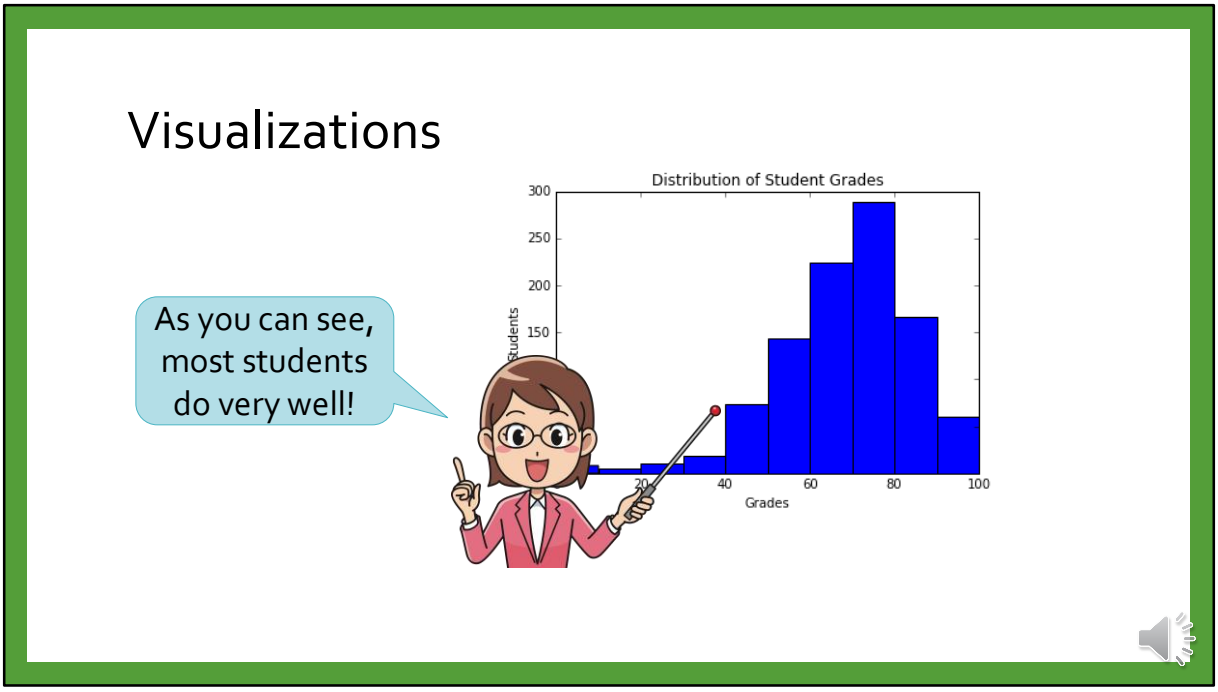
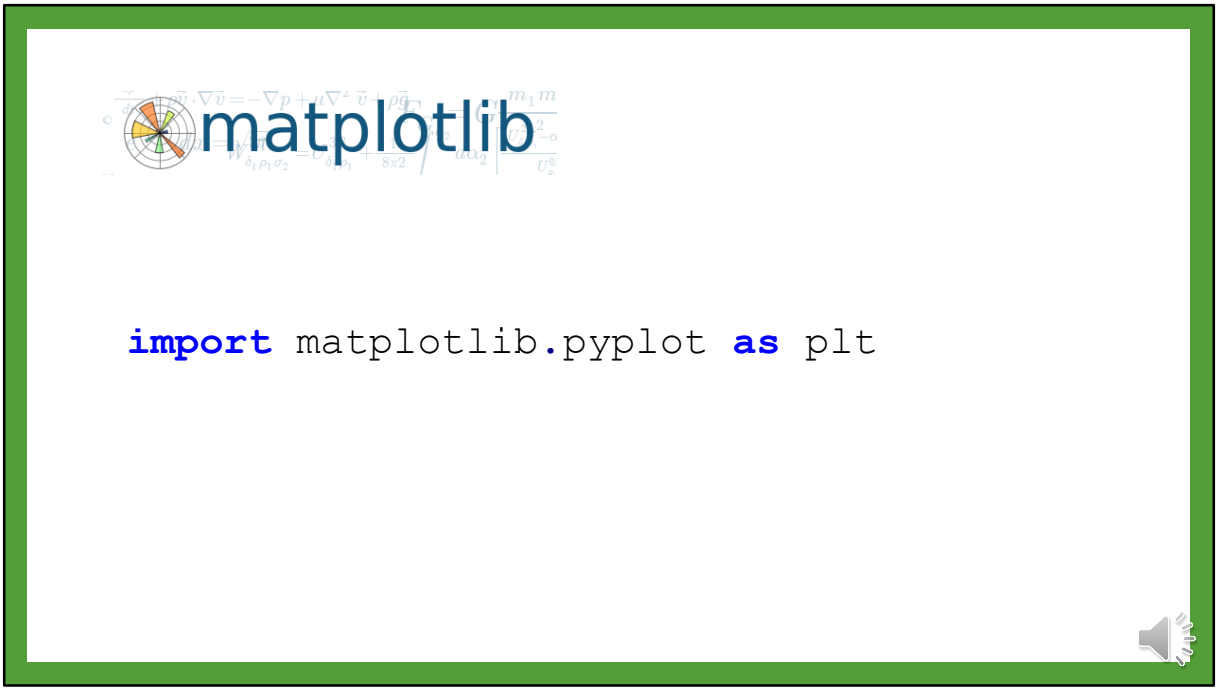
PLOTTING

An Introduction to Computer Science

Let's learn about Plotting



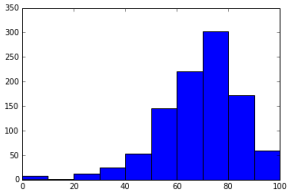
When you have a large amount of data, you can create visualizations to get a more intuitive understanding.
These visualizations, known as plots or graphs, take advantage of the fact that humans are good at processing pictures.
Making visualizations in Python is surprisingly easy.



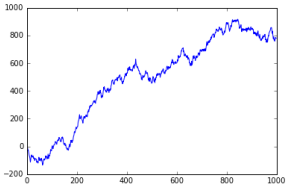
The most popular library for plotting in Python is named MatPlotLib.
As a very large package, we only need to import only a certain module.
For convenience, we rename that module to be "plt".

Types of Plots

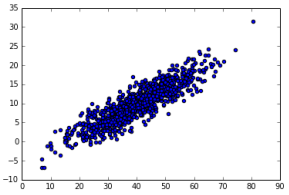
Histogram



Line Plot



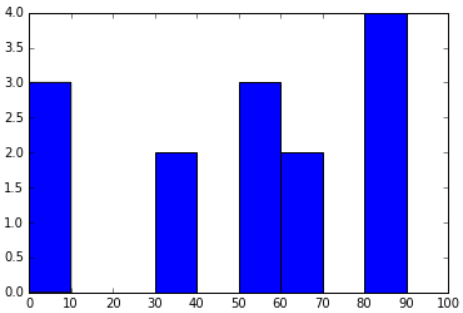
Scatter Plot



There are 3 kinds of plots that we'll learn about.
Each of the plots shown take in lists of numbers.
MatPlotLib actually has many other kinds of plots.
To create other kinds of plots, you can refer to the documentation.

Histograms

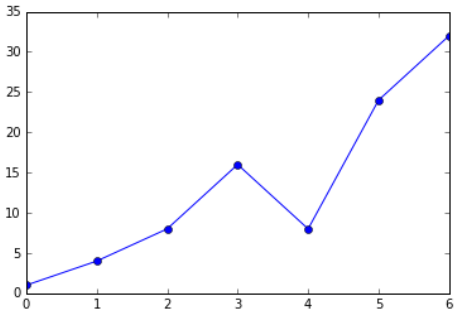
```
data = [0, 5, 7,
        35, 35,
        51, 53, 57,
        61, 64,
        82, 84, 84, 88]

plt.hist(data)
```



Although many people find Histograms confusing, they are actually a very simple kind of plot.
They should not be confused with Bar Charts, which are a more generalized kind of graph.
Instead, histograms take a list of numbers and group them into bins, or ranges of numbers.
For example, if we have the list of numbers shown, we could group them into bins of size 10.
Each tick on the y-axis means another number in that bin.
Histograms allow us to see the distribution or spread of the data.

Line Plots

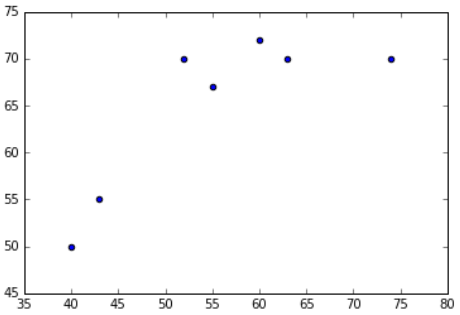
```
data = [1, 4, 8,
        16, 8, 24,
        32]

plt.plot(data)
```



Line plots are another simple kind of graph, and can be used to show a trend over time or some other factor.
Be careful when creating line plots, since people often use them when a Histogram would be more appropriate.

Scatter Plots

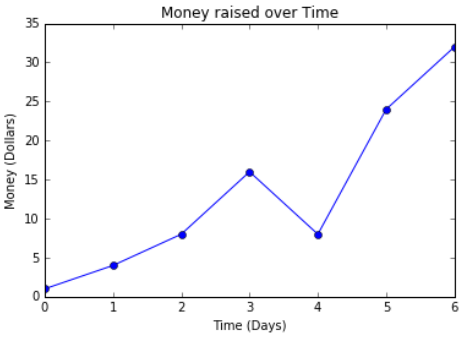
```
exam1 = [40, 43, 55, 60,
         52, 63, 74]

exam2 = [50, 55, 67, 72,
         70, 71, 70]

plt.scatter(exam1, exam2)
```



When you have two lists of data that you want to match up, a Scatter Plot is the graph to use.
For example, the graph shown here matches students' grade on a first exam with their grade on a second exam.
A scatter plot is used to find relationships between the two lists of numbers.

Labeling

```
plt.xlabel("Time (Days)")
plt.ylabel("Money (Dollars)")
plt.title("Money over Time")
```

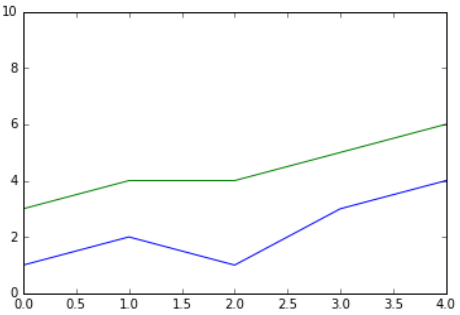


Professional data scientists always label their graphs.
MatPlotLib makes it easy to label the X, Y, and title of the graph.
The relevant functions are xlabel, ylabel, and title, each of which consumes a string.

Show

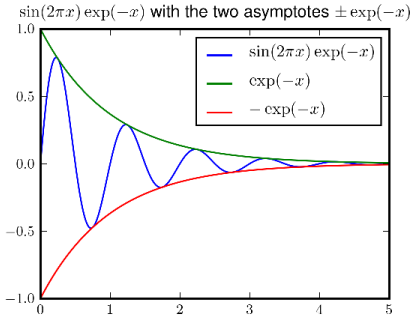
```
plt.plot([1, 2, 1, 3, 4])
plt.plot([3, 4, 4, 5, 6])
plt.show()
```

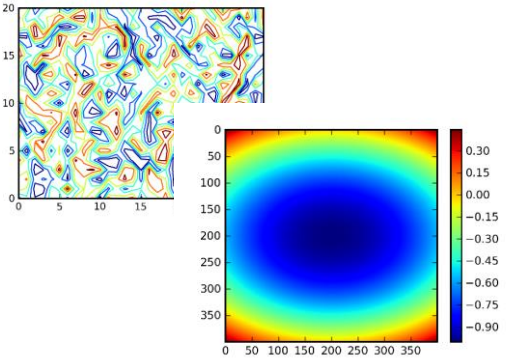
MUST show!



A trick about creating plots is that you need to call the "show" function afterwards.
If you do not call the show function, no graph will appear.
A useful feature of this function is that if you create multiple plots before showing, these graphs will be combined on the same canvas.

Advanced Features





Tosi, Sandro. *Matplotlib for Python developers*. Packt Publishing Ltd, 2009.

MatPlotLib is an extremely sophisticated library with many, many advanced features. It is very easy to make legends, adjust colors, more complicated graphs, and much more. Refer to the MatPlotLib documentation and look up examples of how to do more with MatPlotLib.