

### TO START

```
conda install matplotlib
import matplotlib.pyplot as plt
# to print graphs in notebooks
%matplotlib inline
```

### BASICS

#### FUNCTIONAL METHOD

<code>plt.plot(x,y)</code>	simple line plot
<code>-plt.xlabel('str')</code>	set x label
<code>-plt.ylabel('str')</code>	set y label
<code>-plt.title('str')</code>	set title
<code>plt.show()</code>	show plot
<code>plt.subplot(r,c,1)*</code>	create multiplots
<code>plt.plot(x, y)</code>	
<code>plt.subplot(r,c,2)</code>	
<code>plt.plot(y, x, 'g*-')</code>	

#### OBJECT ORIENTED METHOD (more control)

<code>fig = plt.figure()</code>	create canvas
<code>ax = fig.add_axes([0,0,1,1])</code>	create axes*
<code>ax.plot(x, y, 'b')</code>	create plot
<code>ax.set_xlabel("str")</code>	set x label
<code>ax.set_ylabel("str")</code>	set y label
<code>ax.set_title("str")</code>	set title

\*\* add more axis to have more figures

<code>fig, ax = plt.subplots(r,c)*</code>	subplots
<code>axes[0].plot(x,y)</code>	create pl ax1
<code>axes[1].plot(x,y)</code>	create pl ax2
<code>axes[0].set_title('str')</code>	set plot 1 title
<code>axes[1].set_title("str")</code>	set plot 2 title

**subplot()** command requires to specify the number of row and column we want to print the plots, and the third parameter specify what of the graph we are going to handle.

**axes:** ([left, bottom, width, height])

**fig, axes** allow you to auto-manage axis, you don't have to create them. Axes, now, will be an array of axis. We could use **for** loop to populate labels on axis.

### SIZE, SAVE, LEGEND

<code>plt.tight_layout()</code>	avoid overlap
<code>plt.fig(figsize=(x,x))</code>	set figsize
<code>plt.fig(figsize=(x,x), dpi=x)</code>	set dpi
<code>fig.savefig("name.png")</code>	save figure
<code>fig.savefig("name", dpi=200)</code>	..and set dpi
<code>ax.plot(x, y, label="str")</code>	set legend
<code>ax.legend()</code>	show legend
<code>ax.legend(loc=0)</code>	best
<code>ax.legend(loc=1)</code>	upper right
<code>ax.legend(loc=2)</code>	upper left
<code>ax.legend(loc=3)</code>	lower left
<code>ax.legend(loc=4)</code>	lower right

**subplot()** command requires to specify the number of row and column we want to print the plots, and the third parameter specify what of the graph we are going to handle.

**axes:** ([left, bottom, width, height])

**fig, axes** allow you to auto-manage axis, you don't have to create them. Axes, now, will be an array of axis. We could use **for** loop to populate labels on axis.

### COLORS, LINEWIDTHS, LINETYPES

<code>fig = plt.figure()</code>	----
<code>ax = fig.add_axes([0,0,1,1])</code>	----
<code>ax.plot(x,y</code>	----
<code>color='#xxxxxx',</code>	set color
<code>lw=x,</code>	set linewidth
<code>alpha=x,</code>	set alpha
<code>ls=","</code>	set linestyle
<code>marker=","</code>	set markertype
<code>markersize=x,</code>	set marker size
<code>markerfacecolor=","</code>	set mark color
<code>markeredgecolor=","</code>	set external col
<code>markeredgewidth=x)</code>	set marker wdt
<code>ax.set_xlim([0,1])</code>	set x axes limit
<code>ax.set_ylim([0,1])</code>	set y axes limit
<code>ax.plot(x, y, 'r--')</code>	MATLAB style



By [DarioPittera \(aggialavura\)](#)

Not published yet.

Last updated 14th June, 2019.

Page 1 of 1.

Sponsored by [Readable.com](#)

Measure your website readability!

<https://readable.com>