1_getting_setup

September 28, 2023

1 Getting Started

1.1 Learning Outcomes

- Use the `print` function.
- Use Python as a simple calculator.
- Use three different Python data types: integer, float, and complex.
- Add comments to your Python codes.

1.2 Preamble

We'll start slowly to make sure everyone feels confident with the absolute basics. We'll start by showing you how to get started using Jupyter, a web browser application that is one of several ways of coding in Python.

You will then work through some simple calculations and work with data types and variables. You are free to move on to future chapters if you complete this quickly or already have an understanding of Python programming.

Let's start by making a flow chart to describe the following:

Something that includes a series of steps that you do so often that it is second nature (don't be rude/personal/gross! Shouldn't need saying but past experiences…), e.g. making a cup of tea, having a shower, getting the bus to campus…..

These flowcharts should be done by hand initially, but if you are keen and want to learn a new skill, you could try making them using online flowchart software such as `lucidchart.com` or `gliffy.com` or `drawio.com`. Please note that not all of these packages work on Edge/Internet Explorer (try Chrome instead). These flow charts do not have anything with Jupyter. The intention is to get you thinking about how programming works, not to get you to write code.

Each year we are asked, "Why are we doing flowcharts?". So here is the answer: To fully understand the operations of a program, flowcharts are indispensable tools. In simple terms, programs operate by stepping through a set of instructions/decisions (the code) and executing them one at a time (just as you step through a flowchart). Although the examples above are simple, programs can get very complicated (astronomical survey analysis, data reduction at CERN, cosmological simulations). You can have tens of thousands of lines of code with multiple paths for data, a structure too complicated to just keep in your head. Flowcharts give a schematic view of the different paths data can take through these programs, and the possible decisions based on that data. Flowcharts get you to think about **how a program functions** before you've even written it, and are **an important skill** you may need later in your career.

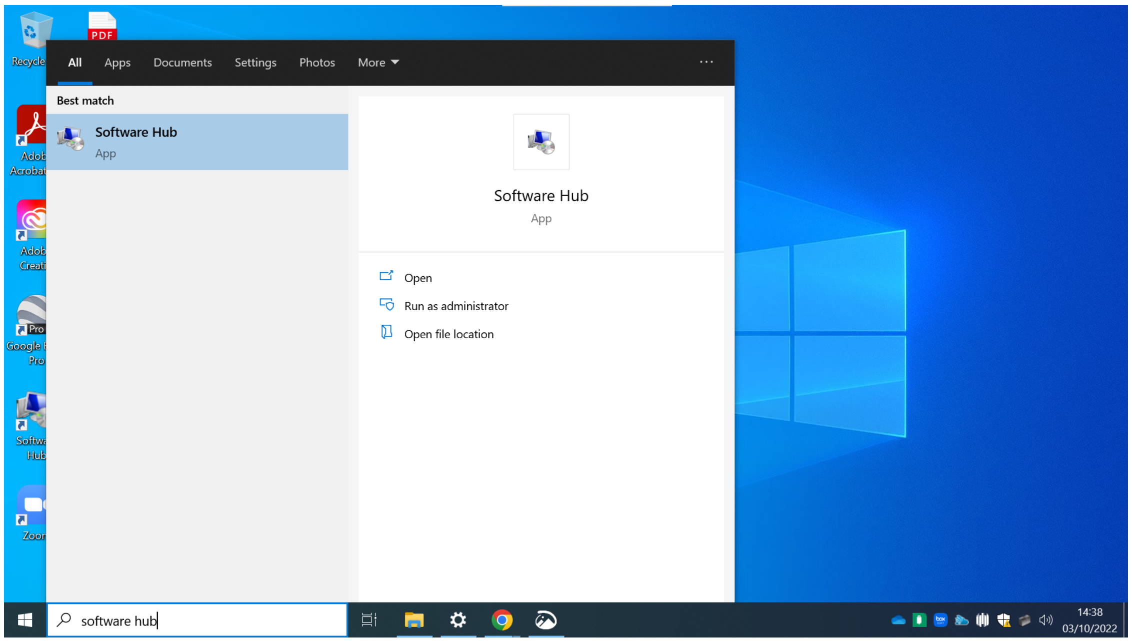
Simple example flow charts can be found in the appendix of one of this document's author's published papers (Roper et al. 2020), mentioned here just to drive home that they are useful!

1.3 Running `Python` on the University computers

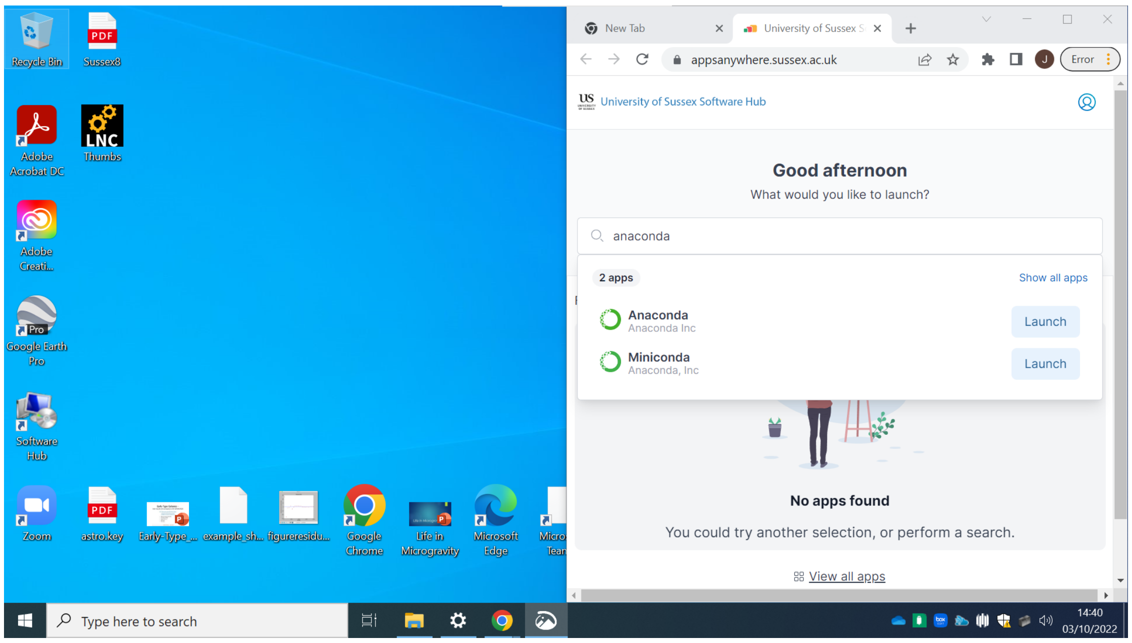
Throughout this term, we will be using Jupyter Notebooks. Jupyter is a web browser application that allows you to create and share programs, which in our case will be written in Python. If you choose to use your own computer you can disregard this section and move on to the next.

1.3.1 Running `Anaconda` for the first time

The first time you use Anaconda on a University PC, you need to launch Anaconda via the Software Hub. To do this, type 'software hub' into the search box at the bottom left of the screen, and run the *Software Hub* app.

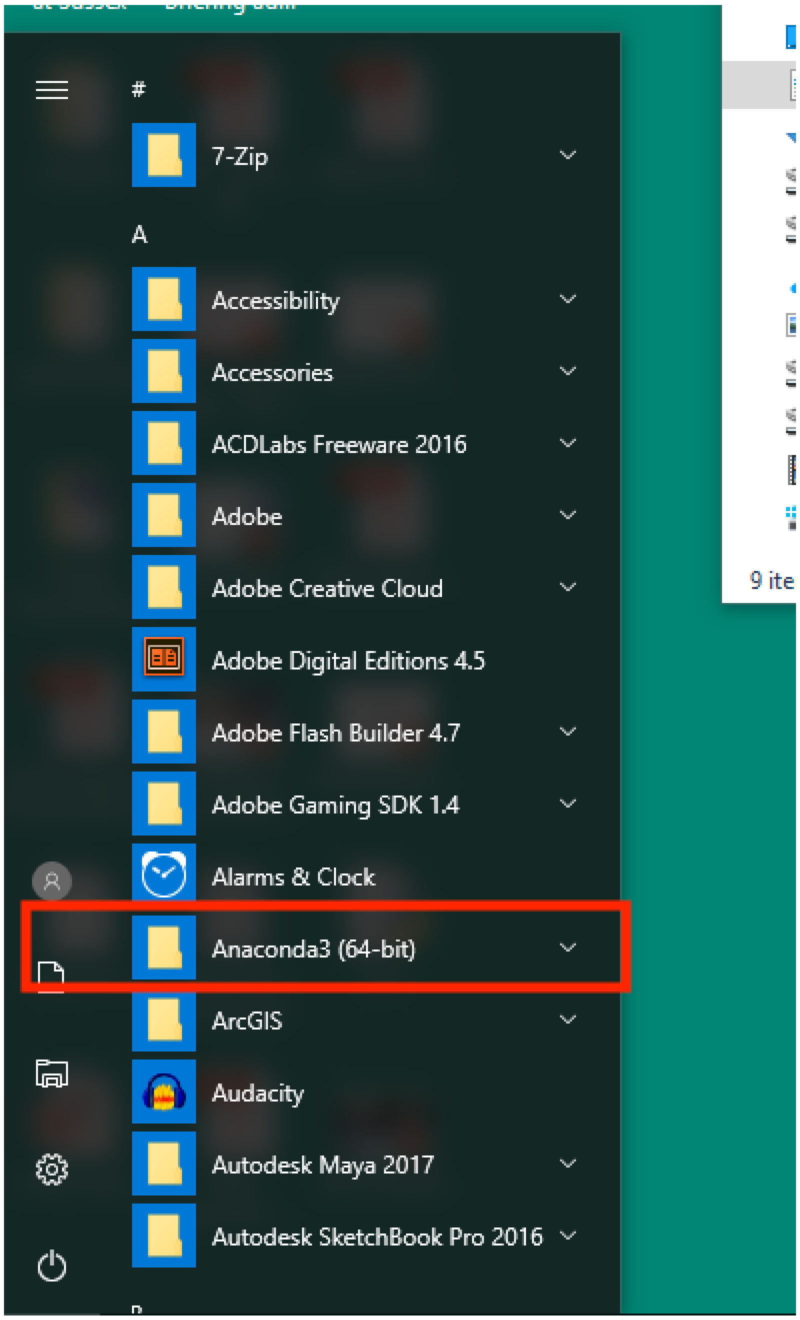


Type `anaconda` into the software hub search box and click Launch.



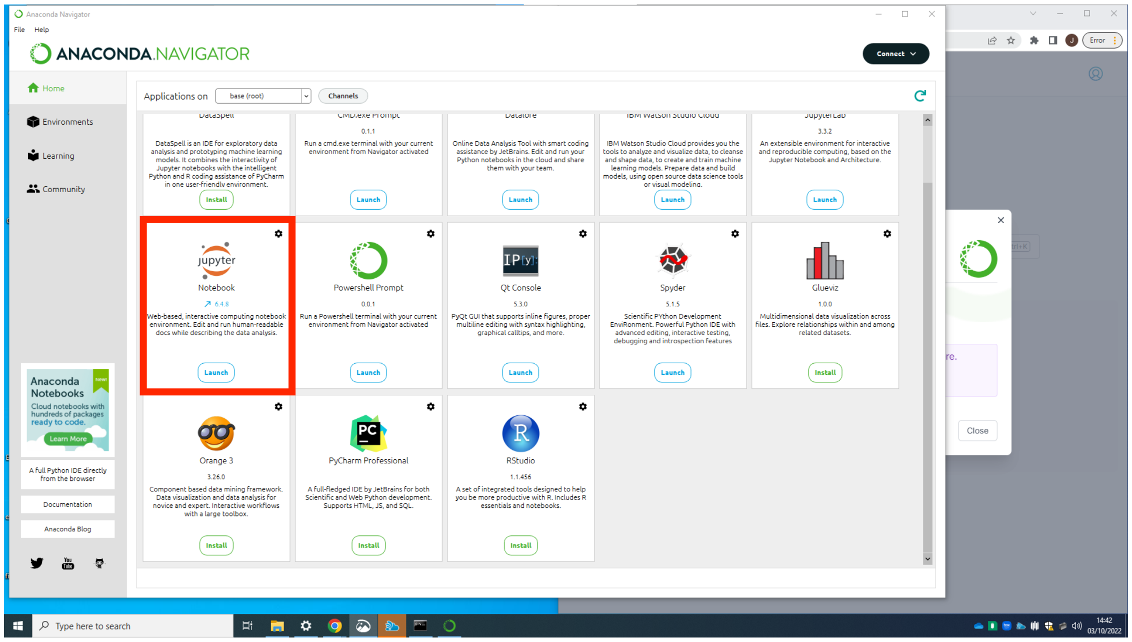
1.3.2 Running `Anaconda` in subsequent sessions

On subsequent use, you should only need to select Anaconda from the Windows `Start` menu.



1.3.3 Running Jupyter Notebook

After a few moments, the Anaconda Navigator will appear. From it, launch Jupyter Notebook as shown below.



DataSpell

DataSpell is an IDE for exploratory data analysis and prototyping machine learning models. It combines the interactivity of Jupyter notebooks with the intelligent Python and R coding assistance of PyCharm in one user-friendly environment.

CMD.exe Prompt
0.1.1
Run a cmd.exe terminal with your current environment from Navigator activated

Datalore

Online Data Analysis Tool with smart coding assistance by JetBrains. Edit and run your Python notebooks in the cloud and share them with your team.

IBM Watson Studio Cloud

IBM Watson Studio Cloud provides you the tools to analyze and visualize data, to cleanse and shape data, to create and train machine learning models. Prepare data and build models, using open source data science tools or visual modeling.

JupyterLab
3.3.2
An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.

Notebook
6.4.8
Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.

Powershell Prompt
0.0.1
Run a Powershell terminal with your current environment from Navigator activated

Qt Console
5.3.0
PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.

Spyder
5.1.5
Scientific PYthon Development EnviRonment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features

Glueviz
1.0.0
Multidimensional data visualization across files. Explore relationships within and among related datasets.

Orange 3
3.26.0
Component based data mining framework. Data visualization and data analysis for novice and expert. Interactive workflows with a large toolbox.

PyCharm Professional

A full-fledged IDE by JetBrains for both Scientific and Web Python development. Supports HTML, JS, and SQL.

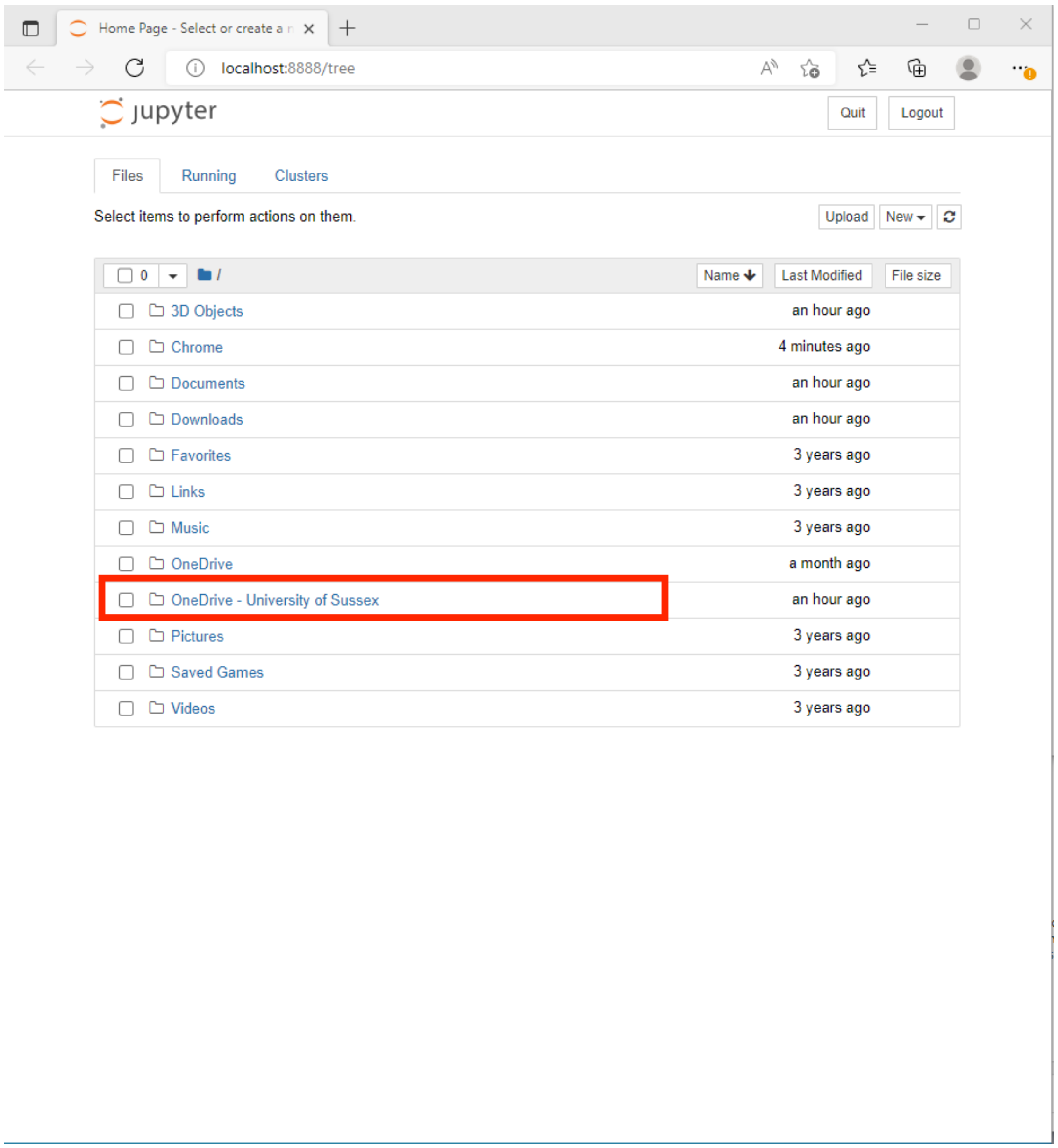
RStudio
1.1.456
A set of integrated tools designed to help you be more productive with R. Includes R essentials and notebooks.

Anaconda Notebooks
Cloud notebooks with hundreds of packages ready to code.
Learn More

A full Python IDE directly from the browser

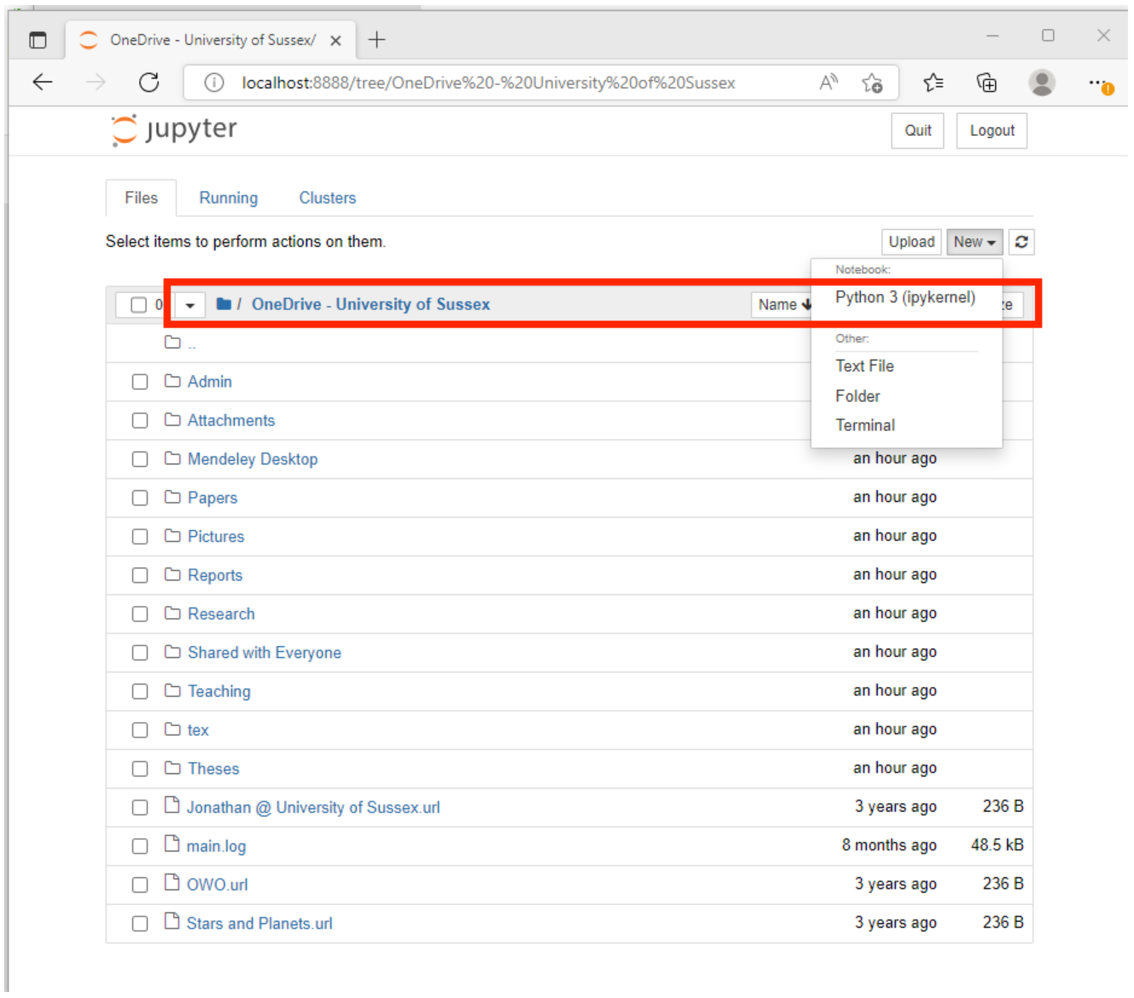
This will open a Jupyter Notebook session in your browser. This is where we will be doing the majority of our work in this module.

Important: any Jupyter notebooks you create should be stored under the folder `OneDrive - University of Sussex`, otherwise they will only be saved on the particular PC that you are using. Open the `OneDrive - University of Sussex` folder and create a new sub-folder with a name such as `PIP_Python`. Open that sub-folder and create a new Jupyter notebook using the `New → Python 3` pull-down menu.



Select items to perform actions on them.

3D Objects	an hour ago
Chrome	4 minutes ago
Documents	an hour ago
Downloads	an hour ago
Favorites	3 years ago
Links	3 years ago
Music	3 years ago
OneDrive	a month ago
OneDrive - University of Sussex	an hour ago
Pictures	3 years ago
Saved Games	3 years ago
Videos	3 years ago



Select items to perform actions on them.

/ OneDrive - University of Sussex

..		
Admin		
Attachments		
Mendeley Desktop	an hour ago	
Papers	an hour ago	
Pictures	an hour ago	
Reports	an hour ago	
Research	an hour ago	
Shared with Everyone	an hour ago	
Teaching	an hour ago	
tex	an hour ago	
Theses	an hour ago	
Jonathan @ University of Sussex.url	3 years ago	236 B
main.log	8 months ago	48.5 kB
OWO.url	3 years ago	236 B
Stars and Planets.url	3 years ago	236 B

1.4 Running Python on your own computer

This section intends to help you install Python on your personal computer. Should you need to use Python on University computers rather than a personal computer the guide for this is above. It is worth giving this a read anyway even if you are using a personal computer as it contains some important information.

The easiest way to set up everything you will need for the assessments and labs on your laptop or home PC, is by installing the Anaconda software package). Anaconda is a ‘distribution’ of Python that has several extra features that make it easier to improve Python’s capabilities; one of the most important is its package manager, which we’ll cover in detail later.

Later in this module, you’ll be introduced to **packages**. These are pieces of code that other people have written to expand Python’s capabilities. One of the main strengths of Python is the vast array of packages that can be imported to make your life easier, why reinvent the wheel? For instance, there are specialist modules for Astrophysics that make it easier for us to do common calculations and operations, such as loading images from telescopes or working with cosmological calculations. Python itself has a built-in package manager for installing new packages called `pip`, but Anaconda adds a more sophisticated package manager which makes sure that all of the modules are the right versions and will work together (amongst other things). But I digress, we will cover this in more detail.

NOTE: Anaconda is great when you are starting out. It will probably serve you well for the majority of your degree and maybe even beyond. HOWEVER, it does some nasty things in the background that can make it difficult to do more complex operations (particularly if you ever dabble in compiled languages like C and C++) and it hides a lot of complexity which is useful to learn in the long run. I purged Anaconda from my system years ago in favor of finer control over what I have installed. Of course, you don't need to worry about this now but in case you're confused down the line it is worth stating.

Once you've opened the website, you should: 1. Scroll down until you find the download button under the OTT "Unleash your innovation". 2. Download the executable for your operating system (On Mac you wil need to choose between the intel and apple silicon versions). 3. Follow the instructions in the executable.

If you do your assessments on a personal machine **please ensure that you have installed the correct version of Python** (a version >3.8). The installer should install the latest version by default but you can check by running `python --version` on the command line. If you fail to check this and your code doesn't work when we try to mark it then I'm afraid it's all on you!

To open a notebook on your machine you can either open Anaconda Navigator and open a notebook as shown above or open a notebook on the command line by first navigating to the directory where you want to open notebooks and then running `jupyter notebook` in the command line.