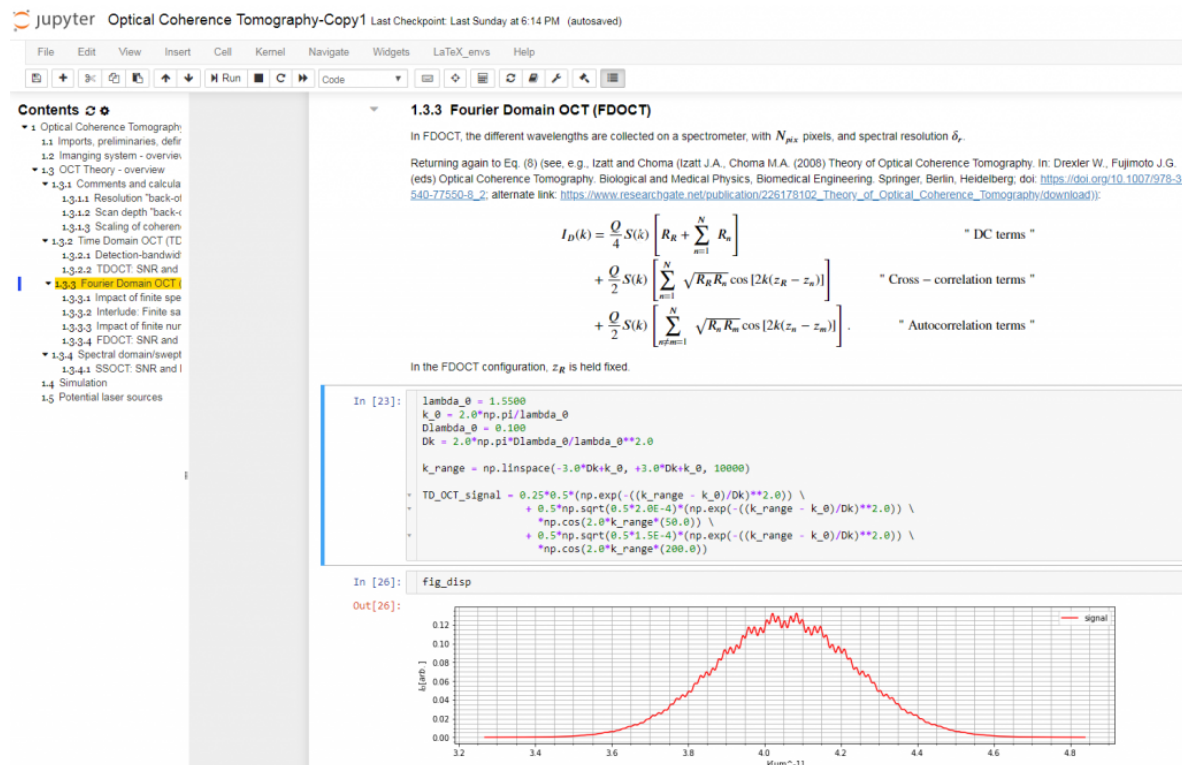


## Running Python in Jupyter Notebooks

Python is a free, openly available programming language, it has support for pretty much every type of ML including deep learning, and it is designed to be “human legible” (sort of...).

The easiest way to get started with running python is through a separate tool called jupyter. Jupyter allows you to run and save code in “notebooks” such as the example screenshot below:

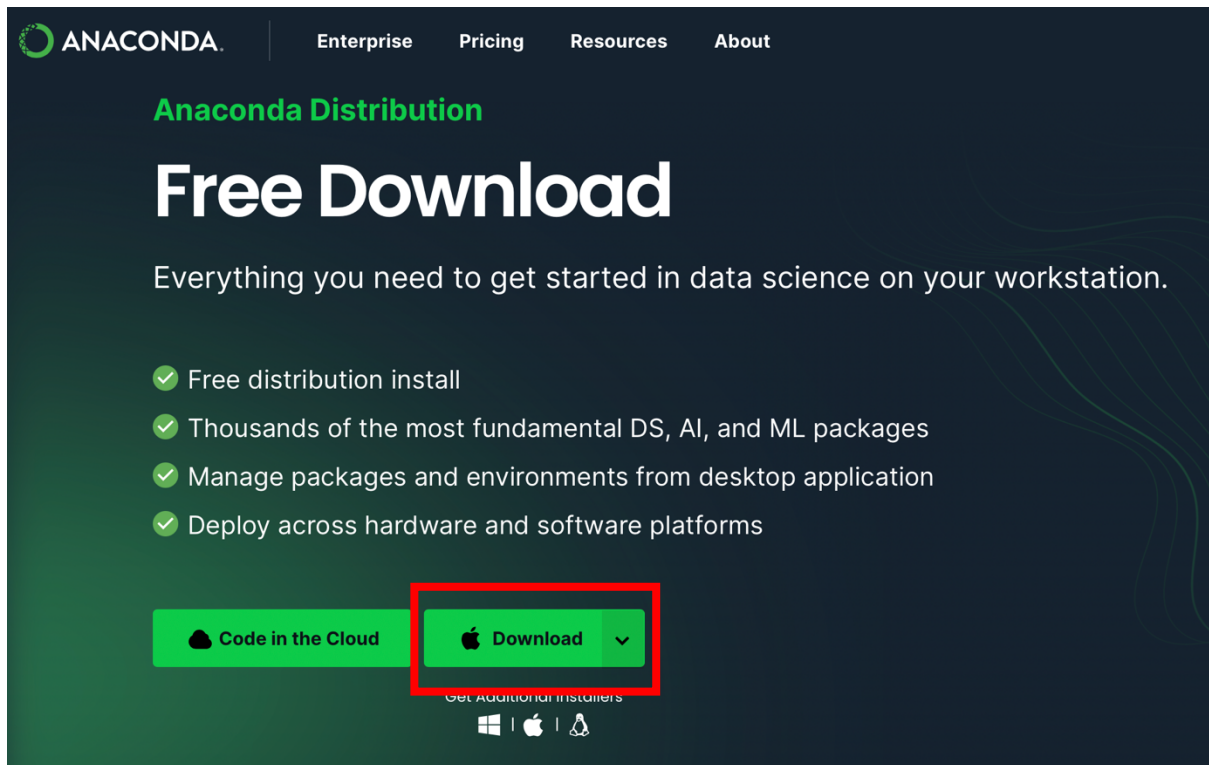


The advantage of this is that notebooks can be saved (including all the figures and code outputs), and shared with others to share both the outputs of the code, but also to enable reproducibility of workflows. **Many journals now expect code to be shared in formats like jupyter notebooks for reproducibility.** See [https://github.com/shorthouse-lab/binding\\_ddg](https://github.com/shorthouse-lab/binding_ddg) for an example of a github repository containing jupyter-notebooks linked to a publication (Jamshidi Parvar et al. 2023). I have prepared a few notebooks to get people started using python and running machine learning models.

Below are some instructions for installing and using jupyter notebooks.

## Installing Anaconda

Anaconda (<https://anaconda.org>) is a data science software tool, it contains prebuilt and installed libraries for doing essentially all ML tasks, including deep learning. It also comes with jupyter notebook pre installed, and can be adapted to run Rstudio, and almost any other data science/coding environment. Anaconda is free to use, and should be installed on shorthouse lab computers. For personal laptops and other devices, anaconda can be downloaded from the website:



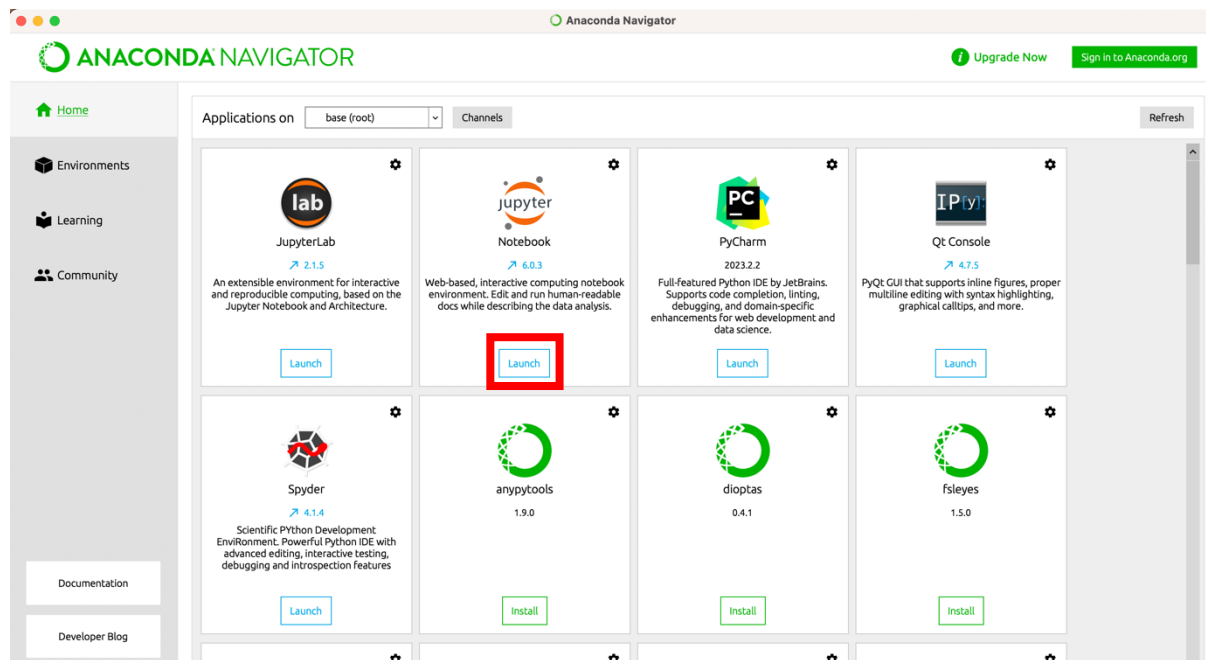
Once downloaded, the software is installed like most other packages (through a .exe on windows, and through a .dmg on mac).

You can run anaconda same as any software through the start menu/app store:



This will open up the anaconda navigator:

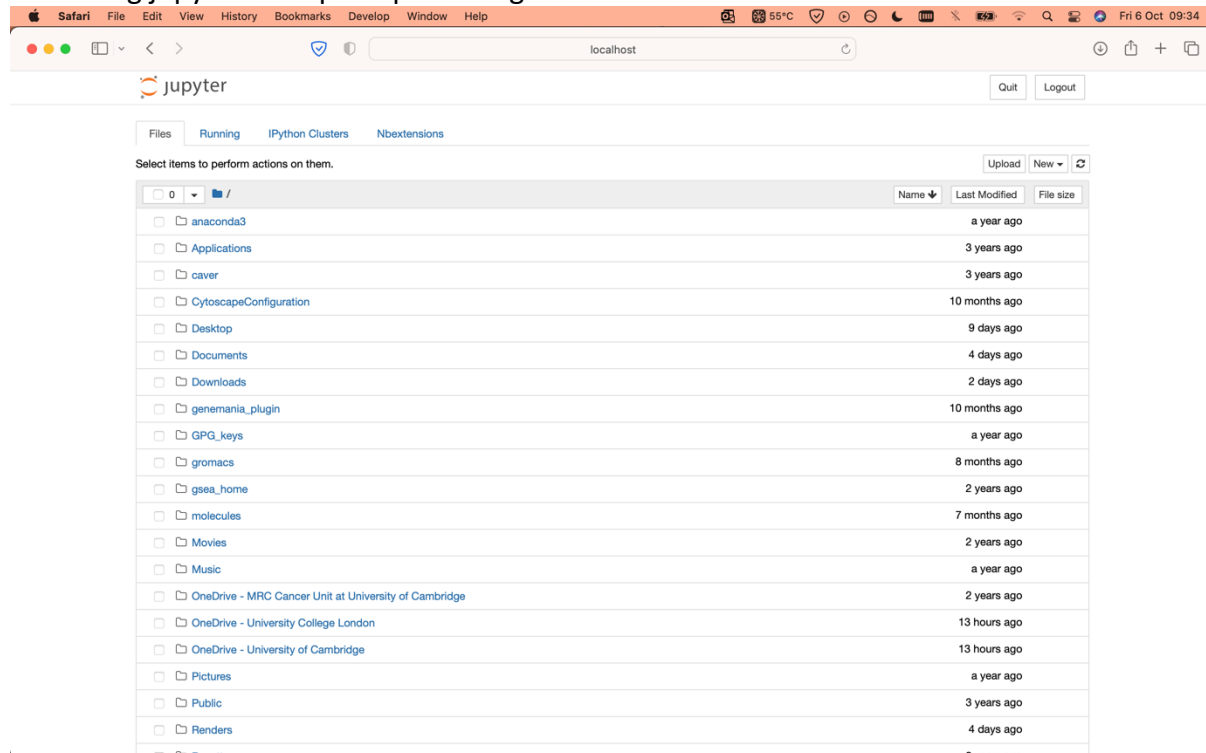
## D.Shorthouse – Getting started with python in Jupyter



Here you can run any of the installed software. Jupyter should be installed by default, and can be started by clicking the “launch” button below it.

Note: If you can use the command line on linux/mac, you can also start jupyter notebooks in any location by running “jupyter-notebook” as a command when in that directory.

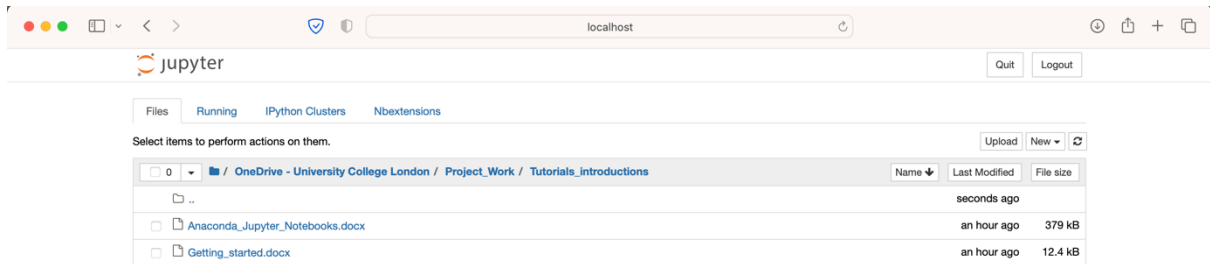
Launching jupyter will open up a navigation window in a web browser:



Mine has opened in safari, but yours will open in whatever web browser is your default (likely chrome). This is the window in which we will run the code and make our notebooks,

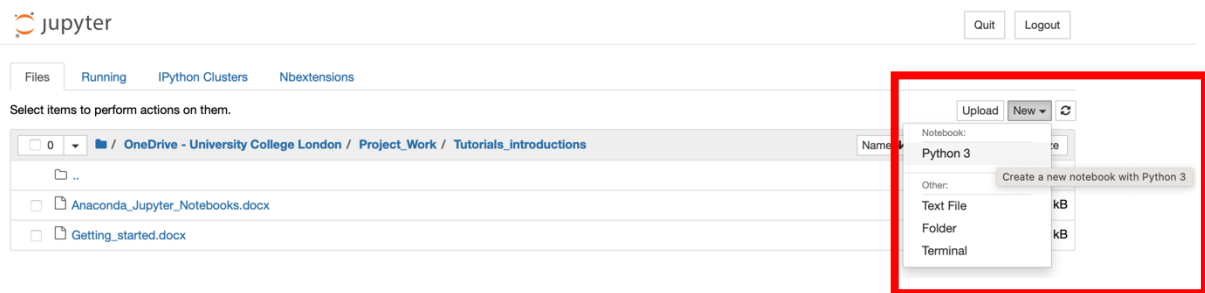
## D.Shorthouse – Getting started with python in Jupyter

but note that **this doesn't require you to be connected to the internet**. The web browser is just the platform jupyter is using. You can now navigate through your folders, see your files, and open up existing jupyter notebooks. In the below screenshot I have moved into a folder, and can see some documents.

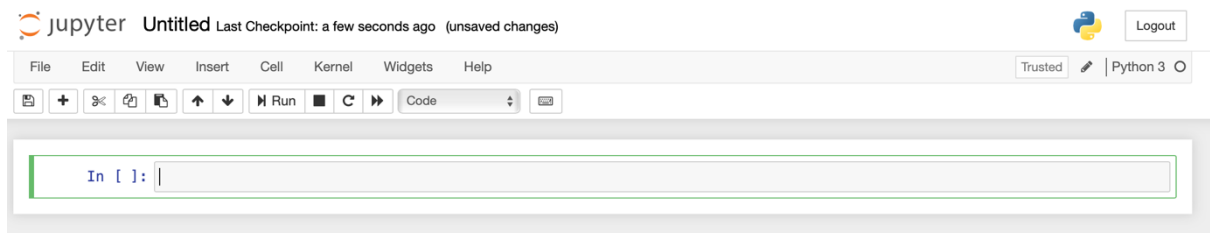


You can open up files and folders – files with the extension “.ipynb” are jupyter notebook files, and can be opened here to display contents and run the code they contain.

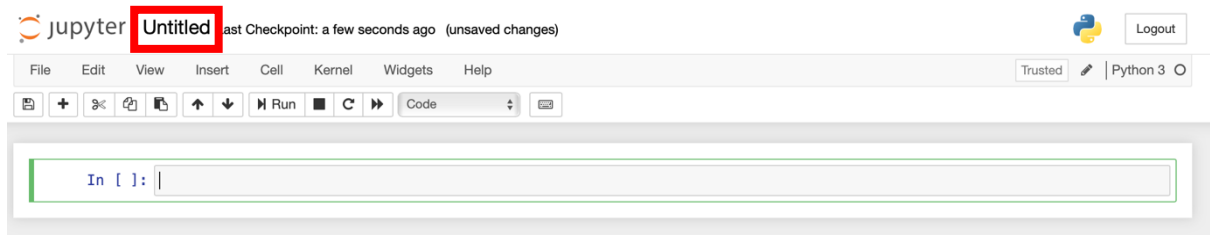
You can start a new notebook in the folder you are currently in through the “new” menu in the interface:



This opens up a blank notebook:



The filename can be changed by clicking here:

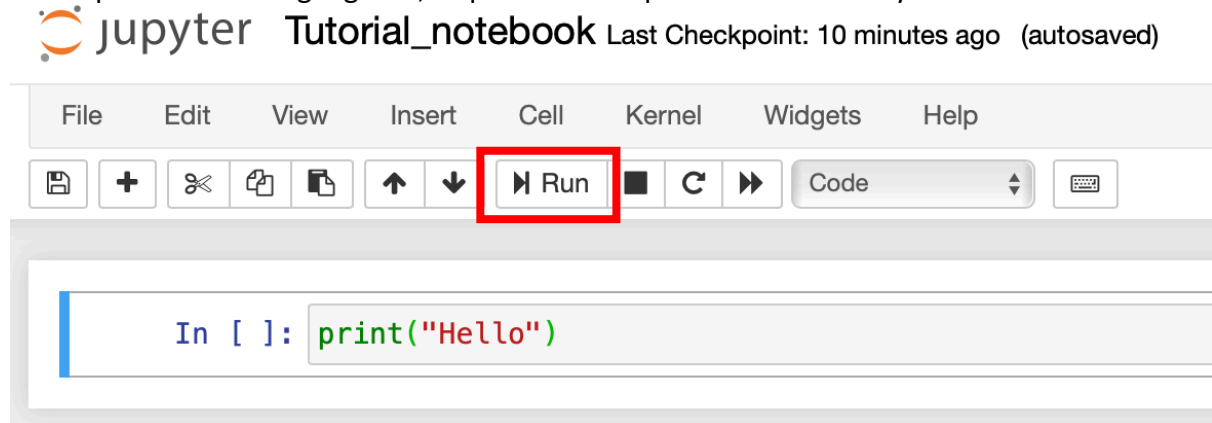


The file, edit, view, insert etc menus contain standard options like “save as” etc.

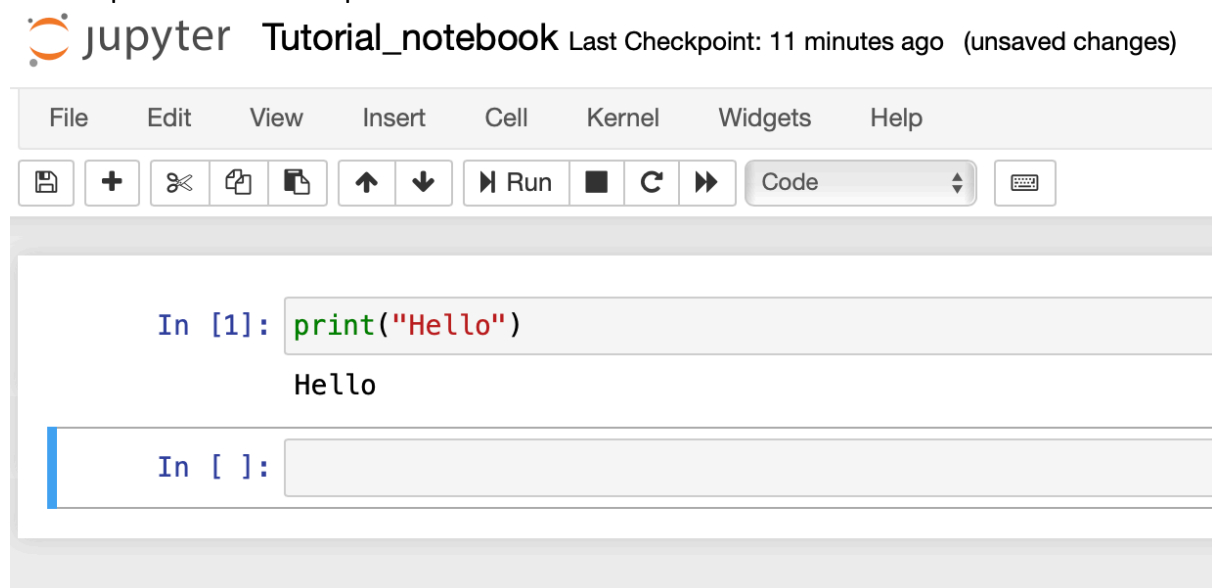
### Getting started with running code in jupyter

Now that jupyter is loaded we can start to run code and work with python. Another tutorial for getting started with python is available at: <https://www.kaggle.com/learn>

To start running code in jupyter, simply type code into the main box (called a “cell”), and either press “run” – highlighted, or press shift + space simultaneously:



The output to the code is printed below:



A new cell has also appeared to allow you to write the next piece of code.

Generally in python:

- Anything written between quotes is treated as a “string”, and is treated like text (numbers aren’t recognised as numbers for example).
- Numbers are treated as integers or floats depending on whether they have decimal places. These can be added, divided etc.
- You can assign any of these types of data to a “variable” with the equals sign. Python will remember what the last assignment is, and so you can call a variable by name.

## D.Shorthouse – Getting started with python in Jupyter

- Anything following a hash (#) is a comment – python doesn't read this, and it is used to help the reader understand the code.
- The "print" function will print out anything in the brackets, but it cannot mix numbers and strings of text, so numbers have to be converted to a string before they can be printed with text at the same time.

See an example of this below:

```
In [8]: # This is a comment – python does not read it
print("Here is some introductory maths:")

# We assign the numbers 7 and 15 to variable1 and variable2
variable1 = 7
variable2 = 15

print("\nVariable 1 times Variable 2 equals:")

# We print out variable1 * variable2
print(variable1 * variable2)
```

Here is some introductory maths:

Variable 1 times Variable 2 equals:  
105

This is the absolute basics of getting started with python. I would highly encourage you to look more at the tutorials on Kaggle: <https://www.kaggle.com/learn> and move on to the other documents.

The next logical document to start looking at is "Working with and plotting data in python".