

Python Quick Reference Card

Basic Math Functions

+	addition	abs(x)	Absolute value of x
-	subtraction	sqrt(x)	Square root of x
*	multiplication	exp(x)	e to the x power
/	division	log(x)	natural log of x
a**b	a to the power of b	log10(x)	log base 10 or the usual log
XeY	X times 10 to the Y (2e4 = 2*(10**4))	sin(x)	sine(x) where x is in rad
pi	value of pi	cos(x)	cosine(x) where x is in rad

Startup

- 1) Go to the directory where you installed your Anaconda files (Windows: Start Menu ->Anaconda, Mac: Applications ->Anaconda).
- 2) Launch the Spyder application. Remember the console should be the tab opened on your right, with the>>>symbol waiting for a command. The editor is on the left, where you can create text files with a series of commands for the console to execute.
- 3) When finished with your work in console, you can always copy and paste over all of work into the editor window to save and print. Be sure to check your work by then running program with the green arrow at the top. Before doing so, remove all of the >>>symbols and comment out all of the outputs with the # symbol, so that Python knows not to execute these commands.

Array Commands

<code>data = np.loadtxt(r"XXXXtestdata.in")</code>	Reads all data from the file XXXXtestdata.in into an array called data where XXXX is the path to the filename. If the tutorial is followed directly, this should be displayed at the top of your editor window. Also, don't forget the quotes.
<code>x = data[:,1]</code>	Puts all rows, column 1 of data into an array x
<code>x = data[1:5,3:6]</code>	Puts rows 1-4 and columns 3-5 into a an array x
<code>np.array([1,2,3,4])</code>	Creates an array [1,2,3,4], recall how this is treated differently than a list
<code>np.arange(10,0,-1)</code>	Example of how to use counting array (would yield y = [10,9,8,7,6,5,4,3,2,1])
<code>np.zeros([3,4])</code>	Creates an array full of zeros with 3 rows and 4 columns
<code>np.random.rand(3,4)</code>	Creates an array full of randomly selected numbers from 0 to 1 with 3 rows and 4 columns
<code>np.mean(x)</code>	Returns the mean of array x. Other statistical functions available include median, max, min, and sum
<code>np.shape(y)</code>	Dimensions of array in parentheses
<code>len(x)</code>	Counts and returns number of elements in x. Argument can be list, array, or string.
<code>np.where(condition)</code>	Uses true/false logic to identify a subset of the array x that meets a condition Examples include: < Less than: <code>np.where(x< 3)</code> returns all indices of the array x less than 3 > Greater Than == Equal to != Not Equal & And: allows you to combine more than one condition with the following syntax <code>np.where((x< 3) & (x> 1))</code> . The argument must pass all conditions to be true. Note the extra parentheses. Or: Similar to & except the argument only needs to pass a single condition to be true. Note this is the vertical bar symbol, usually located above the enter key.

Plotting Commands

<code>plt.clf()</code>	clears the current graphic window
<code>plt.figure(#)</code>	opens a new graphic window or switches to graphic window #
<code>plt.plot(x,y,'b.',markersize=10)</code>	creates a plot of x versus y with blue dots of size 10
<code>plt.xlim(x0,x1)</code>	redefines the x axis to zoom in on plot
<code>plt.hist(data, bins=10, normed=false,range=(x0,x1))</code>	creates a histogram of "data" with 10 bins, normalization disabled, and the x axes limits set from x0 to x1
<code>plt.title('My Plot Title')</code>	creates plot title on the current figure
<code>plt.xlabel('X-axis Label')</code>	creates a label for the x axes of the current figure