

Function	Description
<code>plt.plot(x_values, y_values)</code>	<p>Creates a line. <code>x_values</code> and <code>y_values</code> are lists of numbers (i.e., <code>[1, 2, 3, 4]</code>).</p> <p>Also accepts the following keyword arguments:</p> <ul style="list-style-type: none">- <code>marker</code> - a symbol that will appear at each (x, y) point. Options include '*' for a star, 'o' for a circle, or 's' for a square.- <code>linestyle</code> - whether the line is solid ('-') or dashed ('--' or ':') or no line at all ('')- <code>linewidth</code> - a number representing the thickness of the line; default is 1- <code>color</code> - the color of the line (can be a HEX code or any html color name)
<code>plt.show()</code>	Displays any previous plot commands
<code>plt.close('all')</code>	Closes all previous figures
<code>plt.figure(figsize=(width, length))</code>	Creates a new figure with a specified length and width. <code>width</code> and <code>length</code> are both numbers in inches.
<code>plt.title('My Chart Title')</code>	A title for a chart.
<code>plt.xlabel('My X-Label')</code>	A label for the x-axis
<code>plt.ylabel('My Y-Label')</code>	A label for the y-axis
<code>plt.subplot(num_rows, num_cols, subplot_index)</code>	Creates a subplot for a grid with <code>num_rows</code> rows and <code>num_cols</code> columns. The new subplot is at a position defined by <code>subplot_index</code> . For instance, <code>plt.subplot(2, 3, 4)</code> would create a grid with 2 rows and 3 columns and would create a plot in the second row and the first column (4 "steps" if moving left to right and top to bottom).

Function	Description
<code>ax = plt.subplot()</code>	Creates an axes object (<code>ax</code>) that can be used for adjusting the position and labeling of x- and y-axis tick marks.
<code>plt.legend(['label1', 'label2'])</code>	Creates a legend using the labels given.
<code>plt.legend()</code>	Creates a legend using any <code>label</code> keywords that were given in <code>plt.plot</code> commands.
<code>ax.set_xticks([0, 1, 2, 3, 4])</code>	Creates tick marks at positions <code>[0, 1, 2, 3, 4]</code> on the x-axis. Requires that you created an axes object by using <code>ax = plt.subplot()</code> .
<code>ax.set_yticks([0, 1, 2, 3, 4])</code>	Creates tick marks at positions <code>[0, 1, 2, 3, 4]</code> on the y-axis. Requires that you created an axes object by using <code>ax = plt.subplot()</code> .
<code>ax.set_xticklabels(['label1', 'label2', 'label3'])</code>	Modifies the first three labels of the x-axis ticks marks to be <code>'label1', 'label2', 'label3'</code> Requires that you created an axes object by using <code>ax = plt.subplot()</code> . You'll probably want to start by specifying the positions of your x-ticks using <code>ax.set_xticks</code> .
<code>ax.set_yticklabels(['label1', 'label2', 'label3'])</code>	Modifies the first three labels of the y-axis ticks marks to be <code>'label1', 'label2', 'label3'</code> Requires that you created an axes object by using <code>ax = plt.subplot()</code> . You'll probably want to start by specifying the positions of your y-ticks using <code>ax.set_yticks</code> .

Function	Description
<code>plt.bar(x_values, heights)</code>	Creates a bar chart with bars at each value of <code>x_values</code> using the heights given in <code>heights</code> . If you're only creating one bar chart, <code>x_values</code> can be equal to <code>range(len(heights))</code> .
<pre># n: Number of the series # t: Total number of series to plot # d: Number of data points # w: width of each bart (default is 0.8) [t*x + w*n for x in range(d)]</pre>	Creates x-values for side-by-side bar charts.
<code>plt.bar(x_values, heights, bottoms=other_heights)</code>	Creates a second bar chart stacked on top of another bar chart whose heights are <code>other_heights</code> .
<code>plt.bar(x_values, heights, yerr=errors, capsize=capsize)</code>	Creates a bar chart with error bars where each bar's error height is given by <code>errors</code> and the width of each error bar is given by the integer <code>capsize</code> .
<code>plt.hist(dataset, bins=num_bins, range=(min, max))</code>	Creates a histogram of the dataset <code>dataset</code> . The default number of bins is 10, but if the <code>bins</code> keyword is given, then the number of bins is <code>num_bins</code> . The default <code>range</code> is all values of dataset, but if the keyword <code>range</code> is given, only values between <code>min</code> and <code>max</code> will be used.
<code>plt.hist(dataset, alpha=0.5)</code>	Creates a semi-transparent histogram
<code>plt.hist(dataset, histtype='step')</code>	Creates a histogram with a dark outline and no internal fill color.
<code>plt.hist(dataset, normed=True)</code>	Creates a normalized histogram where the sum of the area under the curve is 1. This is useful when graphing two histograms with different sized datasets.

Function	Description
<code>plt.pie(pie_chart_data)</code>	Creates a pie chart where each slice of the pie is given by the value of <code>pie_chart_data</code> divided by <code>sum(pie_chart_data)</code>
<code>plt.axis('equal')</code>	Makes a pie chart round and not oblong
<code>plt.pie(pie_chart_data, labels=['label1', 'label2', 'label3'])</code>	Creates labels for each slice of the pie chart
<code>plt.pie(pie_chart_data, autopct='%d%%')</code>	Labels for each slice of the pie chart with its percentage (i.e., '23%')
<code>plt.pie(pie_chart_data, autopct='%0.2f%%')</code>	Labels for each slice of the pie chart with its percentage (i.e., '23.05%')
<code>plt.fill_between(x_values, lower_y, upper_y)</code>	Creates a shaded area between the upper and lower bounds given. Usually used to show error around a line plot.