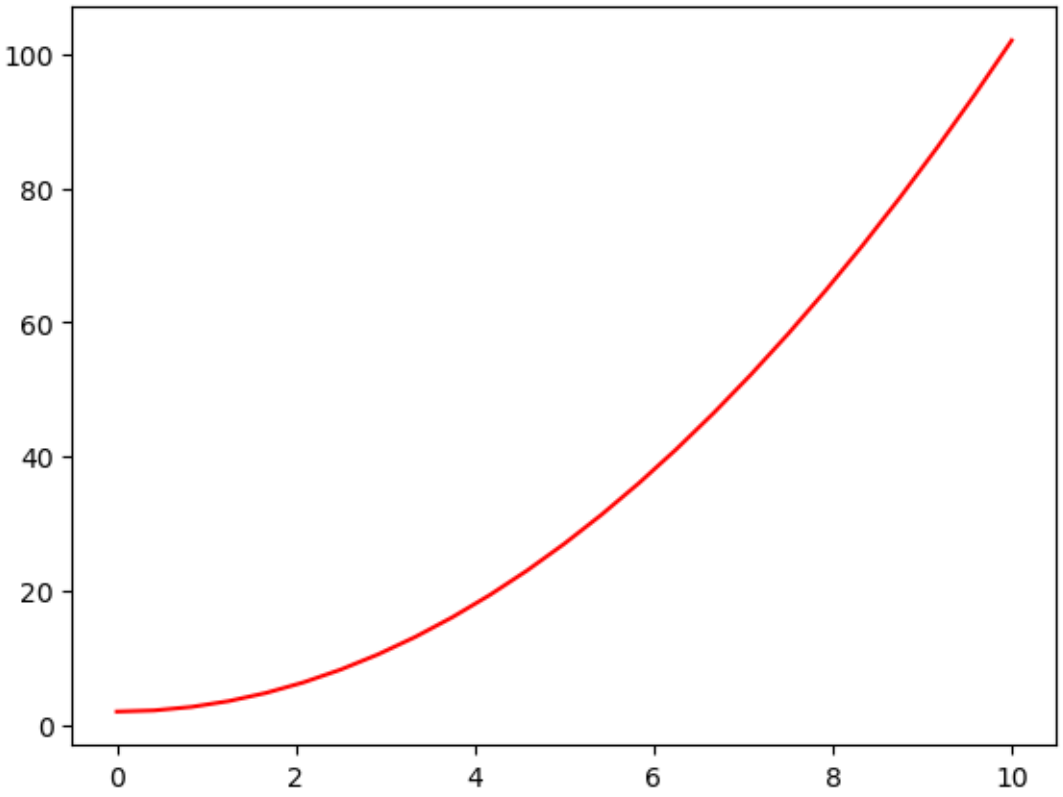
2 Operator Description

fig.add_axes() = Initializes subplot a = fig.add_subplot(222)
fig,b=plt.subplots(nrows=3,nclos=2) = Adds subplot
ax = plt.subplots(2,2) = Creates subplot

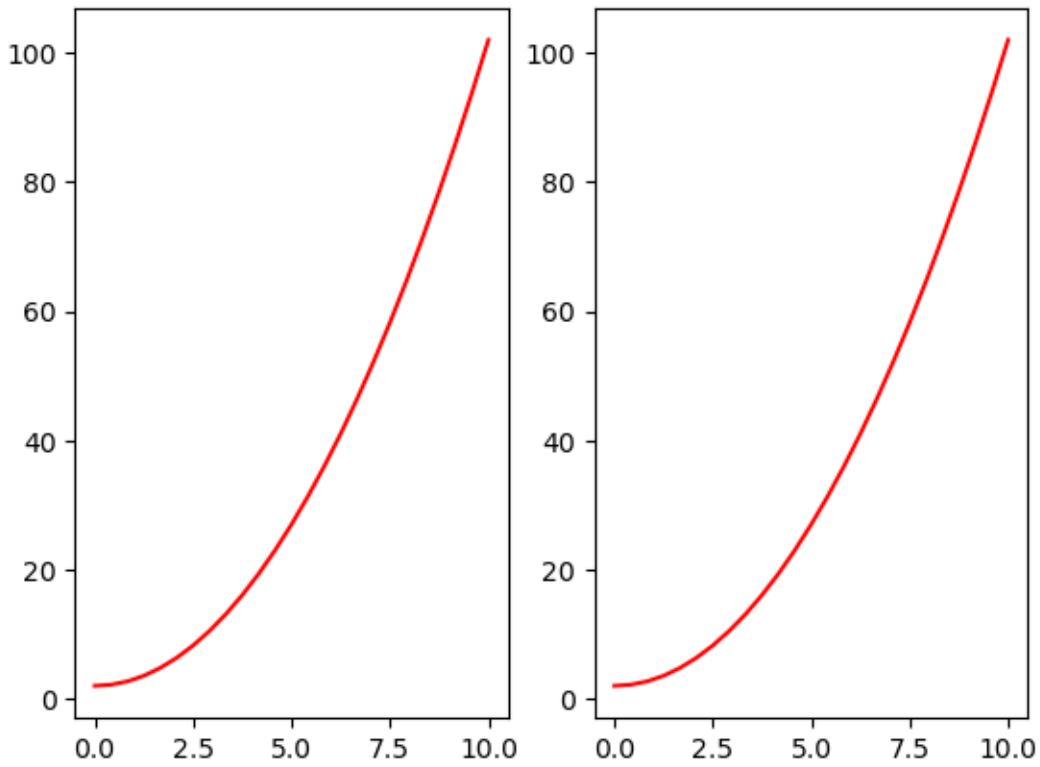
[5]:
```
from matplotlib import pyplot as plt
```

[6]:
```
fig = plt.figure()
axis = fig.add_axes([0.5,0.1,0.8,0.8]) #Control the left,right,width,height of␣
 ↪the canvas (from 0 to 1)
axis.plot(x,y,'r')
```

[6]:
```
[<matplotlib.lines.Line2D at 0x1856198ff70>]
```

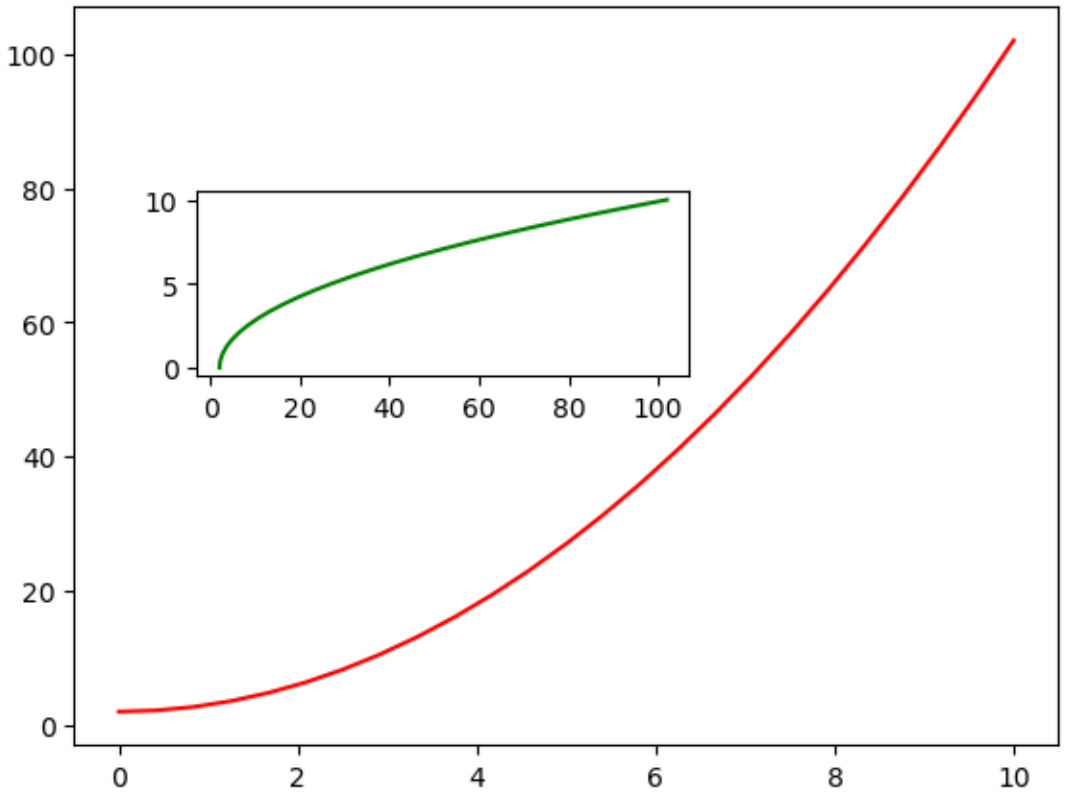


[7]:
```
# again we can draw subgraphs
fig,axes = plt.subplots(nrows = 1,ncols = 2) #submap is of 1row,2 columms
for ax in axes:
    ax.plot(x,y,'r-')
```



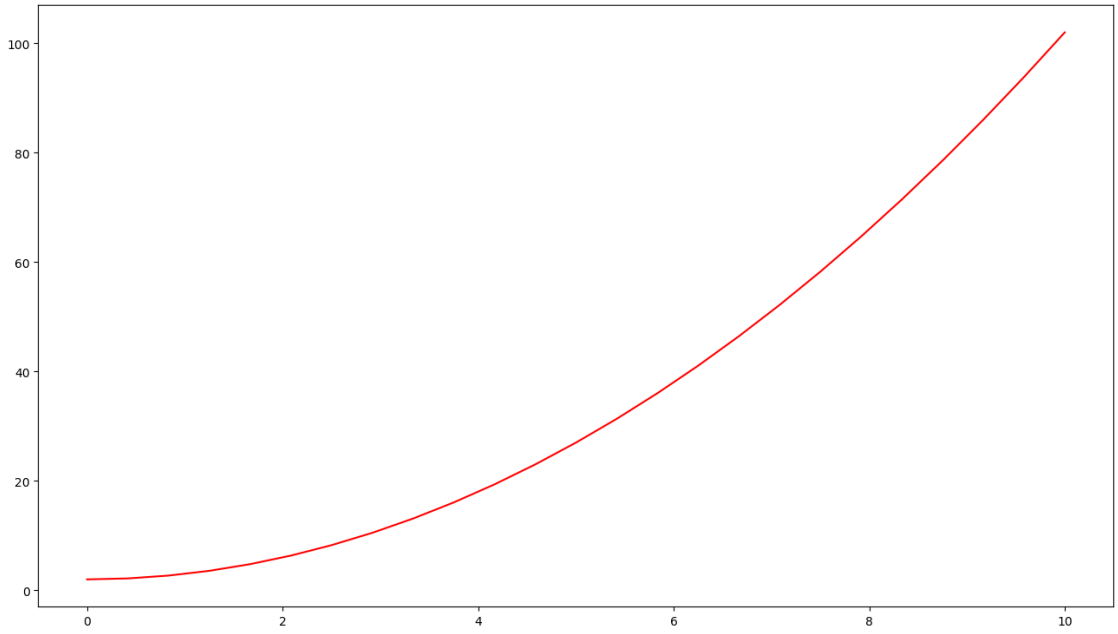
[8]:
```
# we can also draw a picture ,or graph ,inside another graph
fig = plt.figure()

#Control the left,right,width,height of the canvas(from 0 to 1)
axes1 = fig.add_axes([0.1,0.1,0.8,0.8])# big axes
axes2 = fig.add_axes([0.2,0.5,0.4,0.2]) #small canvas

axes1.plot(x,y,'r-')
axes2.plot(y,x,'g-')
```

[8]:
```
[<matplotlib.lines.Line2D at 0x185624199c0>]
```



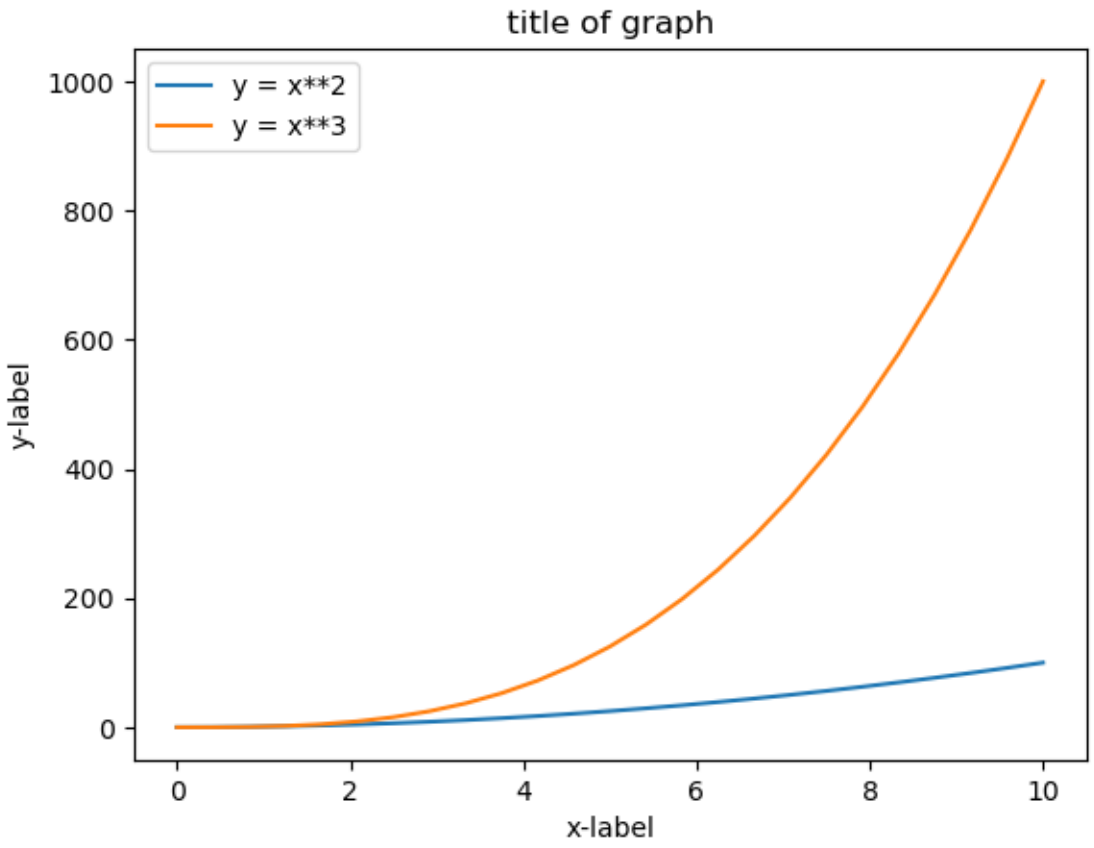
[9]:
```
fig = plt.figure(figsize=(16,9),dpi=100)#New graphic object
fig.add_subplot()
plt.plot(x,y,'r')
```

[9]: [<matplotlib.lines.Line2D at 0x18563473a00>]



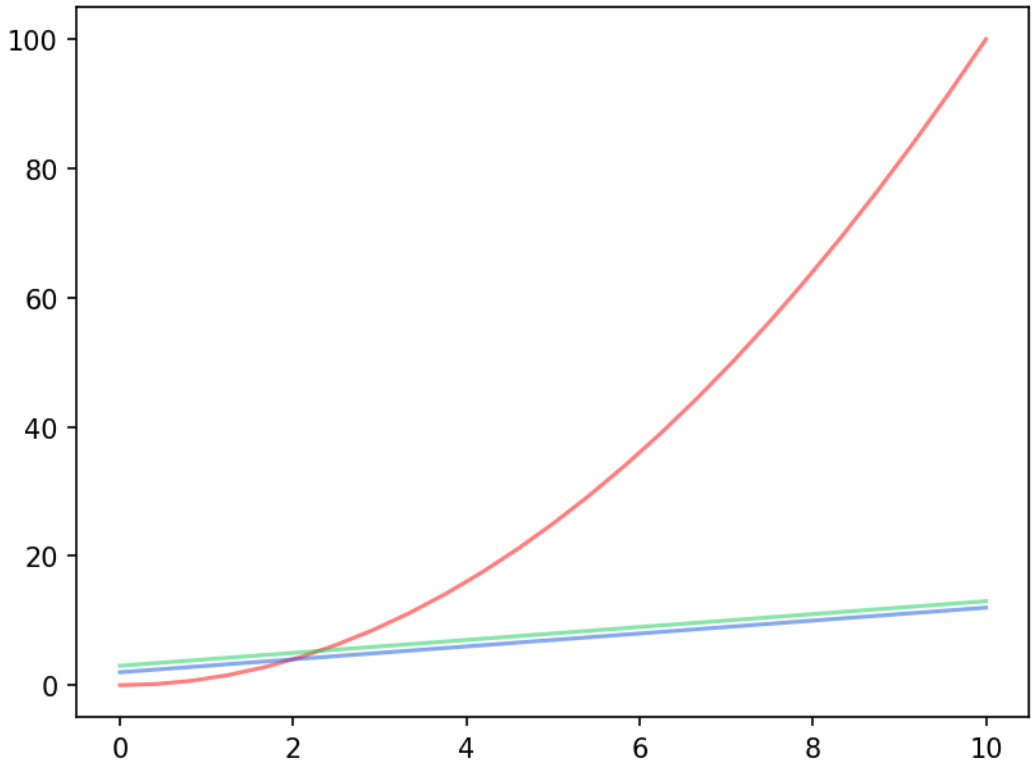
[10]:
```
ax.legend(["label1","label2"])

fig,axes = plt.subplots()
axes.set_title("title of graph")
axes.set_xlabel("x-label")
axes.set_ylabel("y-label")

axes.plot(x,x**2)
axes.plot(x,x**3)

axes.legend(["y = x**2","y = x**3"],loc=2)
```

[10]:
```
<matplotlib.legend.Legend at 0x185634e38b0>
```



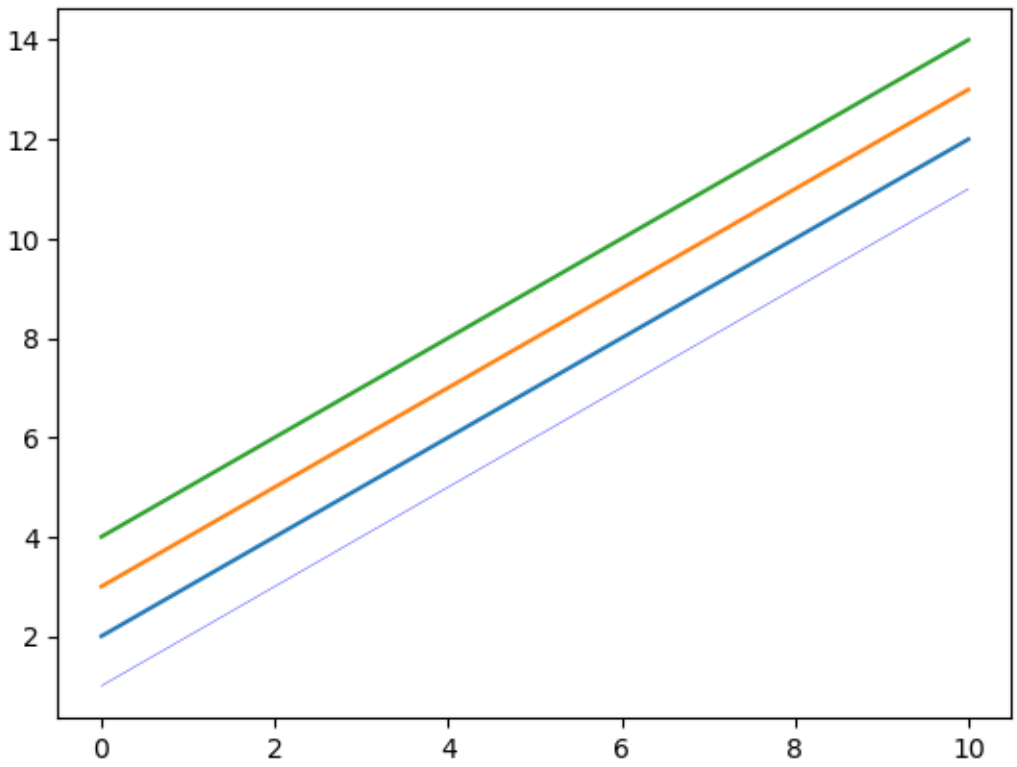
[11]:
```
# Int Matplotlib,you can set other properties such as line␣
↪color,transparency,and more,

fig, axes = plt.subplots(dpi=150)
axes.plot(x,x**2,color='red',alpha=.5)
axes.plot(x,x+2,color="#1155dd",alpha=.5)
axes.plot(x,x+3,color="#15cc55",alpha=.5)
```

[11]:
```
[<matplotlib.lines.Line2D at 0x185635a1060>]
```

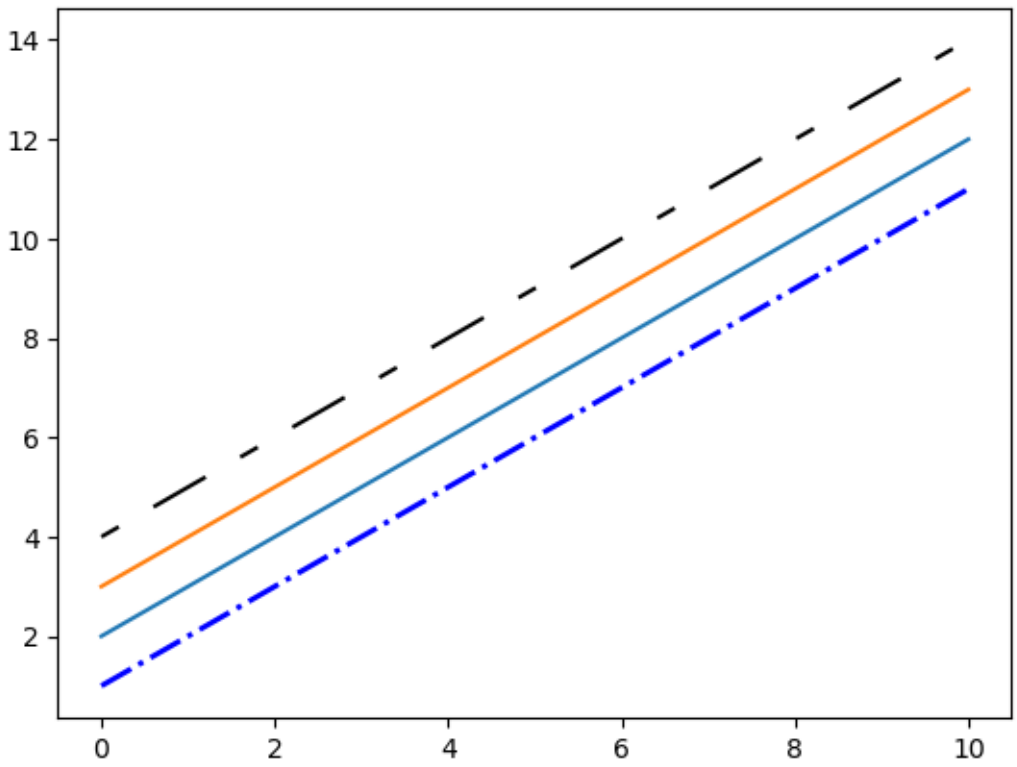


[12]:
```
fig, ax = plt.subplots(dpi=100)

#line width
ax.plot(x,x+1,color='blue',linewidth=0.25)
ax.plot(x,x+2)
ax.plot(x,x+3)
ax.plot(x,x+4)
```

[12]: [<matplotlib.lines.Line2D at 0x18563606e90>]

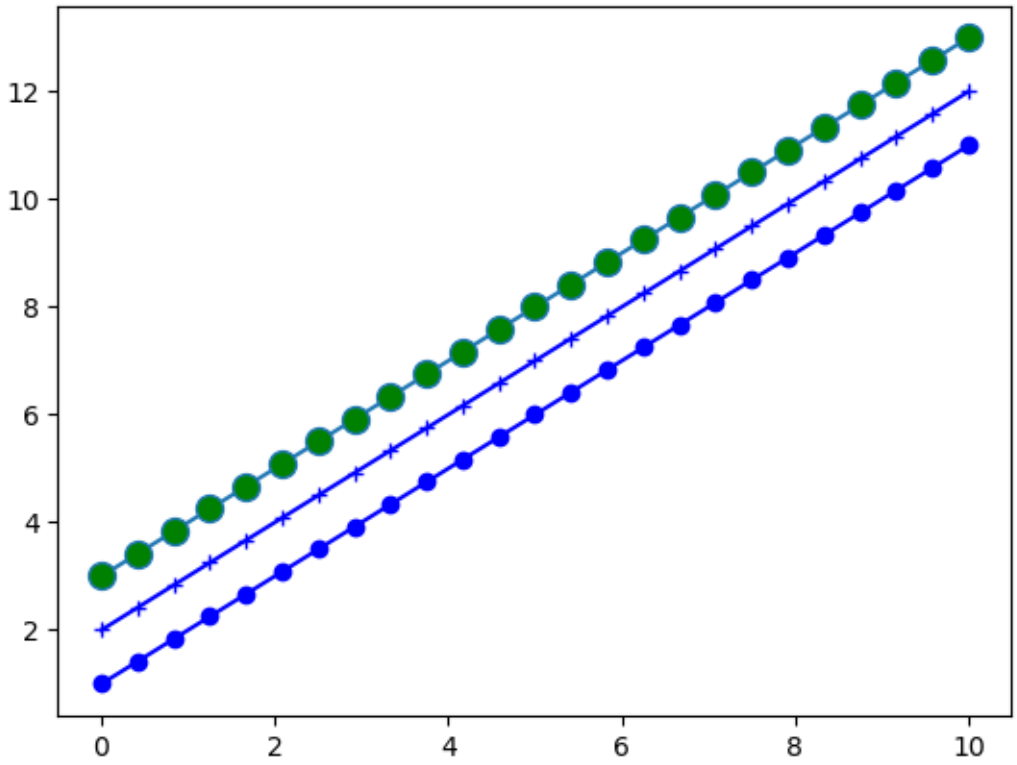


[13]:
```
fig, ax = plt.subplots(dpi=100)

#line width
ax.plot(x,x+1,color='blue',lw=2,linestyle='-.')
ax.plot(x,x+2)
ax.plot(x,x+3)

line, = ax.plot(x,x+4,color="black",lw=1.50)
line.set_dashes([5,10,15,10])
```



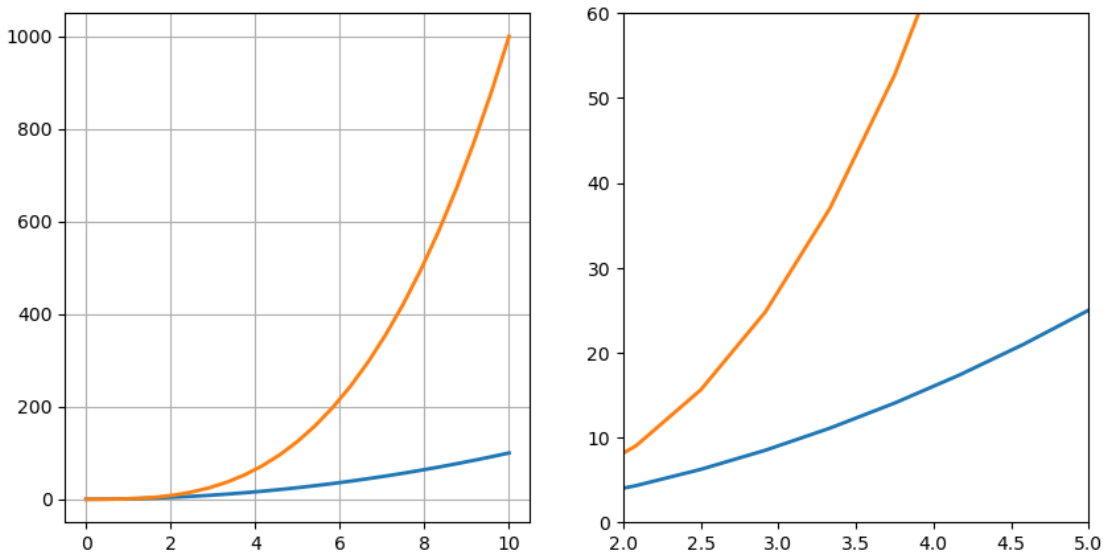
[14]:
```
fig, ax = plt.subplots(dpi=100)

#line width
ax.plot(x,x+1,color='blue',marker='o')
ax.plot(x,x+2,color='blue',marker='+')
ax.plot(x,x+3,marker='o',markersize=10,markerfacecolor='green')
```

[14]: [<matplotlib.lines.Line2D at 0x18563817130>]



[15]:
```
# Set the canvas grid and axis range

fig, Axes = plt.subplots(1,2,figsize=(10,5))

Axes[0].plot(x,x**2,x,x**3,lw=2)
Axes[0].grid(True)

Axes[1].plot(x,x**2,x,x**3,lw=2)
Axes[1].set_ylim([0,60])
Axes[1].set_xlim([2,5])
```

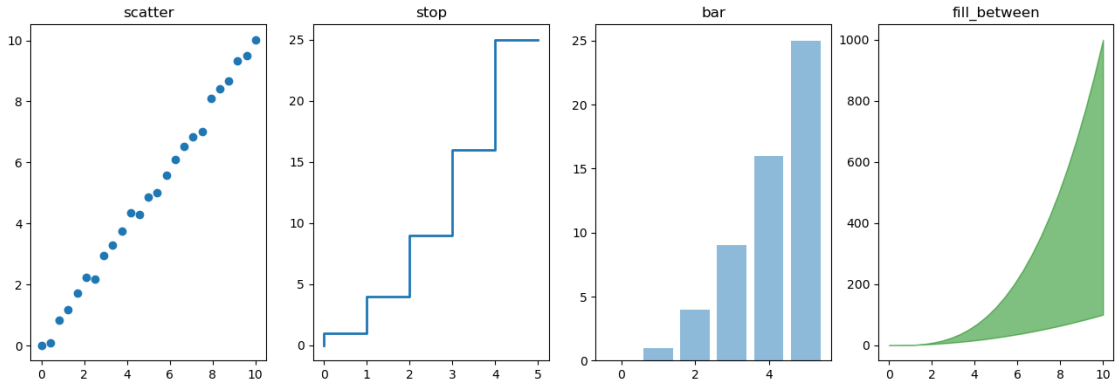
[15]: (2.0, 5.0)



3 Other 2D Graphics

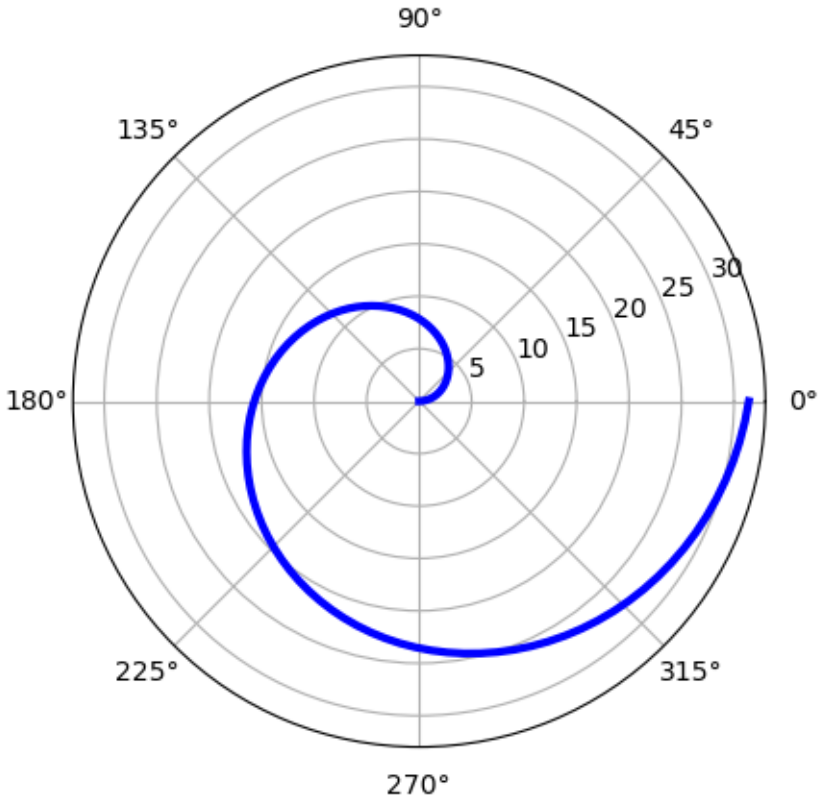
[16]:
```
n = np.array([0,1,2,3,4,5])
fig,axes = plt.subplots(1,4,figsize=(16,5))
axes[0].set_title("scatter")
axes[0].scatter(x,x+0.25*np.random.randn(len(x)))

axes[1].set_title("stop")
axes[1].step(n,n**2,lw=2)

axes[2].set_title("bar")
axes[2].bar(n,n**2,align="center",alpha=0.5)

axes[3].set_title("fill_between")
axes[3].fill_between(x,x**2,x**3,color="green",alpha=0.5)
```

[16]: <matplotlib.collections.PolyCollection at 0x18562338d00>



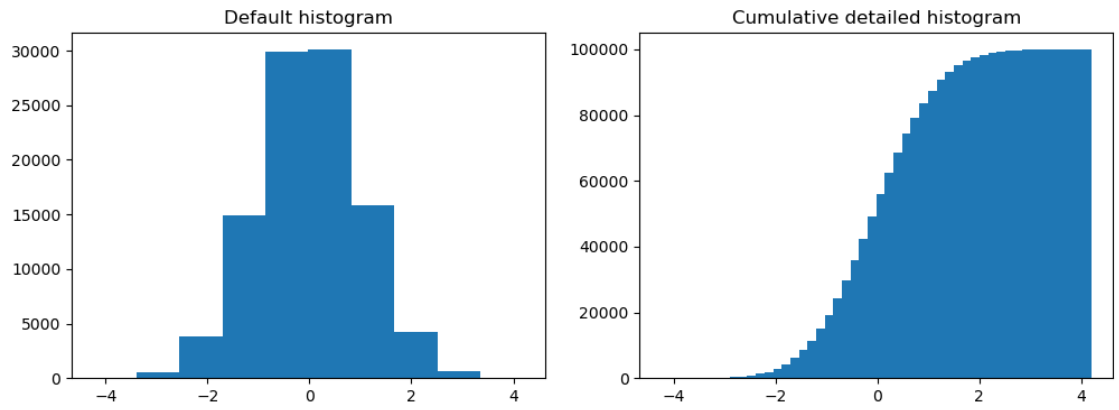
[17]:
```
# Draw a radar chart
fig = plt.figure(figsize=(6,6))
ax = fig.add_axes([0.0,0.0,.6,.6],polar=True)
t = np.linspace(0,2*np.pi,100)
ax.plot(t,t*5,color="blue",lw=3)
```

[17]: [<matplotlib.lines.Line2D at 0x18564172980>]



[18]:
```
# Draw a histogram

n = np.random.randn(100000)
fig, axes = plt.subplots(1,2,figsize=(12,4))
axes[0].set_title("Default histogram")
axes[0].hist(n)

axes[1].set_title("Cumulative detailed histogram")
axes[1].hist(n,cumulative=True,bins=50)
```

[18]:
```
(array([1.0000e+00, 2.0000e+00, 8.0000e+00, 1.9000e+01, 3.5000e+01,
        6.0000e+01, 9.9000e+01, 1.8900e+02, 3.1700e+02, 5.2700e+02,
        8.5000e+02, 1.2960e+03, 1.9470e+03, 2.9550e+03, 4.3360e+03,
        6.1610e+03, 8.4820e+03, 1.1451e+04, 1.5017e+04, 1.9252e+04,
        2.4189e+04, 2.9768e+04, 3.5902e+04, 4.2389e+04, 4.9202e+04,
        5.6082e+04, 6.2578e+04, 6.8676e+04, 7.4279e+04, 7.9315e+04,
        8.3750e+04, 8.7509e+04, 9.0655e+04, 9.3108e+04, 9.5147e+04,
        9.6636e+04, 9.7722e+04, 9.8431e+04, 9.8972e+04, 9.9362e+04,
        9.9620e+04, 9.9786e+04, 9.9877e+04, 9.9923e+04, 9.9959e+04,
        9.9980e+04, 9.9993e+04, 9.9997e+04, 9.9998e+04, 1.0000e+05]),
 array([-4.24444459, -4.07555906, -3.90667354, -3.73778801, -3.56890248,
        -3.40001696, -3.23113143, -3.0622459 , -2.89336037, -2.72447485,
        -2.55558932, -2.38670379, -2.21781827, -2.04893274, -1.88004721,
        -1.71116168, -1.54227616, -1.37339063, -1.2045051 , -1.03561958,
        -0.86673405, -0.69784852, -0.52896299, -0.36007747, -0.19119194,
        -0.02230641,  0.14657911,  0.31546464,  0.48435017,  0.6532357 ,
         0.82212122,  0.99100675,  1.15989228,  1.3287778 ,  1.49766333,
         1.66654886,  1.83543439,  2.00431991,  2.17320544,  2.34209097,
         2.51097649,  2.67986202,  2.84874755,  3.01763308,  3.1865186 ,
         3.35540413,  3.52428966,  3.69317518,  3.86206071,  4.03094624,
         4.19983177]),
 <BarContainer object of 50 artists>)
```

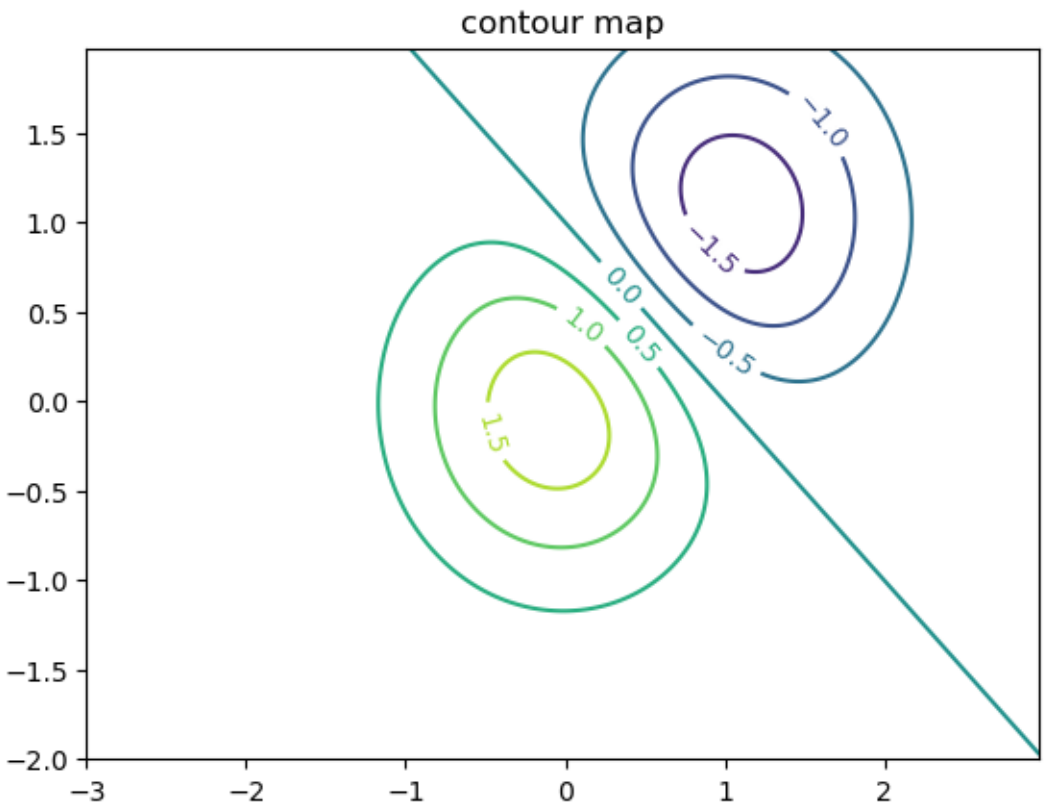


[19]:
```
# Draw contour image

import matplotlib
import numpy as np
import matplotlib.cm as cm
import matplotlib.pyplot as plt

delta = 0.025
x = np.arange(-3.0,3.0,delta)
y = np.arange(-2.0,2.0,delta)
X,Y = np.meshgrid(x,y)
Z1 = np.exp(-X**2 - Y**2)
Z2 = np.exp(-(X-1)**2-(Y-1)**2)
Z = (Z1 - Z2)*2

print(X)
print(Y)
```

```
[[-3.    -2.975 -2.95  …  2.925  2.95   2.975]
 [-3.    -2.975 -2.95  …  2.925  2.95   2.975]
 [-3.    -2.975 -2.95  …  2.925  2.95   2.975]
 …
 [-3.    -2.975 -2.95  …  2.925  2.95   2.975]
 [-3.    -2.975 -2.95  …  2.925  2.95   2.975]
 [-3.    -2.975 -2.95  …  2.925  2.95   2.975]]
[[-2.    -2.    -2.    … -2.    -2.    -2.   ]
 [-1.975 -1.975 -1.975 … -1.975 -1.975 -1.975]
 [-1.95  -1.95  -1.95  … -1.95  -1.95  -1.95 ]
 …
 [ 1.925  1.925  1.925 …  1.925  1.925  1.925]
 [ 1.95   1.95   1.95  …  1.95   1.95   1.95 ]
 [ 1.975  1.975  1.975 …  1.975  1.975  1.975]]
```

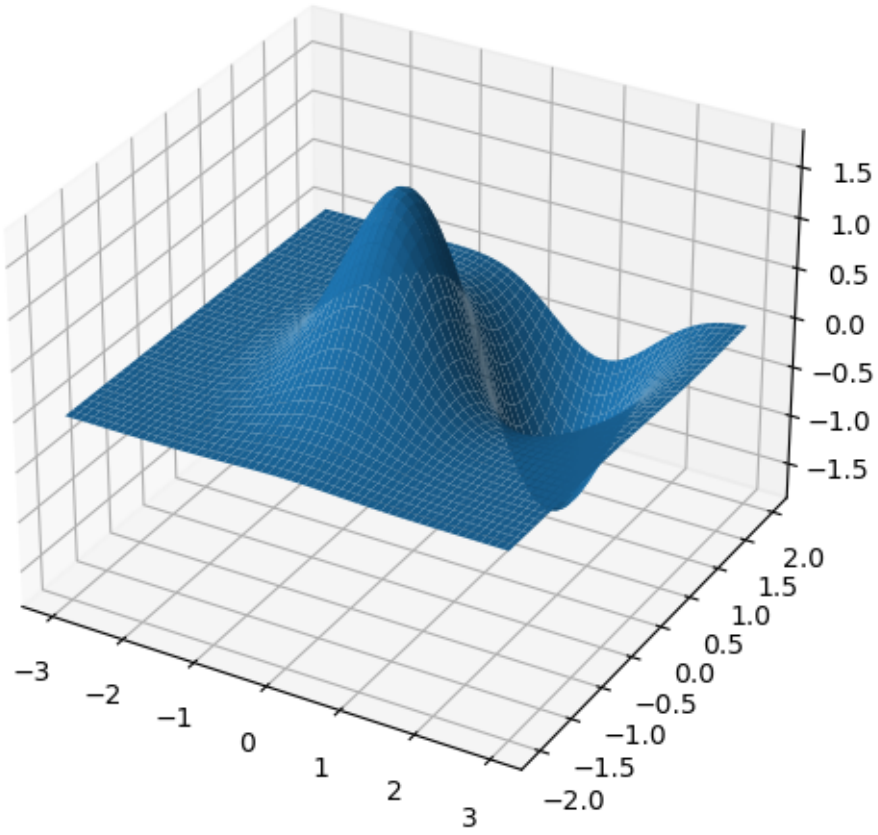
[20]:
```
fig,ax = plt.subplots()
CS = ax.contour(X,Y,Z)
ax.clabel(CS,inline=1,fontsize=10)
ax.set_title('contour map')
```

[20]:
```
Text(0.5, 1.0, 'contour map')
```



[21]:
```
# Draw 3D surface image
from mpl_toolkits.mplot3d.axes3d import Axes3D

fig = plt.figure(figsize=(14,6))

#Specify the 3D graphics to draw,with projection='3d'
ax = fig.add_subplot(1,2,1,projection='3d')
ax.plot_surface(X,Y,Z,rstride=4,cstride=4,linewidth=0)
```

[21]: <mpl_toolkits.mplot3d.art3d.Poly3DCollection at 0x18564408ca0>



[22]:
```
# heatmap ... color map
```

4 Practice Example

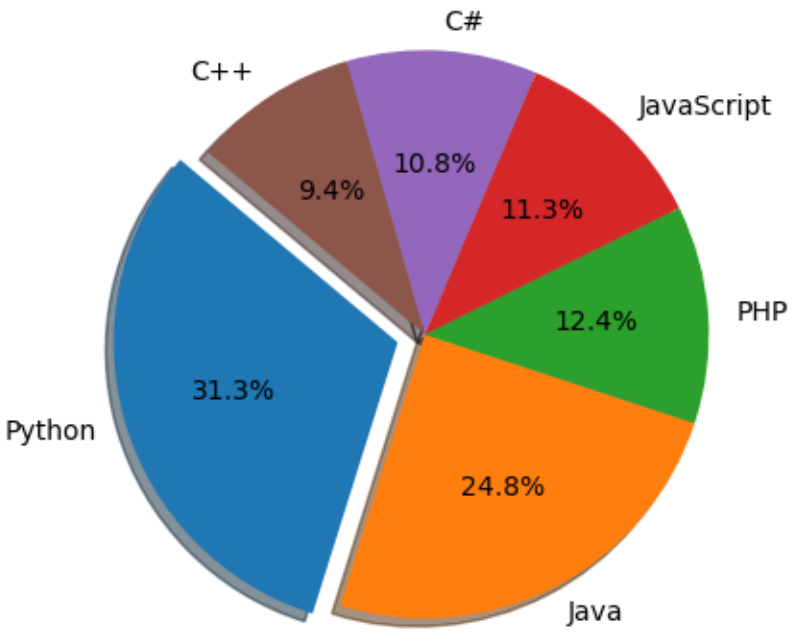
5 Write a Python programming to create a pie chart of the popularity of programming Languages

[23]:
```
import matplotlib.pyplot as plt
# Data to plot
languages = 'Python','Java','PHP','JavaScript','C#','C++'
popularity = [22.2,17.6,8.8,8,7.7,6.7]
colors = ["#1f77b4","#ff7f0e","#2ca02c","#d62728","#9467bd","#8c564b"]
languages
```

[23]:
```
('Python', 'Java', 'PHP', 'JavaScript', 'C#', 'C++')
```

[24]:
```
# explode 1st slice
explode = (0.1,0,0,0,0,0)
plt.pie(popularity,explode=explode,labels=languages,colors=colors,
        autopct='%1.1f%%',shadow=True,startangle=140)
```

[24]:
```
([<matplotlib.patches.Wedge at 0x185647ccbe0>,
  <matplotlib.patches.Wedge at 0x185647ccb20>,
  <matplotlib.patches.Wedge at 0x185647cda50>,
  <matplotlib.patches.Wedge at 0x185647ce320>,
  <matplotlib.patches.Wedge at 0x185647cebf0>,
  <matplotlib.patches.Wedge at 0x185647cf4c0>],
 [Text(-1.1518739051683529, -0.33643202373170245, 'Python'),
  Text(0.5025192070582963, -0.978506232242545, 'Java'),
  Text(1.0971674240514186, 0.07889007288863878, 'PHP'),
  Text(0.754341041824552, 0.8006057660415953, 'JavaScript'),
  Text(0.06701830757132049, 1.0979565321315212, 'C#'),
  Text(-0.5993297985645449, 0.9223902604389219, 'C++')],
 [Text(-0.6719264446815391, -0.19625201384349306, '31.3%'),
  Text(0.2741013856681616, -0.5337306721322972, '24.8%'),
  Text(0.5984549585735011, 0.043030948848348426, '12.4%'),
  Text(0.41145875008611926, 0.4366940542045065, '11.3%'),
  Text(0.036555440493447534, 0.5988853811626479, '10.8%'),
  Text(-0.3269071628533881, 0.5031219602394119, '9.4%')])
```



[ ]: