

Getting started with python 3 on MacOS

Python has two main versions currently in use: 2.7 and 3.8. Most versions of Mac come pre-installed with Python 2.7. You can check this by doing cmd+space bar then typing "terminal" in the search box, then typing python into the command line. You will likely see Python 2.7 is installed unless you have previously installed a different version.

However, we will be using Python 3.8 for this course, so please follow these instructions to install it:

The easiest way to do it is to install anaconda. Instructions to install Anaconda, which has the latest python distribution, can be found [here](#):

If you want to be able to run python programs in the command line with this installation of python, by just writing "python filename.py", then you have to select "Add Anaconda3 to my PATH environment variable". If you are fine with just using an IDE like VS Code to run your programs, then you don't need to select this option.

I'd advise you to choose the default option for "Register Anaconda3 as my default Python 3.8", that is, leave it checked. This will make it easier for IDEs like VS Code to find this Python installation.

IDE

The IDE (where you will actually write your code) that we recommend is VSCode, which is free and has many useful features. You can download it [here](#) and find a useful python tutorial [here](#)

If you prefer to use jupyter notebooks, check out the instructions below.

Installing jupyter notebook

Jupyter notebook is a popular interface tool that is useful for coding and data visualisation, and runs by typing code into cells that can be executed one after the other. It's particularly useful for visualising your data on the fly, which helps if you regularly need to plot your data or want to see it's dimensions without rerunning all of the code.

To install jupyter notebook using anaconda, simply type this into the command line:

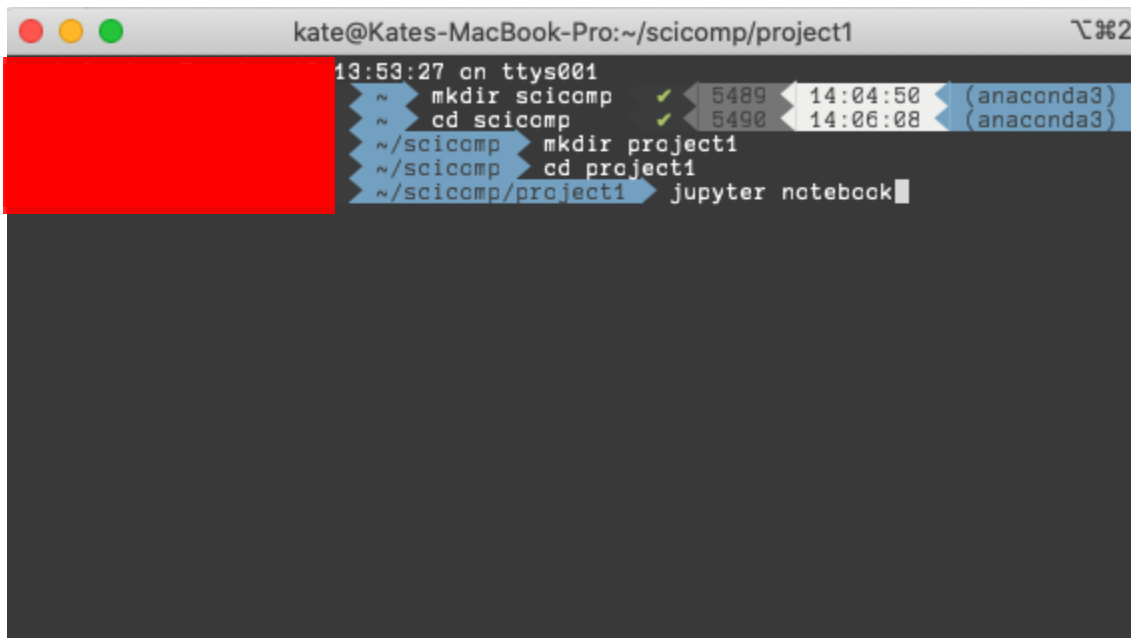
```
conda install -c conda-forge notebook
```

If this doesn't work, or you need to try an alternative way, the [jupyter notebook](#) page is very useful.

Example to open a notebook in a chosen directory

You can then type “jupyter notebook” into the command line and the notebook interface will open in a web browser. To open a notebook inside a chosen directory, check out the following example:

Say you want to create a folder for this course, with a subfolder for the first project. You can do this by opening the terminal and first creating a new folder in the command line, by typing “mkdir scicomp”, which stands for “make directory scicomp”. Move into this directory by typing “cd scicomp”, where cd stands for “change directory”. You can then make a sub folder called “project 1” by typing “mkdir project1” and cd into that. Finally, you can open a jupyter notebook and start writing code!

A screenshot of a macOS terminal window titled 'kate@Kates-MacBook-Pro:~/scicomp/project1'. The terminal shows a sequence of commands and their outputs. A large red rectangular box obscures the first two lines of the terminal output. The visible lines are: '~ mkdir scicomp' with a green checkmark, '~ cd scicomp' with a green checkmark, '~ /scicomp mkdir project1', '~ /scicomp cd project1', and '~ /scicomp/project1 jupyter notebook'. On the right side of the terminal, there is a log of system events showing the creation of the 'scicomp' and 'project1' directories at 14:04:50 and 14:06:08 respectively, both using the 'anaconda3' environment. The terminal prompt is at the end of the last command line.

```
kate@Kates-MacBook-Pro:~/scicomp/project1 ㉿ 2
13:53:27 on ttys001
~ mkdir scicomp ✓ 5489 14:04:50 (anaconda3)
~ cd scicomp ✓ 5490 14:06:08 (anaconda3)
~/scicomp mkdir project1
~/scicomp cd project1
~/scicomp/project1 jupyter notebook
```

IMPORTANT - It's good practice to select “restart kernel and run all” if you need to run the code, and you should avoid running cells out of order, which can lead to easy mistakes or reassigning variables by accident.

