

Matplotlib-1

November 19, 2019

Matplotlib module

https://matplotlib.org/api/_as_gen/matplotlib.pyplot.html

```
In [86]: import matplotlib as mpl
import matplotlib.pyplot as plt # import the module
mpl.rcParams['figure.dpi'] = 130 # set the resolution to x dpi

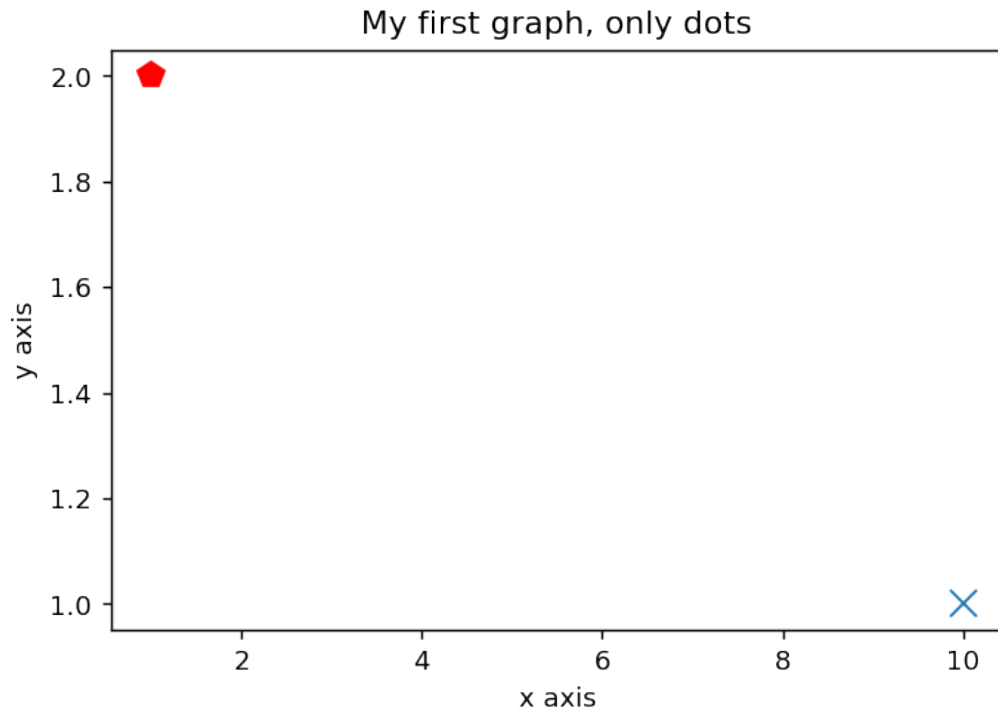
In [87]: plt.figure() # create a new figure, not strictly necessary

plt.title("My first graph, only dots") # the title to give to the figure
plt.xlabel("x axis") # x-axis label
plt.ylabel("y axis") # y-axis label

plt.plot(1, 2, marker='p', color='red', markersize=10)

plt.plot(10, 1, marker='x', markersize=10)

plt.show()
```



```
In [88]: plt.figure() # create a new figure, not strictly necessary

plt.title("My first graph, only dots") # the title to give to the figure
plt.xlabel("x axis") # x-axis label
plt.ylabel("y axis") # y-axis label

# each command plots an (x,y) point with a color and a marker
#
plt.plot(1, 1, marker='.', color='blue')
plt.plot(2, 4, marker='o', color='red')

plt.plot(3, 9, 'b.') # short form, b stands for blue, . is marker
plt.plot(4, 16, 'ro') # short form, r stands for red, o is marker
plt.plot(25, 'g+')

yy = [2,5,7,11,13]
plt.plot(yy, 'rx')

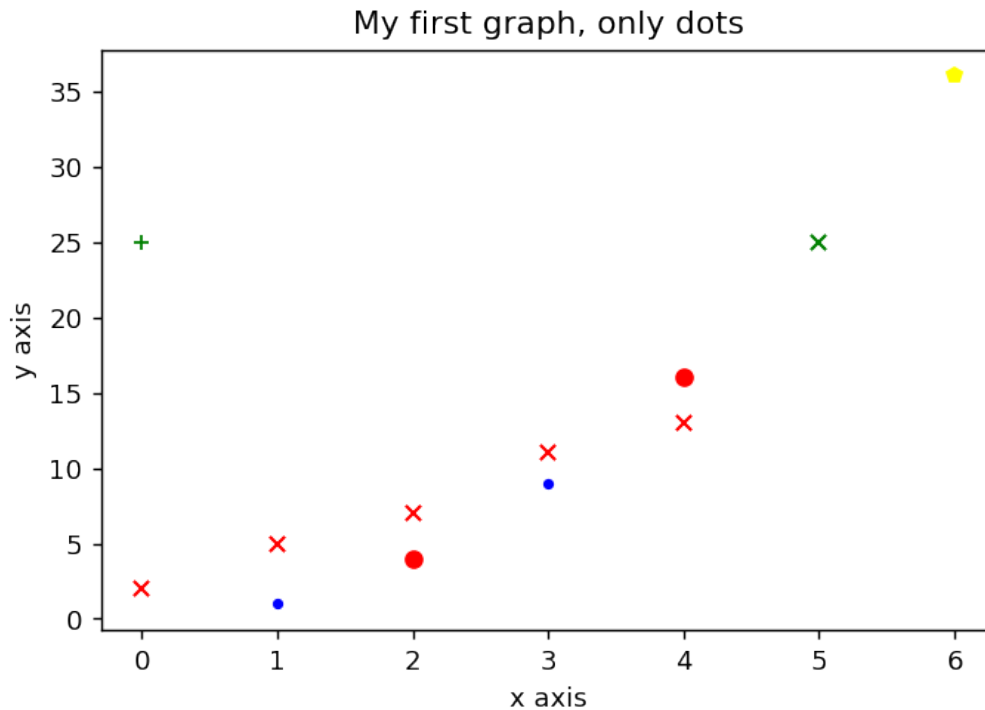
xx = [0,1,2,3,4]
plt.plot(xx, yy, 'rx')

plt.plot(5, 25, marker='x', color='green')
plt.plot(5, 25, marker='x', color='green')
```

```
plt.plot(6, 36, marker='p', color='yellow')

plt.show() # show in the figure all the issued plot commands

#plt.plot([1,2,3,4,5],[1,4,9,16,25], color="red")
#plt.show()
```



```
In [89]: # Many different markers are available for the shape of the data points
# =====
# character      description
# =====
# ``'.'``       point marker
# ``','``       pixel marker
# ``'o'``       circle marker
# ``'v'``       triangle_down marker
# ``'^'``       triangle_up marker
# ``'<'``       triangle_left marker
# ``'>'``       triangle_right marker
# ``'1'``       tri_down marker
# ``'2'``       tri_up marker
# ``'3'``       tri_left marker
# ``'4'``       tri_right marker
# ``'s'``       square marker
# ``'p'``       pentagon marker
```

```

#   ``*``      star marker
#   ``h``      hexagon1 marker
#   ``H``      hexagon2 marker
#   ``+``      plus marker
#   ``x``      x marker
#   ``D``      diamond marker
#   ``d``      thin_diamond marker
#   ``/``      vline marker
#   ``_``      hline marker
#   =====

```

```
In [90]: import numpy as np # import numpy to use arange and math functions
```

```
plt.figure(figsize=(5, 3)) # create a new figure with a given size
```

```
plt.title("Plot (x,y) sequences with lines")
```

```
plt.xlabel("x")
```

```
plt.ylabel("y")
```

```
x = np.arange(-5, 6, 1) # 11 x coordinate points
```

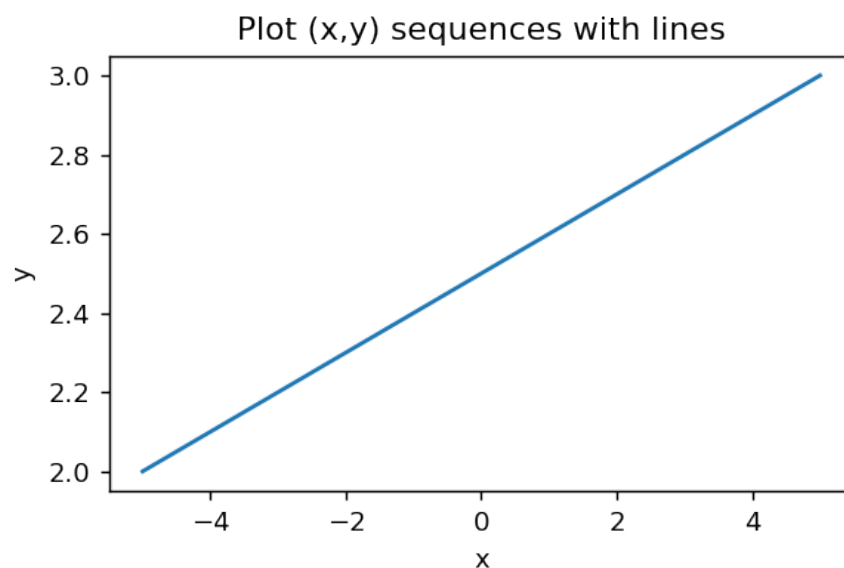
```
y = np.arange(2, 3.1, 0.1) # 11 y coordinate points
```

```
#plt.plot(x, y, marker='.', color='blue')
```

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

```
# with lists, the default behavior is to plot with connecting lines
```

```
plt.show()
```



```
In [91]: plt.figure(figsize=(2, 3)) # create a new figure with a given size

plt.title("How to control x,y limits and tick marks\n")
plt.xlabel("x")
plt.ylabel("y")

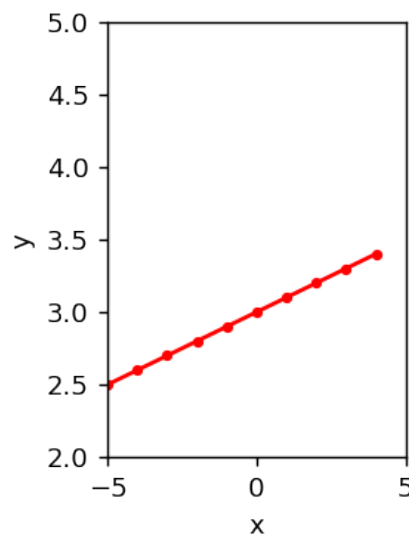
x = range(-5, 5, 1) # 10 x coordinate points
plt.xlim(-5, 5) # limits for the x axis

y = np.arange(2.5, 3.5, 0.1)
plt.ylim(2, 5) # limits for the y axis
plt.yticks(np.arange(2, 5.5, 0.5)) # where to place the tick marks on y axis

plt.plot(x, y, marker='.', color='red')

plt.show()
```

How to control x,y limits and tick marks



```
In [92]: import random as rnd

plt.figure(figsize=(5, 3)) # create a new figure with a given size

plt.title("Plot randomly generated NASDAQ data, use textual labels\n", fontsize=10)
plt.xlabel("Month (Year 2050)", fontsize=9)
plt.ylabel("NASDAQ", fontsize=9)

# we want to plot a randomly generated value for NASDAQ for each month of the year
x = range(1, 13, 1) # 12 months in a year
```

```

# a list of textual labels to place at the tick marks on the x axis
months = ['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug',
          'Sep', 'Oct', 'Nov', 'Dec']

# the labels in list months are places at the positions defined by the list x
plt.xticks(x, months, fontsize=8)

# let's give a reasonable range for nasdaq values
nasdaq_min = 10000
nasdaq_max = 15000

# generate 12 random values for the NASDAQ one per each month, between min and max
y = []
for n in range(12):
    y.append(rnd.uniform(nasdaq_min, nasdaq_max))
    # the uniform(a,b) method from random module generates a random number
    # uniformly selected in the real interval range [a,b]

'''
#this is another way yo do it using the method sample() from module random
list_to_sample_from = list(np.arange(nasdaq_min, nasdaq_max, 0.25))
num_samples = 12
y = rnd.sample( list_to_sample_from, num_samples)
'''

# set the limits on the y axis based on the min and max values used for sampling
#plt.ylim(nasdaq_min, nasdaq_max)
#plt.ylim(12000, 13000)
# set the tick marcks for y axis every 500
plt.yticks(range(nasdaq_min, nasdaq_max+1, 500), fontsize=8)

plt.plot(x, y, color='red', marker='d', linestyle='--', linewidth=1)

plt.show()

```