Jupyter Notebook Tutorial

- The Jupyter Notebook is an incredibly powerful tool for interactively developing and presenting data science projects.
- A notebook (it’s a project with CodeBlocs) integrates code and its output into a single document that combines visualisations, narrative text, mathematical equations, and other rich media.

Creating Your First Notebook:

- you can start the dashboard on any system via the command prompt (or terminal on Unix systems) by entering the command jupyter notebook; in this case, the current working directory will be the start-up directory.

- the URL for the dashboard is something like http://localhost:8888/tree. Localhost is not a website, but indicates that the content is being served from your *local* machine: your own computer.
  Jupyter's Notebooks and dashboard are web apps, and Jupyter starts up a local Python server to serve these apps to your web browser, making it essentially platform independent and opening the door to easier sharing on the web.
- To create a Notebook : Browse to the folder in which you would like to create your first notebook, click the "New" drop-down button in the top-right and select "Python 3".
- You will see the new file Untitled.ipynb: It’s a text file that describes the contents of your notebook in a format called JSON.

Cells and Kernels:

- A kernel is a "computational engine" that executes the code contained in a notebook document.
- A cell is a container for text to be displayed in the notebook or code to be executed by the notebook's kernel.

1. **Cells :**

In the screenshot of a new notebook in the section above, that box with the green outline is an empty cell. There are two main cell types that we will cover:

- A code cell contains code to be executed in the kernel and displays its output below.
- A Markdown cell contains text formatted using Markdown and displays its output in-place when it is run.

→ To run cell click the run button in the toolbar above or press Ctrl + Enter.

Keyboard shortcuts

- Toggle between edit and command mode with Esc and Enter, respectively.
- Once in command mode:
    - Scroll up and down your cells with your Up and Down keys.
    - Press A or B to insert a new cell above or below the active cell.
    - M will transform the active cell to a Markdown cell.
    - Y will set the active cell to a code cell.
    - D + D (D twice) will delete the active cell.
    - Z will undo cell deletion.
    - Hold Shift and press Up or Down to select multiple cells at once.
        - With multiple cells selected, Shift + M will merge your selection.
- Ctrl + Shift + -, in edit mode, will split the active cell at the cursor.
- You can also click and Shift + Click in the margin to the left of your cells to select them.

Markdown

Markdown is a lightweight, easy to learn markup language for formatting plain text. Its syntax has a one-to-one correspondence with HTML tags, so some prior knowledge here would be helpful but is definitely not a prerequisite. Remember that this article was written in a Jupyter notebook, so all of the narrative text and images you have seen so far was achieved in Markdown. Let's cover the basics with a quick example.

2. Kernels

Behind every notebook runs a kernel. When you run a code cell, that code is executed within the kernel and any output is returned back to the cell to be displayed. The kernel's state persists over time and between cells — it pertains to the document as a whole and not individual cells.

For example, if you import libraries or declare variables in one cell, they will be available in another. In this way, you can think of a **notebook document as being somewhat comparable to a script file,** except that it is multimedia.

There are seve useful options from the Kernel menu:

- Restart: restarts the kernel, thus clearing all the variables etc that were defined.
- Restart & Clear Output: same as above but will also wipe the output displayed below your code cells.

- Restart & Run All: same as above but will also run all your cells in order from first to last.

Naming your notebooks:

- you cannot name or rename your notebooks from the notebook app itself, but must use either the dashboard or your file browser to rename the .ipynb file.
- You cannot rename a notebook while it is running, so you've first got to shut it down. The easiest way to do this is to select "File > Close and Halt" from the notebook menu.

Setup :

It's common to start off with a code cell specifically for imports and setup, so that if you choose to add or change anything, you can simply edit and re-run the cell without causing any side-effects.

Save and Checkpoint:

- Pressing Ctrl + S will save your notebook by calling the "Save and Checkpoint" command.
- Every time you create a new notebook, a checkpoint file is created as well as your notebook file; it will be located within a hidden subdirectory of your save location called .ipynb_checkpoints and is also a .ipynb file. By default, Jupyter will autosave your notebook every 120 seconds to this checkpoint file without altering your primary notebook file. When you "Save and Checkpoint," both the notebook and checkpoint files are updated. Hence, the checkpoint enables you to recover your unsaved work in the event of an unexpected issue. You can revert to the checkpoint from the menu via "File > Revert to Checkpoint."