NORTHERN ILLINOIS UNIVERSITY

PHYSICS DEPARTMENT

Physics 374 – Junior Physics Lab Spring 2023

Python Tutorial #3

**Plotting Data**

In this tutorial, we will learn how to make basic plots. The *matplotlib.org* webpage: <https://matplotlib.org/3.3.3/tutorials/introductory/usage.html#sphx-glr-tutorials-introductory-usage-py> has a nice tutorial on making plots. If you go to <https://matplotlib.org/3.3.3/tutorials/index.html> you can see more tutorials.

To use *matplotlib*, we must install its module using *pip* (see Python Tutorial #1)

py -m pip install matplotlib

Matplotlib is a Python module containing a collection of plotting routines commonly used by scientists and engineers. The routines are very similar to those used in MATLAB. To use matplotlib, you will need to import it with the line

import matplotlib.pyplot as plt

where *plt* is the alias for the *matplotlib.pyplot* module.

Let us now plot the collection of grades in the *grades.txt* file of Tutorial #2. Create a Python project called *GradesPlot.py* and insert the following code

import numpy as np # the alias for "numpy" will be "np"

import matplotlib.pyplot as plt # the alias for "matplotlib.pyplot" will be "plt"

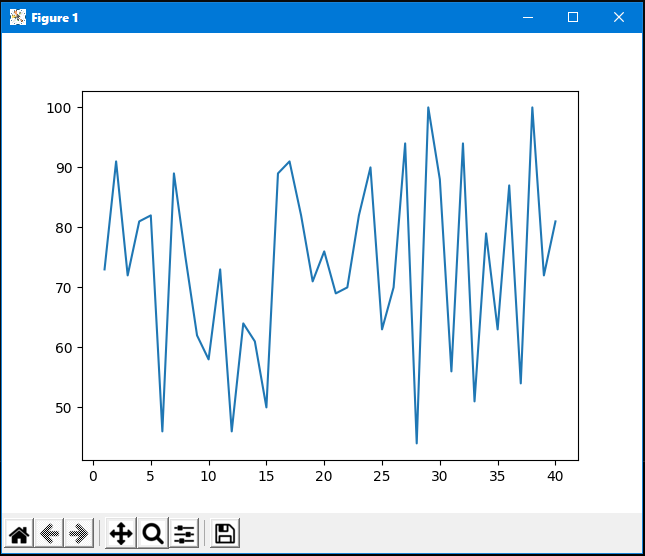
x, y = np.loadtxt('grades.txt', unpack=True) # unpack=True transposes columns

plt.plot(x,y) # Sets x = array of x data points (horizontal axis),

#y = array of y data points (vertical axis)

plt.show() # command to draw the plot on the screen

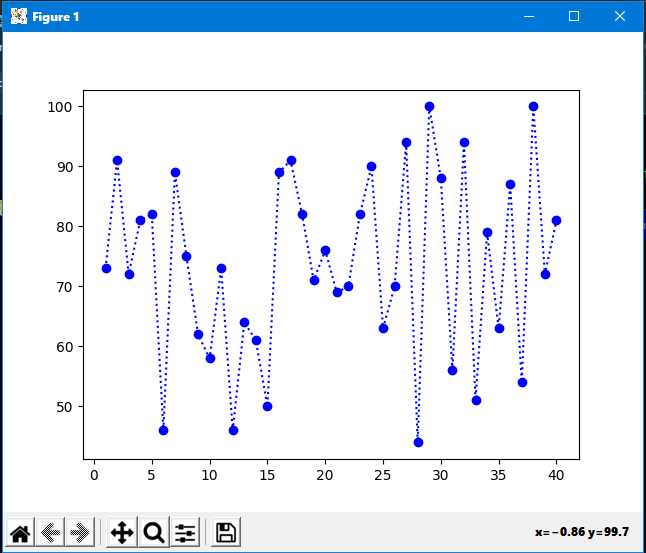
We need to import *numpy* to read the file *grades.txt* with the *loadtxt* routine. The routine *plt.plot* notifies *matplotlib* what the arrays are that have the of  data points (for the horizontal axis) and the  data points (for the vertical axis). To draw the plot on the screen, *plt.show* is used. Running the program should yield the plot below:



There are many ways to format the plot. See [https://matplotlib.org/3.3.3/api/\_as\_gen/matplotlib.pyplot.plot.html#matplotlib.pyplot.plot](https://matplotlib.org/3.3.3/api/_as_gen/matplotlib.pyplot.plot.html%23matplotlib.pyplot.plot) for the many ways to format a plot. For instance, to draw blue circles for the data markers with dotted lines between the points, change *plt.plot(x,y)* in your code to:

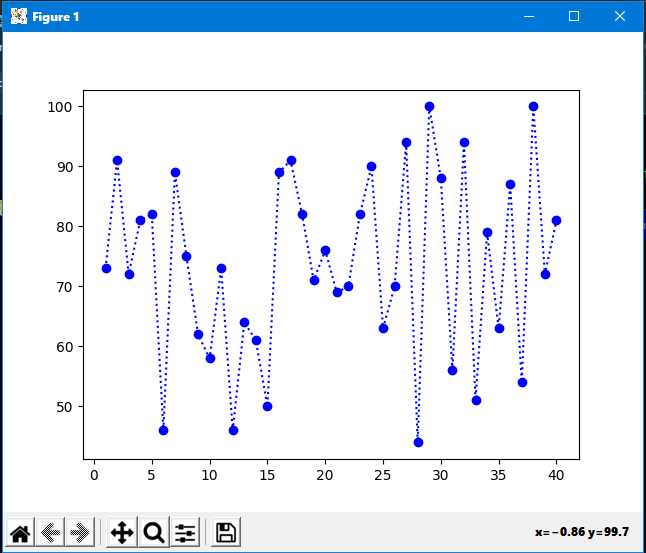
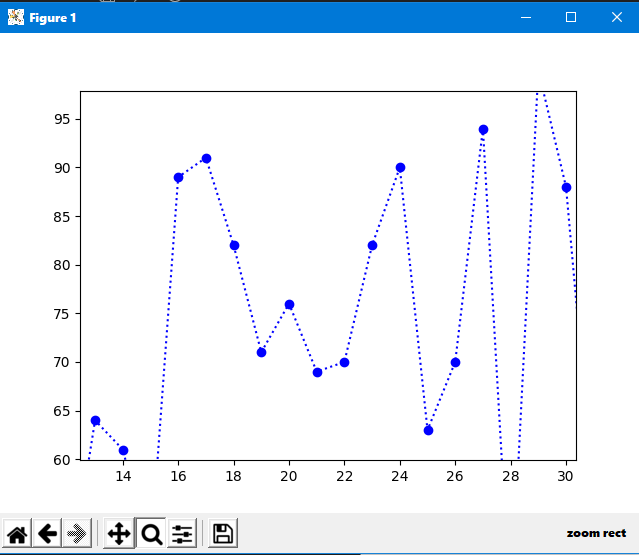
plt.plot(x,y,'b:o')

You should now get the following plot:

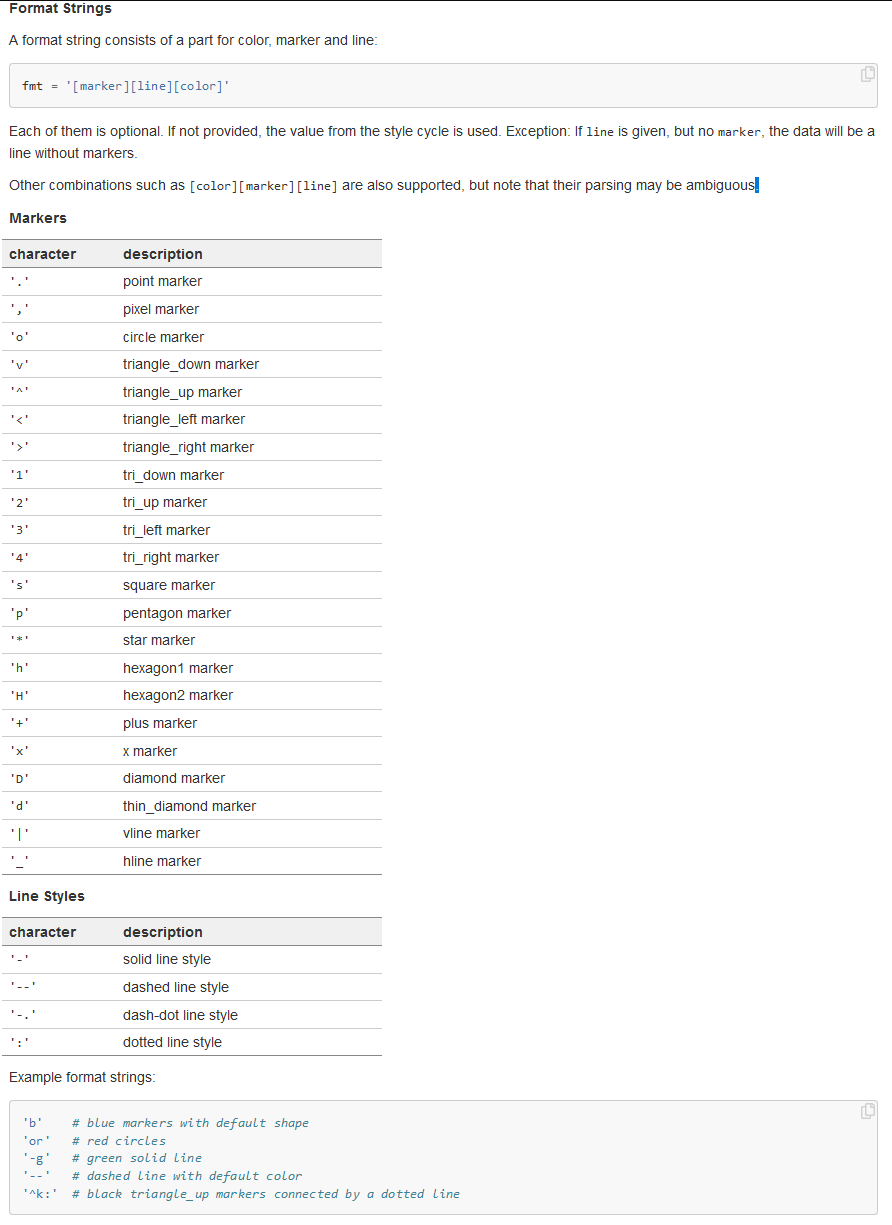


Zoom to rectangle

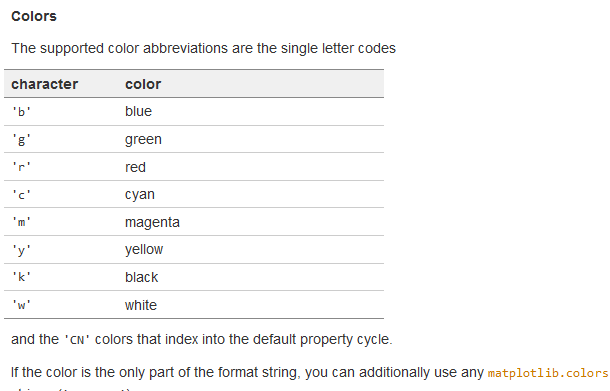
Notice that the matplotlib plots have a nice collection of controls at the bottom of the plot window. For instance the “Zoom to rectangle” control allows one to expand a region of the plot to examine fine details. The picture below shows what happens when it is used for a region near the center of the plot



Below is a list of format statements (like 'b:o' used above) from the matplotlib webpage:



**Finding the Mean of a Set of Grades**



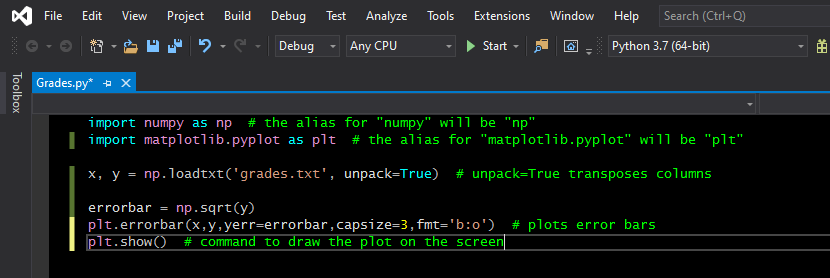




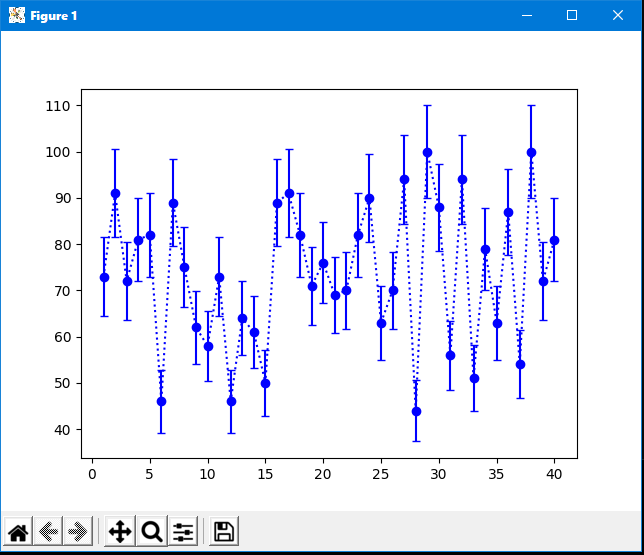
For data, we also need to be able to put in error bars. Insert the following code into your project in place of the code: plt.plot(x,y)

errorbar = np.sqrt(y)

plt.errorbar(x,y,yerr=errorbar,capsize=3,fmt='b:o') # plots error bars



I have arbitrarily made the error bar for a particular grade equal to its square root. An array called *errorbar* contains all the error bars for each point. The parameter *capsize* is used to put caps on the top and bottom of the error bar (ignoring *capsize* causes the program to use a default size of 0 which means no caps will be drawn).



**Homework**

Upload to Blackboard the source code (\*.py) of your Python program *GradesPlot.py*. You will see an assignment on Blackboard called **Python Tutorial #3**.