Untitled

June 6, 2021

1 MATPLOTLIB tutorial

[2]:
```
from matplotlib import pylab
print (pylab.__version__)
```

```
1.19.2
```

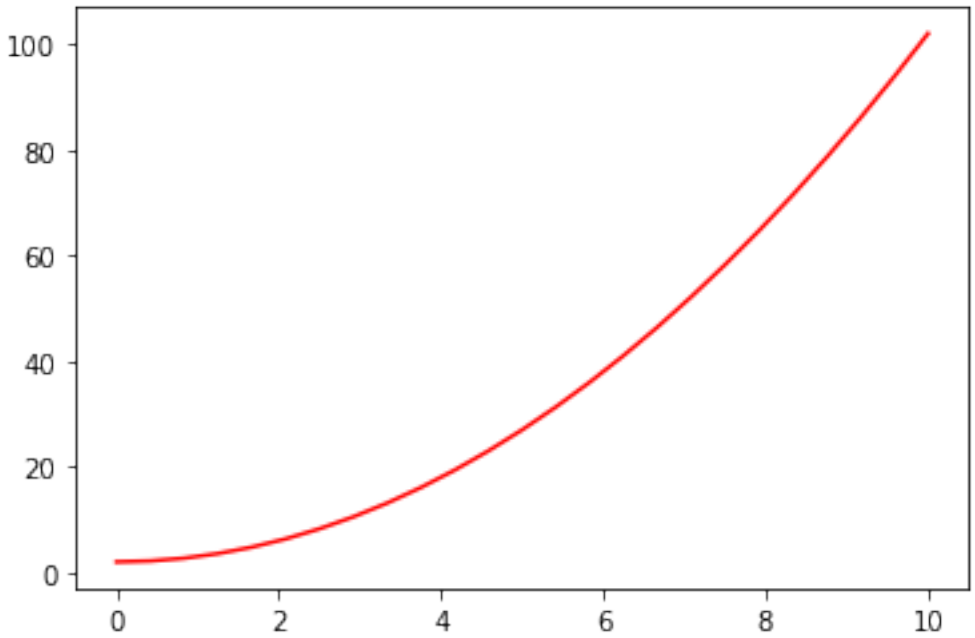
[8]:
```
import numpy as np

x = np.linspace(0, 10, 25)
print (x)
y = x**2 + 2
print (y)
print (np.array([x, y]).reshape(2, 25))
print (np.array([x, y]).reshape(25, 2))
```

```
[ 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.        ]]
[[  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.        ]]
```

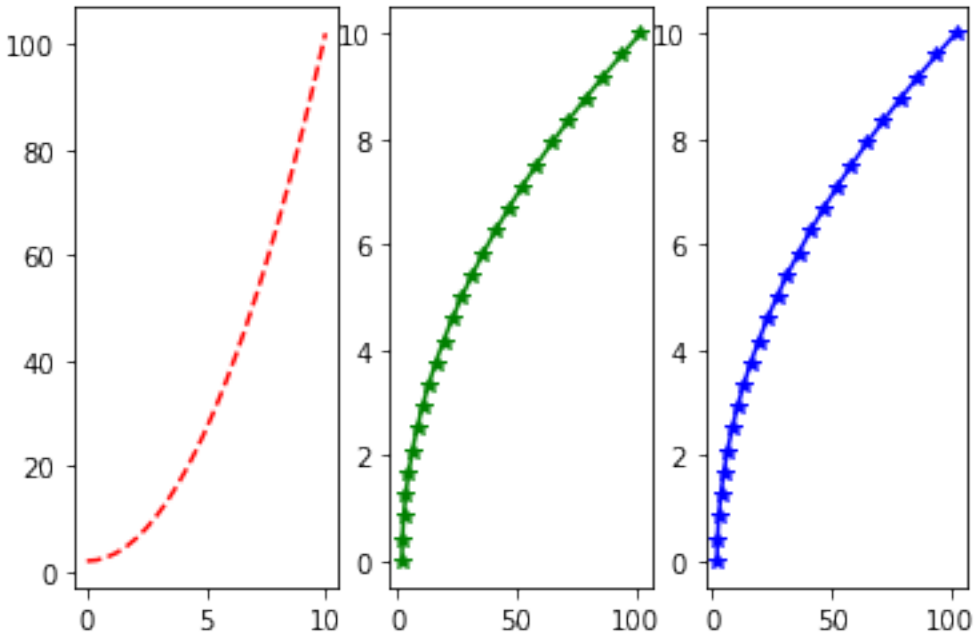
[9]:
```
# only one command to draw

pylab.plot(x, y, 'r') # r stands for red
```

[9]: [<matplotlib.lines.Line2D at 0x279cc833dc0>]



[29]:
```
# drawing subgraphs
pylab.subplot(1, 3, 1) # the contents of the brackets represnt (raws, columns,␣
 ↪indexes)
pylab.plot(x, y, 'r--') # The third parametr here determines color and linestile
pylab.subplot(1, 3, 2)
pylab.plot(y, x, 'g*-')
pylab.subplot(1, 3, 3)
pylab.plot(y, x, 'b*-')
```

[29]:
```
[<matplotlib.lines.Line2D at 0x279ce6a79a0>]
```



2 Operator describtion

fig.add_axes() = initialize subplots a = fig.add_subplot(222)

fig, b = plt.subplots(nrows= 3, ncols=2) = adds subplots

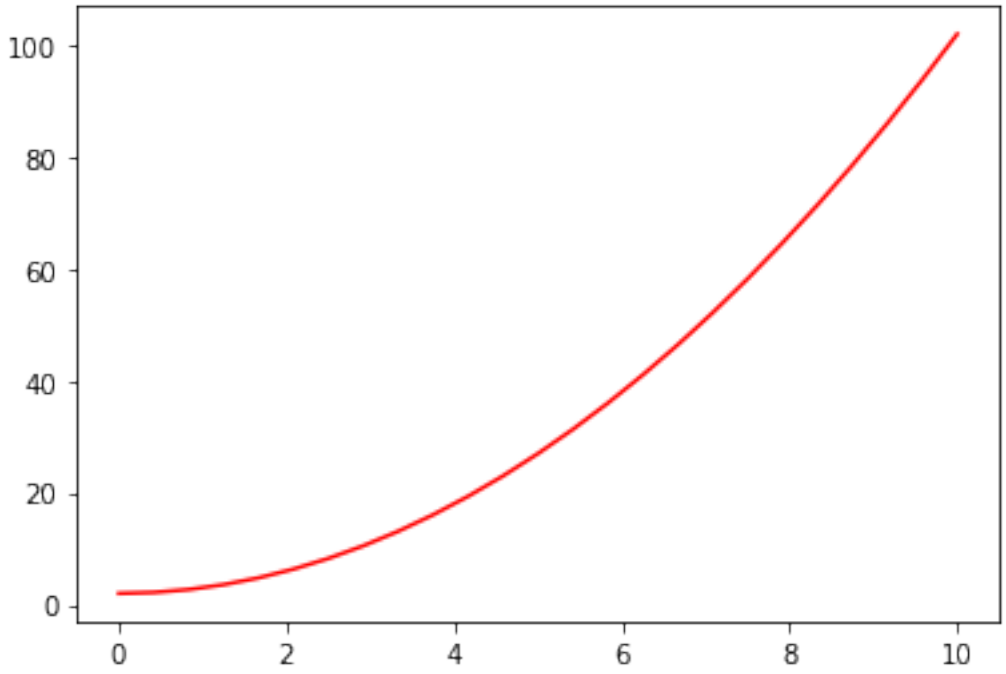
ax = plt.subplots(2, 2) = Create subplots

[30]:
```
from matplotlib import pyplot as plt
```

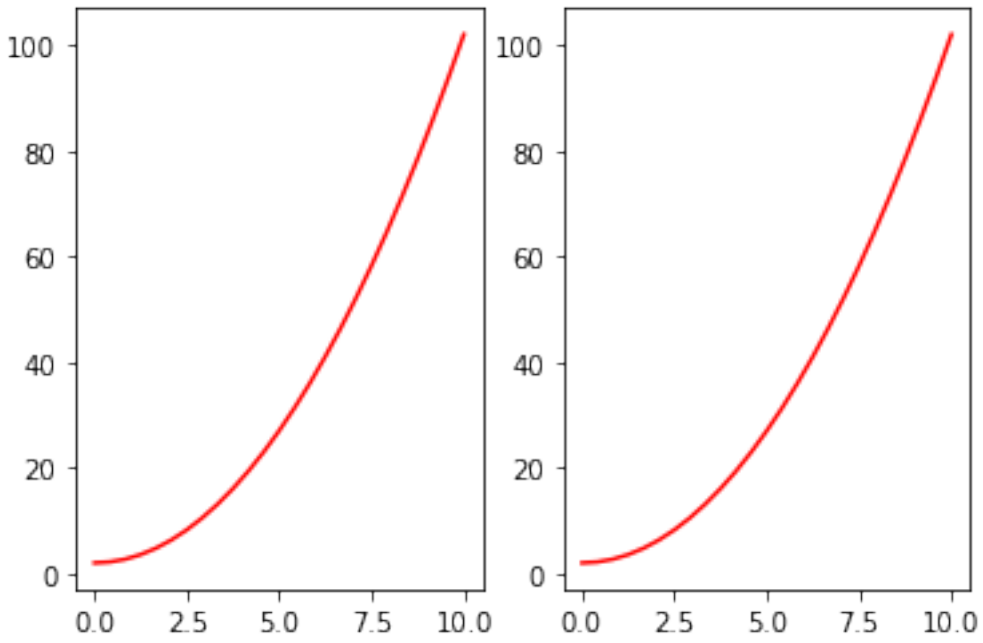
[34]:
```
fig = plt.figure()
```

```
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')
```

[34]: [<matplotlib.lines.Line2D at 0x279cf7db130>]

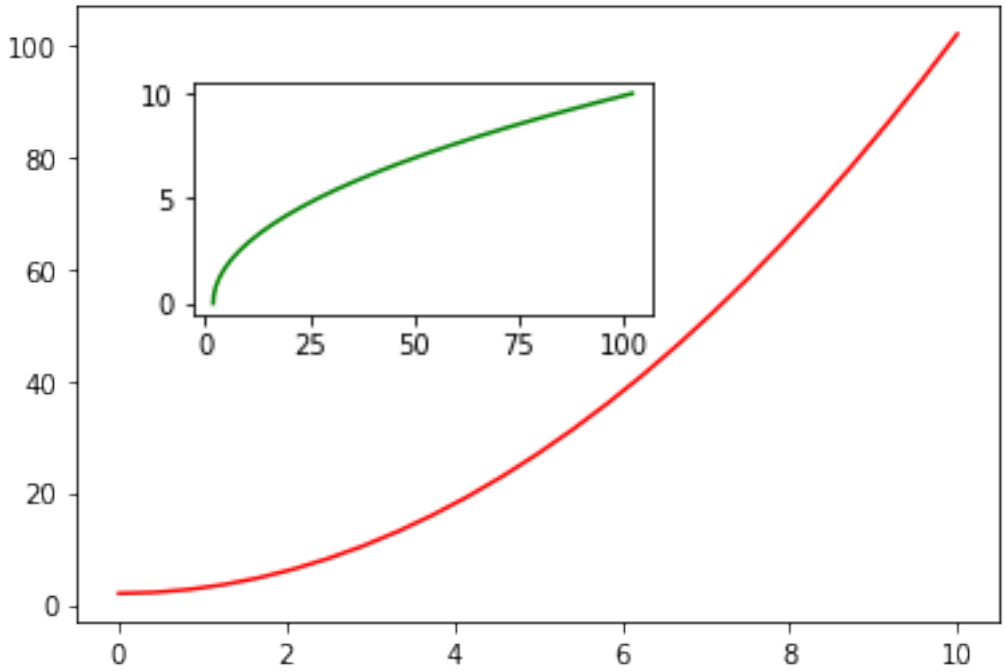


[37]:
```
# again we can draw some subgraphs
fig, axes = plt.subplots(nrows=1, ncols=2) # submap is of 1 row and 2 columns
for ax in axes:
    ax.plot(x, y, 'r')
```



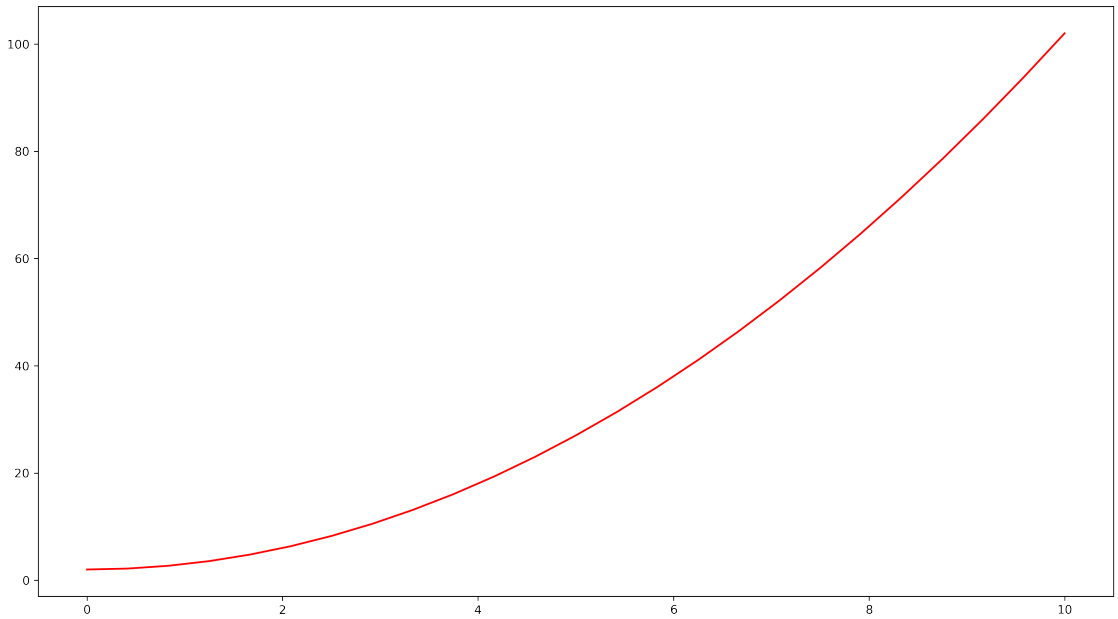
[41]:
```
# We can also draw a picture or graph inside another graph

fig = plt.figure()
axes1 = fig.add_axes([0.1, 0.1, 0.8, 0.8])  # big axes
axes2 = fig.add_axes([0.2, 0.5, 0.4, 0.3])  # small canvas

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

[41]:
```
[<matplotlib.lines.Line2D at 0x279ce6d9eb0>]
```



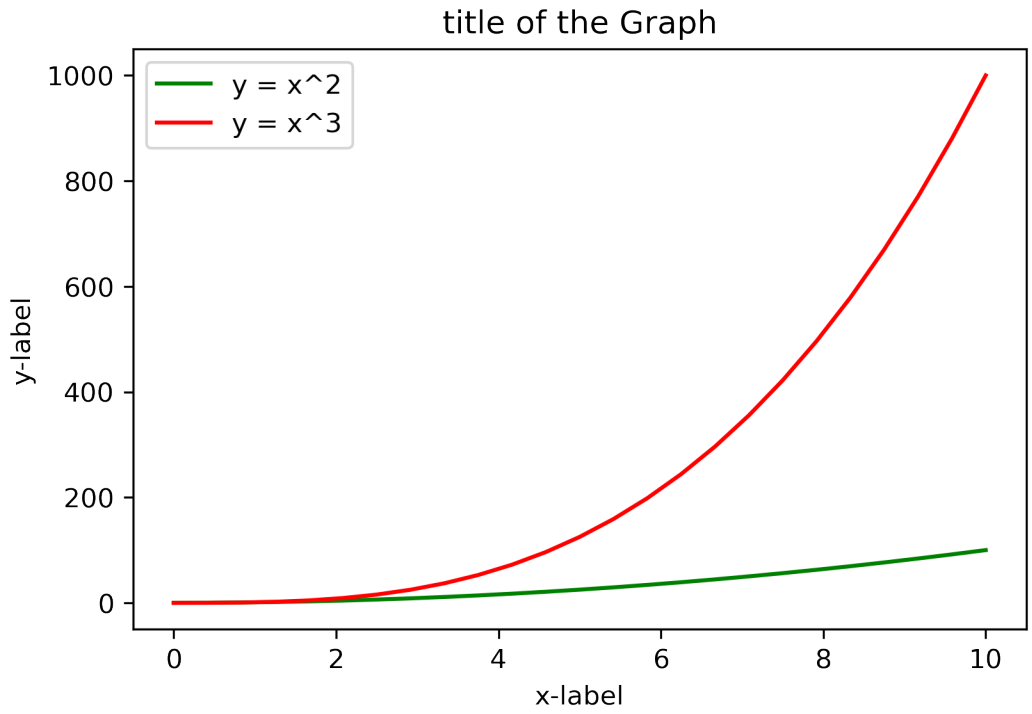
[43]:
```
fig = plt.figure(figsize=(16, 9), dpi=300) # new graphic object

fig.add_subplot()

plt.plot(x, y, 'r')
```

[43]: [<matplotlib.lines.Line2D at 0x279cface6a0>]



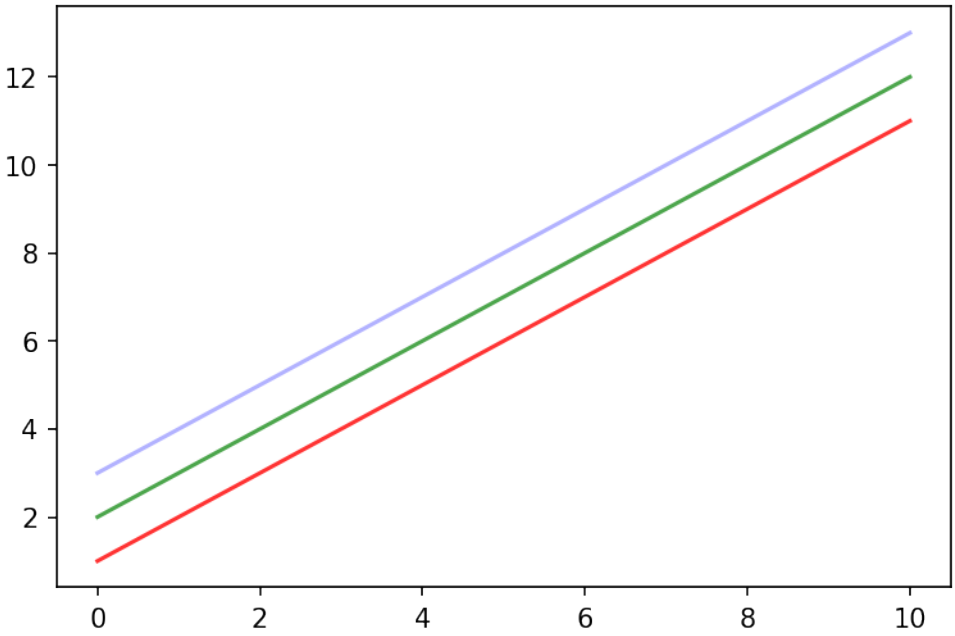
[47]:
```
#ax.legend(['label1', 'label2'])

fig, axes = plt.subplots(dpi=300)
axes.set_ylabel('y-label')
axes.set_xlabel('x-label')
axes.set_title('title of the Graph')

axes.plot(x, x**2, 'g')
axes.plot(x, x**3, 'r')

axes.legend(['y = x^2', 'y = x^3'], loc= 2)
```

[47]:
```
<matplotlib.legend.Legend at 0x279cfb3bc70>
```



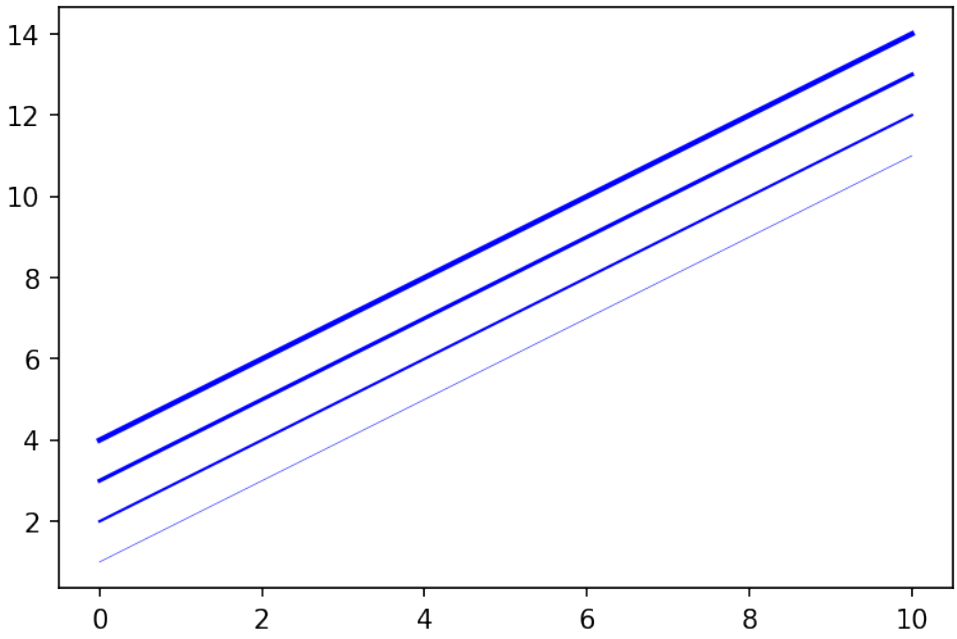
[50]:
```
# in matplotlib you can set another properties such as line color, transparency␣
↪and more

fig, axes = plt.subplots(dpi = 150)
axes.plot(x, x+1, color='red', alpha=.8)
axes.plot(x, x+ 2, color='green', alpha=.7)
axes.plot(x, x+3, color='blue', alpha=.3)
```

[50]: [<matplotlib.lines.Line2D at 0x279d031e5e0>]

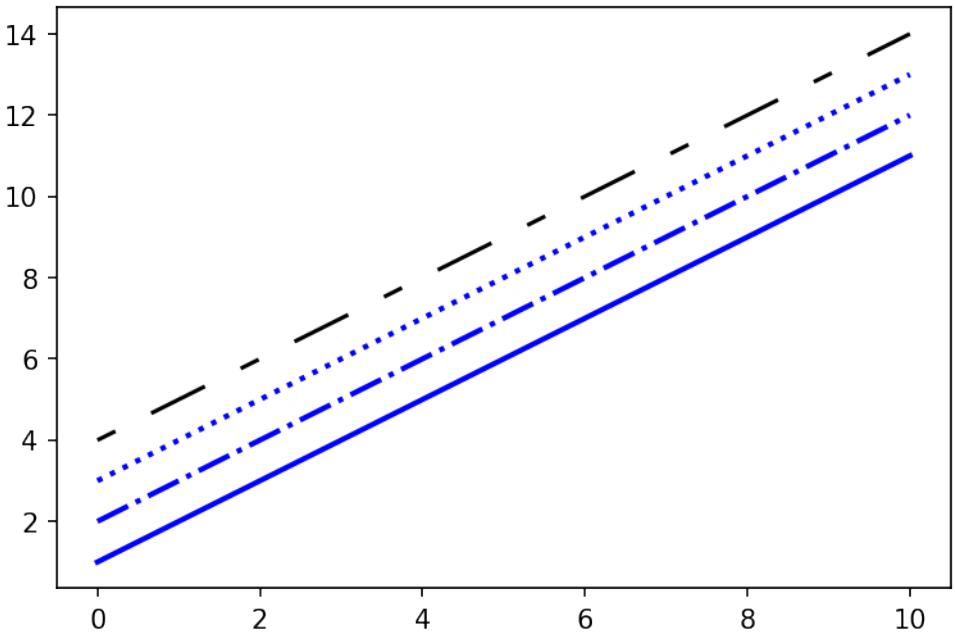


[52]:
```
fig, axes = plt.subplots(dpi=150)
# varify the line width
axes.plot(x, x+1, color='blue', linewidth=0.25)
axes.plot(x, x+2, color='blue', linewidth=1)
axes.plot(x, x+3, color='blue', linewidth=1.5)
axes.plot(x, x+4, color='blue', linewidth=2)
```

[52]: [<matplotlib.lines.Line2D at 0x279cf6e3610>]

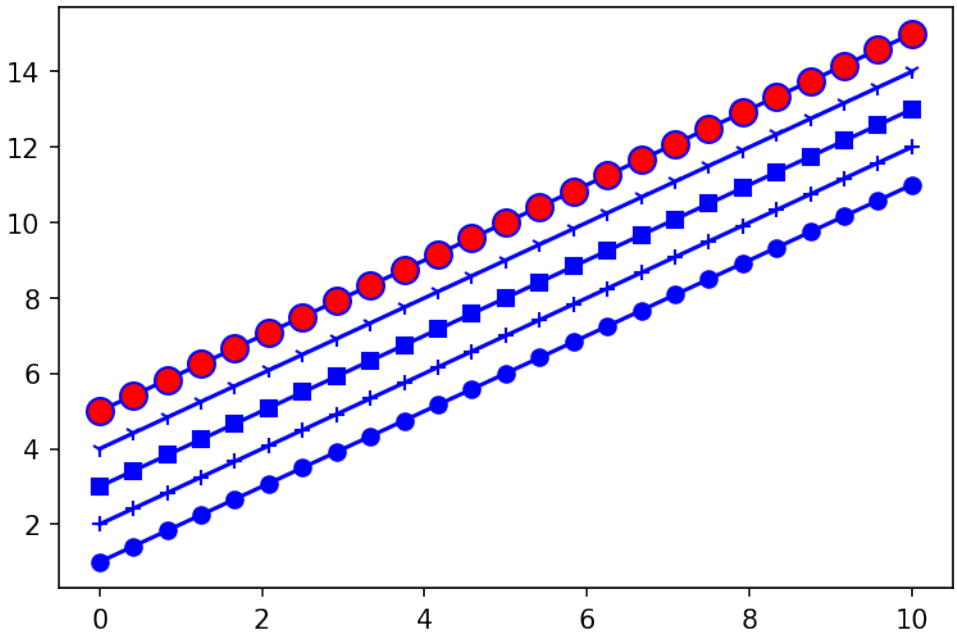


[57]:
```
fig, axes = plt.subplots(dpi=150)
axes.plot(x, x+1, color='blue', lw=2, linestyle='-')
axes.plot(x, x+2, color='blue', lw=2, linestyle='-.')
axes.plot(x, x+3, color='blue', lw=2, linestyle=':')

line, = axes.plot(x, x+4, color='black', lw=1.5)
line.set_dashes([5, 10, 15, 10])
```



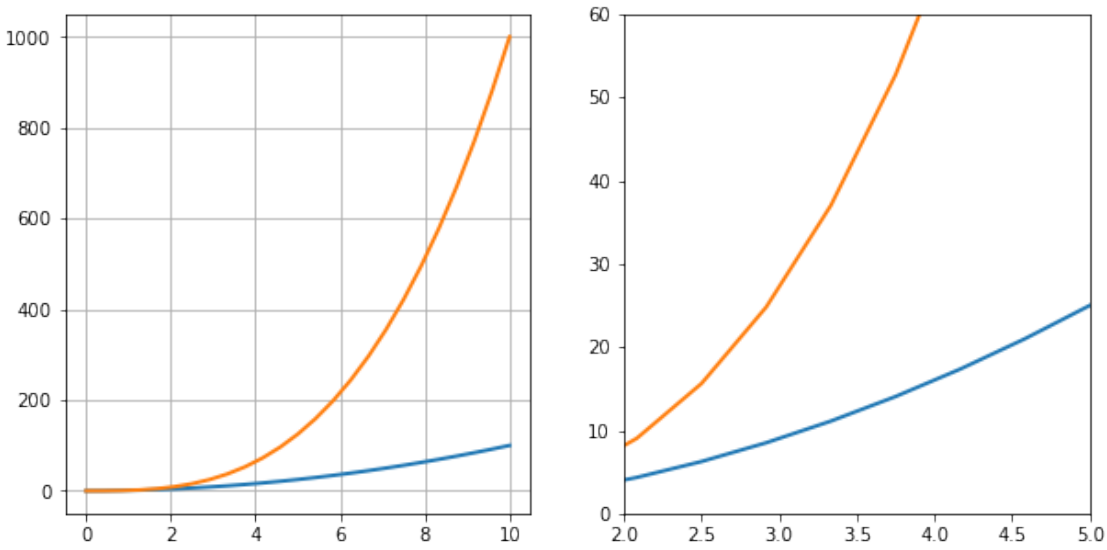
[62]:
```
fig, axes = plt.subplots(dpi=150)
# varify the line width
axes.plot(x, x+1, color='blue', marker='o')
axes.plot(x, x+2, color='blue', marker='+')
axes.plot(x, x+3, color='blue', marker='s')
axes.plot(x, x+4, color='blue', marker='1')

axes.plot(x, x+5, color='blue', marker='o', markersize=10,␣
 ↪markerfacecolor='red')
```

[62]:
```
[<matplotlib.lines.Line2D at 0x279ce534c70>]
```



[66]:
```
# set the canvas grid and axis range
fig, ax = plt.subplots(1, 2, figsize=(10, 5))
ax[0].plot(x, x**2, x, x**3, lw=2)
ax[0].grid(True)

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

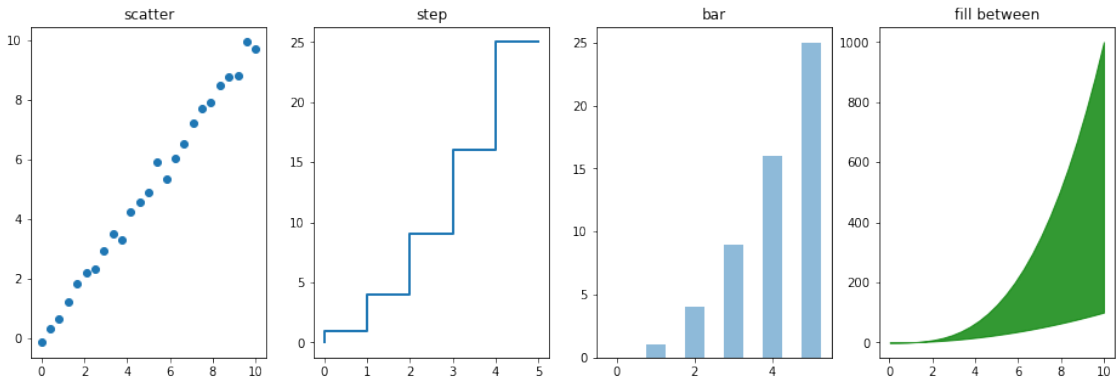
[66]: (2.0, 5.0)



3 Other 2D graphs

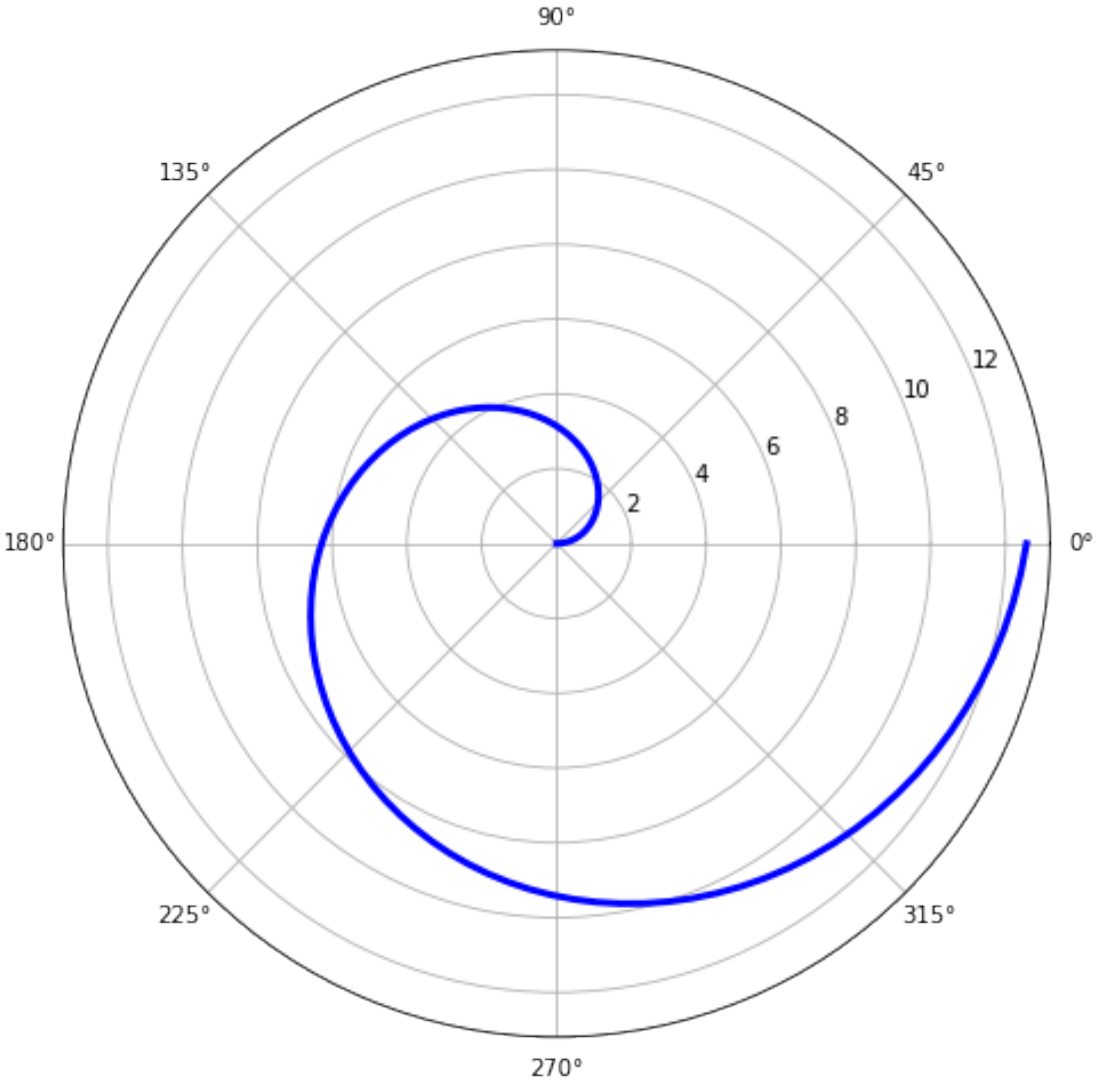
[72]:
```
n = np.array([0, 1, 2, 3, 4, 5])

fig, ax = plt.subplots(1, 4, figsize=(16, 5))
ax[0].set_title('scatter')
ax[0].scatter(x, x+0.25*np.random.randn(len(x)))

ax[1].set_title('step')
ax[1].step(n, n**2, lw=2)

ax[2].set_title('bar')
ax[2].bar(n, n**2, align='center', width=0.5, alpha=0.5)

ax[3].set_title('fill between')
ax[3].fill_between(x, x**2, x**3, color='green', alpha=0.8)
```

[72]:
```
<matplotlib.collections.PolyCollection at 0x279d5e9d430>
```



[80]:
```
# draw a radar chart

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

ax = fig.add_axes([0.0, 0.0, 1, 1], polar=True)
t = np.linspace(0 , 2*np.pi, 100)
ax.plot(t, t*2, lw=3, color='blue')
```

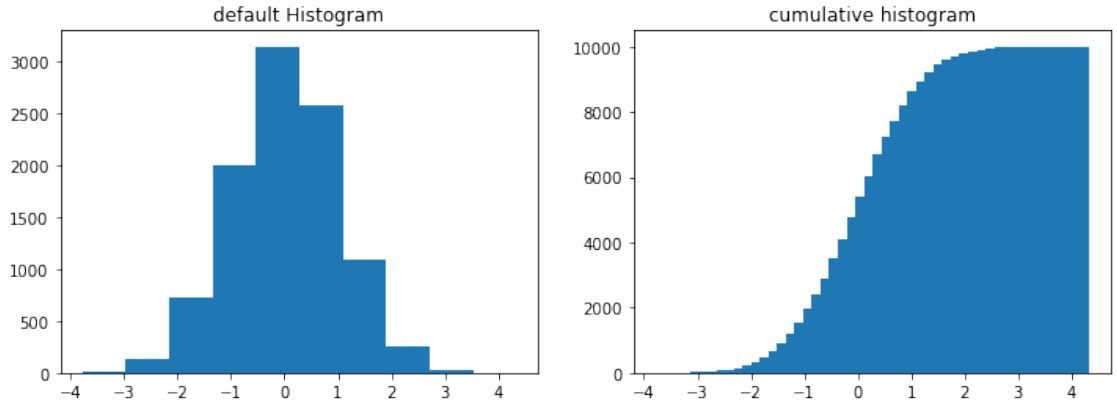
[80]: [<matplotlib.lines.Line2D at 0x279d0583970>]



[86]:
```
# draw a histogram

n = np.random.randn(10000)
fig, ax = plt.subplots(1, 2, figsize=(12,4))
ax[0].hist(n)
ax[0].set_title('default Histogram')

ax[1].set_title('cumulative histogram')
ax[1].hist(n, cumulative=True, bins=50)
```

[86]:
```
(array([1.000e+00, 1.000e+00, 3.000e+00, 7.000e+00, 1.500e+01, 2.300e+01,
        3.900e+01, 6.700e+01, 9.900e+01, 1.530e+02, 2.390e+02, 3.400e+02,
        4.650e+02, 6.560e+02, 8.860e+02, 1.181e+03, 1.542e+03, 1.963e+03,
```

```
       2.427e+03, 2.890e+03, 3.495e+03, 4.110e+03, 4.752e+03, 5.386e+03,
       6.033e+03, 6.704e+03, 7.240e+03, 7.717e+03, 8.183e+03, 8.610e+03,
       8.938e+03, 9.197e+03, 9.440e+03, 9.598e+03, 9.706e+03, 9.803e+03,
       9.863e+03, 9.913e+03, 9.948e+03, 9.966e+03, 9.984e+03, 9.991e+03,
       9.996e+03, 9.998e+03, 9.998e+03, 9.998e+03, 9.999e+03, 9.999e+03,
       9.999e+03, 1.000e+04]),
array([-3.78487282, -3.62242683, -3.45998085, -3.29753486, -3.13508887,
       -2.97264289, -2.8101969 , -2.64775091, -2.48530493, -2.32285894,
       -2.16041296, -1.99796697, -1.83552098, -1.673075  , -1.51062901,
       -1.34818302, -1.18573704, -1.02329105, -0.86084507, -0.69839908,
       -0.53595309, -0.37350711, -0.21106112, -0.04861514,  0.11383085,
        0.27627684,  0.43872282,  0.60116881,  0.7636148 ,  0.92606078,
        1.08850677,  1.25095275,  1.41339874,  1.57584473,  1.73829071,
        1.9007367 ,  2.06318269,  2.22562867,  2.38807466,  2.55052064,
        2.71296663,  2.87541262,  3.0378586 ,  3.20030459,  3.36275057,
        3.52519656,  3.68764255,  3.85008853,  4.01253452,  4.17498051,
        4.33742649]),
<BarContainer object of 50 artists>)
```



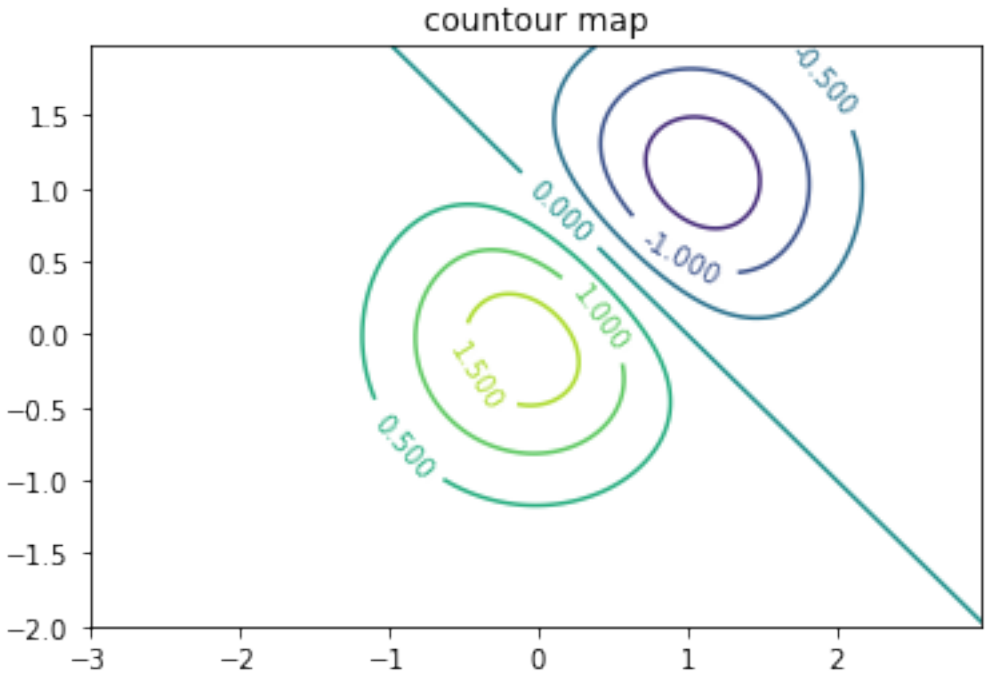
[88]:
```
# Draw contour image

import matplotlib
import numpy as np
import matplotlib.cm as cm  # colormaps
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]]
```

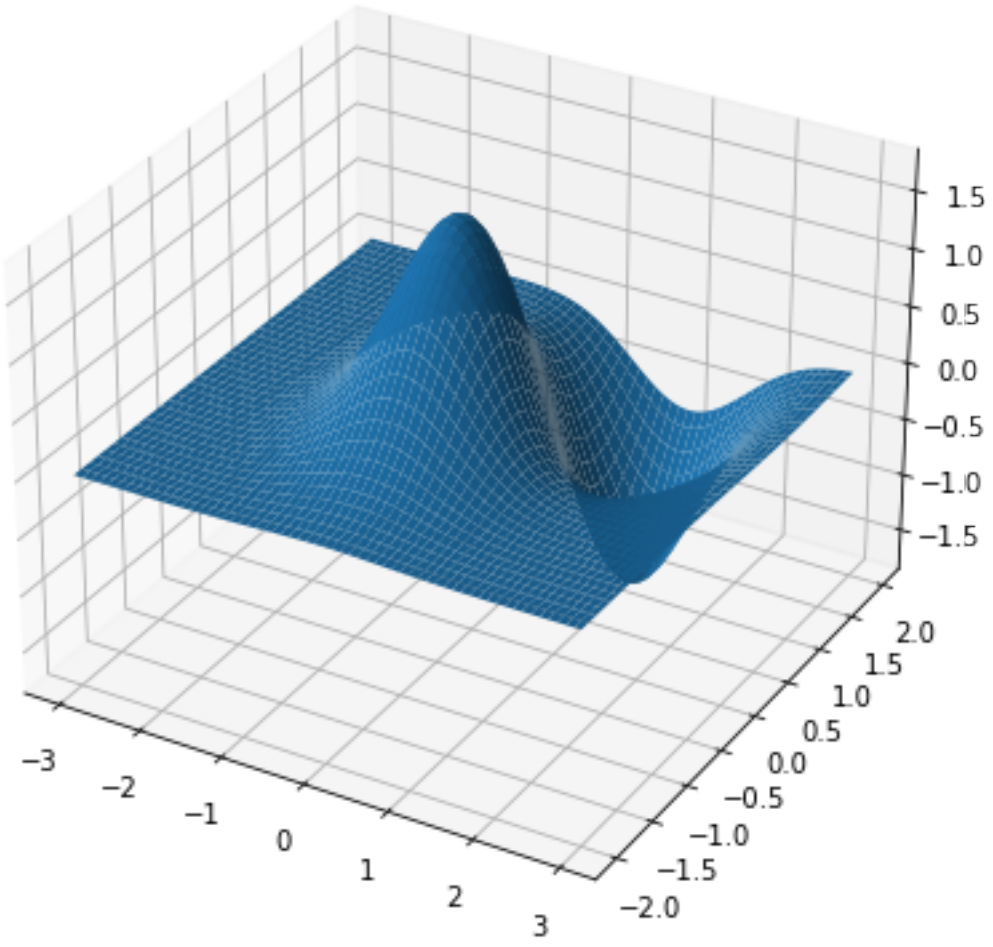
[92]:
```
fig, ax =plt.subplots()
CS = ax.contour(X, Y, Z)
ax.clabel(CS, inline=1, fontsize=10)
ax.set_title('countour map')
```

[92]: Text(0.5, 1.0, 'countour map')



[102]:
```
# Draw a 3D surface image
from mpl_toolkits.mplot3d.axes3d import Axes3D

fig = plt.figure(figsize = (19, 7))

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

[102]:
```
<mpl_toolkits.mplot3d.art3d.Poly3DCollection at 0x279ce6a7b20>
```

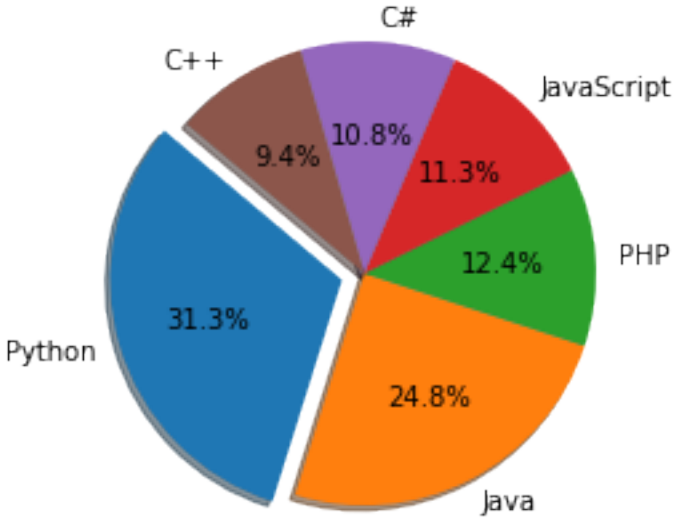


[103]:
```
# heat map
```

4 practice

[106]:
```
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]

#explode 1st slice
explode = (0.1, 0, 0, 0, 0, 0)
plt.pie(popularity, explode=explode, labels=languages, autopct='%1.1f%%',␣
 ↪shadow=True, startangle=140)
```

[106]:
```
([<matplotlib.patches.Wedge at 0x279d7e85dc0>,
  <matplotlib.patches.Wedge at 0x279d7e8b730>,
  <matplotlib.patches.Wedge at 0x279d7e8bfd0>,
  <matplotlib.patches.Wedge at 0x279d7e93940>,
  <matplotlib.patches.Wedge at 0x279d7e9a250>,
  <matplotlib.patches.Wedge at 0x279d7e9ab20>],
 [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%')])
```


C#
C++
JavaScript
10.8%
9.4%
11.3%
PHP
12.4%
31.3%
Python
24.8%
Java

[ ]: