

## Getting Started in Jupyter Notebook – Basic Tutorial

Video: [https://www.youtube.com/watch?v=dEWsl4OUJ\\_c&t=1199s](https://www.youtube.com/watch?v=dEWsl4OUJ_c&t=1199s)

### **Outline of topics covered:**

- \* accessing Jupyter (three different options are outlined on our webpage), starting a new Python 3 notebook (and giving your notebook a title!)
  - Google Jupyter Notebook images
  
- \* some basic ideas/terms:
  - cells, running cells (can learn shortcuts or use buttons), navigating notebook (you can find out more using “help” feature)
  - markdown/code (what happens if you type plain text and then run while in ‘code’ mode)
  
- \* importing math packages, assigning shortcuts
  - you will begin each notebook by running this code (either type by hand or just copy and paste from a previous notebook... if you loose your connection, or close/open a file, you will need to run this code again so that if you end up using/calling one of these packages, Jupyter recognizes it and can use it)
  
- \* some basic techniques we’ll be using in Math 1LS3
  - commenting out lines using “#”
  - defining arrays for our variables by:
    - manually entering values for x, y
    - appending values to (initially empty) arrays using a loop
    - using “np.linspace(a,b,n)”
  - coding common functions ( $y=x^2$ ,  $y=2x$ ,  $y=\cos x$ ,  $y=\arctan x$ ,  $y=e^x$ , etc.)
  - plots/scatter plots, plotting multiple graphs in the same coordinate system
  - calling/printing particular values from an array (using indices), listing ordered pairs (using a loop)

### *Exercises from video (some I’ve modified here):*

1. manually enter values in x- and y-arrays, scatterplot/plot (both, then commenting one out), changing esthetics (title, axes labels, grid, ‘show’, colour (if time))
2. create empty arrays, create a loop to generate x-and y-values, and then append to arrays (using the function  $y=x^2$ ... compare code for  $y=2x$ ), print x, print y, then print a particular ordered pair (for example,  $(x[0], y[0])$  or  $(x[8], y[8])$ ), then print all ordered pairs using a loop
3. define array of x-values using “np.linspace(a,b,n),” print x, define  $y=x^2$ , print y, plot (try this on your own by copying/pasting some code from above)
4. copy/paste code from exercise 3, change functions ( $\sin x$ ,  $e^x$ , etc.), plot two functions in one coordinate system, add a legend (label = “type label here”)

\* trouble-shooting

- downloading, unzipping, uploading a **.ipynb** file  
(make sure to 'unzip' it! watch file extension! know where your downloaded files are saved!)
- make sure you follow the rules!!
  - commas, indents, spacing, etc. all have meaning!
  - $3x$  must be written as  $3*x$ ,  $x^3$  must be written as  $x**3$ , etc.

\* some other general tips that I've found useful

- copy and paste code into a new cell and edit the new cell... that way, if something goes terribly wrong, you can just delete the cell, copy and paste the code you know works into a fresh cell and try again
- run the code frequently after making small changes (rather than make all changes at once and then try to run at the end)... it's easier to identify where you've gone wrong if you've only changed 1-2 parts of the code
- start small! Begin with an example that you already know a lot about theoretically (for example,  $y = x^2$ ) to test that your code is doing what you want it to... then, build it up piece by piece, running code frequently to check, until you can extend the code to explore new complex situations
- make your code easy to read by adding comments (preceded by #) everywhere.... anytime I use some new (to me) code, I write a comment about what it does so that I don't have to keep looking it up
- I often use the "print" function to print my output values so I can figure out if my code is doing what I want it to do