**How to set up a Python environment and Jupyter Lab**

* You DO NOT need a cloud-based environment to complete the EY Data Challenges. These challenges have been designed to work on local computers (PC or Mac) with basic configurations (e.g., 4 cores, 32GB memory).
* In order to develop and run Python code on your local computer, you will need a Python virtual environment and Jupyter Lab for managing notebooks (code, text, output).
* You can use “**Anaconda**” to setup a Python environment with Jupyter Lab and lots of common Python libraries. It is best for “beginners” but loads lots of extra content and has less flexibility.
* For a “manual” Python virtual environment setup, check out this video for PC or Mac: <https://www.youtube.com/watch?v=9tPS-7TWjq0>
* Some popular terminal commands are: “python3 -V” to view the installed Python version, “pip3 list” to view installed Python libraries, and “pip3 install ###” to install new Python libraries named “###”.
* For installing JupyterLab on your Mac computer, check out this video:  
  <https://www.youtube.com/watch?v=578B63wZ7rI>
* For installing Jupyter Lab on a PC or Unix system, just Google “Jupyter Lab installation” for many more videos and details.

**How to use Jupyter Python Notebooks**

Jupyter Notebooks allow you to create and share documents that contain live Python code, equations, visualizations and narrative text.

* A screenshot of a computer

  Description automatically generatedFor a better Jupyter Lab setup, you might try these options. Select “View > File Browser” to see of the files in your local directory. You can also navigate to other folders. Select “View > Appearance > Simple” to change the interface to “with tabs” or “without tabs”. This will allow you to see which notebooks are open for quick editing.
* You will find two types of "cells" in notebooks. A cell used for text or comment is called "**Markdown**" format. A cell used for Python code is called "**Code**" format. When you want to add a new cell in the notebook (use + in menu), be sure you use the correct cell type (dropdown selection on top-right).
* There are several ways to "run" the notebook code. To run the entire notebook (starting from the top), you can select "**Kernel > Restart & Run All**". Once the code has been executed (top to bottom) you can change individual cell content and rerun portions of the code by going to any cell and hitting "**Shift - Enter**". You will notice this approach will renumber the code blocks starting with the last number that was executed. So, it may be confusing. To reset the numbering (1 to xxx), just run the entire script, as suggested above. If you want to stop the execution at any time, just select "**Kernel > Interrupt**".
* When the code is "running" you will notice the cell blocks will look like "In [\*]". The "star" means the code is executing. When the cell is done executing the "star" will turn into a sequential number, starting with the last executed block number. You will see that some blocks run very fast, and others take some time. If you run the entire stack, you can scroll to the top and see the code execute along the way as it creates output as it moves along the blocks.
* The "#" symbol can be used to make any code line a comment and it is not executed. You can also add # comment tags at the end of code lines.
* If your code gets "hung up" and does not appear to be executing, you can go back to the main Jupyter Notebook page and select the "Running" tab to view which notebooks are being "executed". In some cases, these notebooks are actually running, but in other cases they are just "open" and sitting in the memory and ready for editing or running. You can "**Shutdown**" any notebook from this screen.