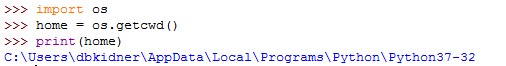
**PYTHON Programming Exercises 7**

We’re going to take a look at some modules that come hidden away in the depths of Python.

**Task 1: The OS Module**

The OS module lets you interact directly with the built-in commands found in your operating system. Commands will vary depending on the OS you’re running, as some will work with Windows; others will work with Linux or macOS.

Open the Python IDLE and at the command line, type:



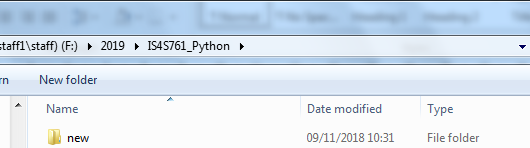
You can change the current directory to your workspace directory, e.g. …



BUT USE YOUR WORKSPACE or USB path!!! Whilst listdir creates a list containing the names of the entries in the directory given by the path. The list is in arbitrary order. It does not include the special entries '.' and '..' even if they are present in the directory. (Possibly useful for producing a list of filenames, such as Crime Month csv datasets?).

There are other directory manipulation commands such as making a new directory:

>>> os.mkdir(“new”)



Renaming a directory:

>>> os.rename(“new”, “old”)

And deleting a directory:

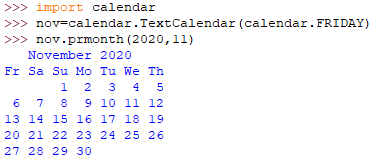
>>> os.rmdir(“old”)

Also, we can use the os.system() function to interact with external programs:

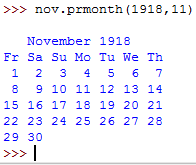


Essentially, we can get Python to run a variety of commands or other programs from within our current program.

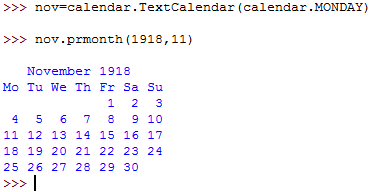
**Task 2: The Calendar Module:** Type the following:



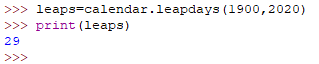
The Calendar module allows us to see the days of the month (November 2020) displayed in a wall calendar fashion, starting on a Friday (IS4S761 days). Naturally, you can change the (2020,11) part to any year and month you want. Try it with YOUR Birth Year and Month, or any other memorable date:



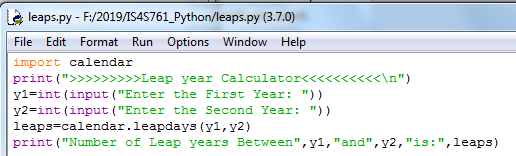
Change the start day to a Monday:



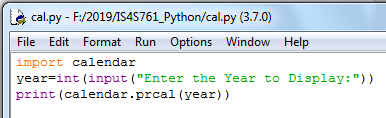
There are numerous functions within the calendar module that may be of interest when forming your own code. For example, the number of leap years between two specific years:



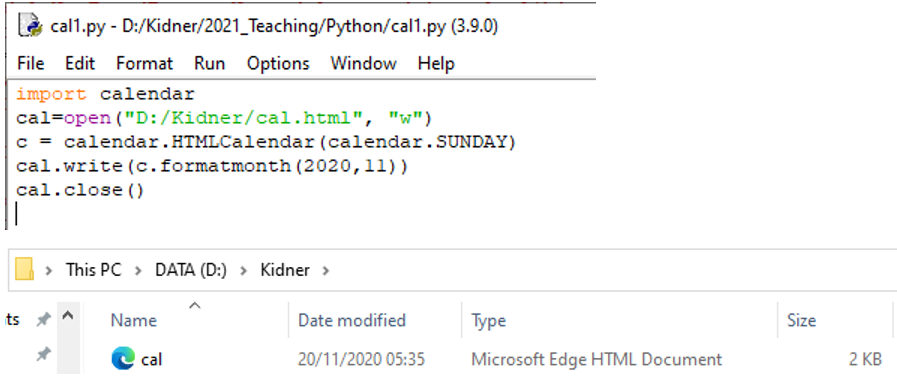
Which we can develop into some more useful code:



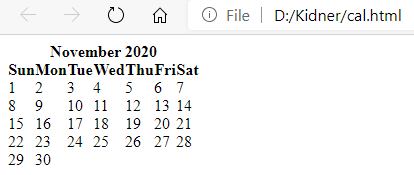
…or a calendar for a given year:



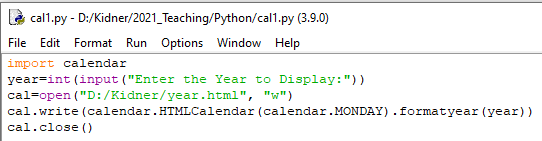
The calendar module also allows us to write the functions in HTML, so that you can display it on a website. Let’s give it a go with this code (***MAKE SURE YOU SET THE PATH NAME TO YOUR OWN!!***):



Which will create the cal.html file in your workspace. Double-click on it and …



Now amend the code to produce a web-page calendar for any given year:



**Task 3: The Time Module:**

Quite often we might want to judge the performance of our code – and whether it is efficient or not.

For example, suppose we wanted a list where each item is a product of the multiplication of the items of another list by two. You can either use a for loop or a list comprehension. Here is how you would check the time of each of the scripts:

**Using a for loop (type the following):**

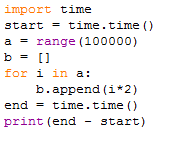
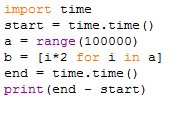


Figure out what the code does and then Run the code and make a note of the time it takes to execute:



Which on my PC takes about 5 one hundreds of a second to compute. WOW? Can you figure out a way to speed up the program (i.e. directly set the values into the list, rather than define a list and append values independently). Run the following:





Double WOW! This code is nearly twice as quick as before. OK, not really an issue for 100,000 calculations, but what about 100,000 billion calculations? This is something that we will need to think about for handling BIG DATA!

***EXERCISE 1: Experiment with these functions (os, calendar, time) and some of the others in the lecture, e.g. random, glob, and timeit.***

**EXERCISE 2:** Download and uncompress a whole 12 months (or 36 months) of crime data from the police.uk website and store the individual folders in one master folder “crime” in your workspace. Develop a Python program to generate a list of all of the individual filenames including paths of each CSV file. (This might be handy for your coursework).

**PYTHON Programming Exercises 8**

A very quick 10-minute exercise to introduce you to Jupyter Notebook, Pandas and Folium. This will get you started with the Coursework data, but it is up to YOU what you do! The exercise assumes that you have installed Anaconda.

**Task 1: Install Folium**

As we might want to visualise our data in the form of a map, let’s install Folium. In the **Windows Start menu > Anaconda3 (64-bit) > Anaconda Prompt (anaconda3).** This will open a little prompt window. Type …

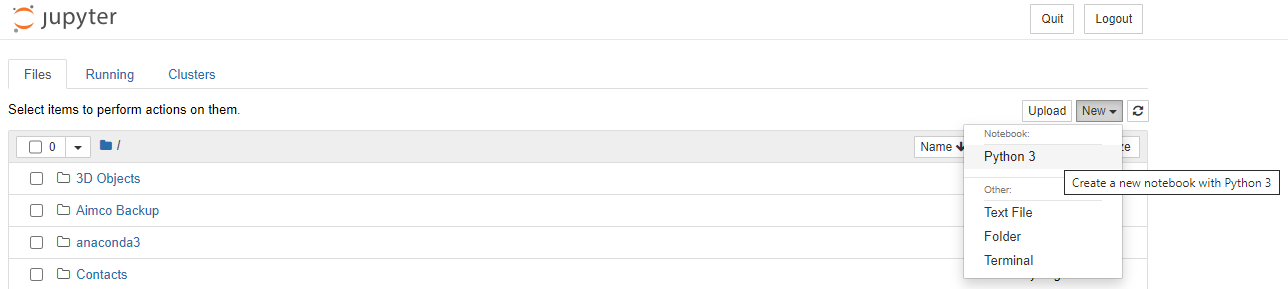


And click **y** to proceed (if prompted).

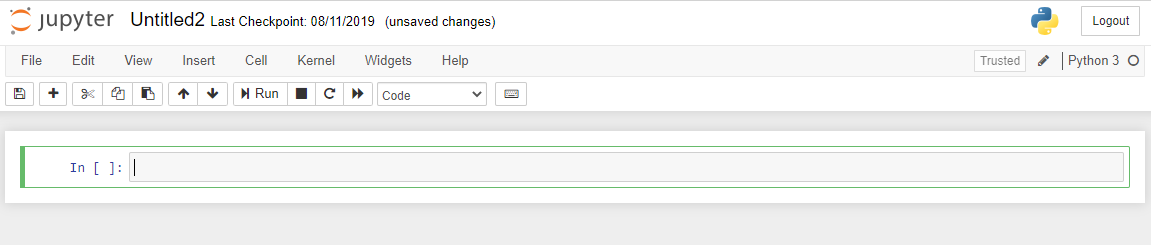
This is a one-time operation.

**Task 2: Open a New Jupyter Notebook**

In the **Windows Start menu > Anaconda3 (64-bit) > Jupyter Notebook (anaconda3).** This will open up the Jupyter envronment. Select New > Python 3 …



And this will open a new web page …

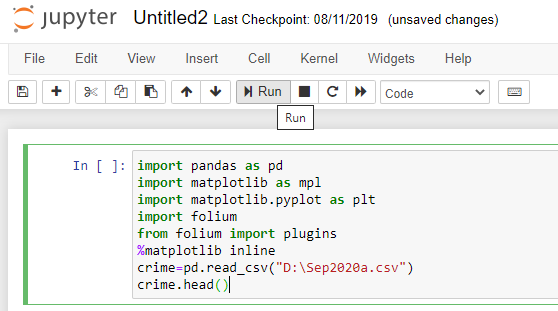


So we’re ready to start our coding. (IN YOUR OWN TIME, check out the HELP and other online guides to using Jupyter Notebook).

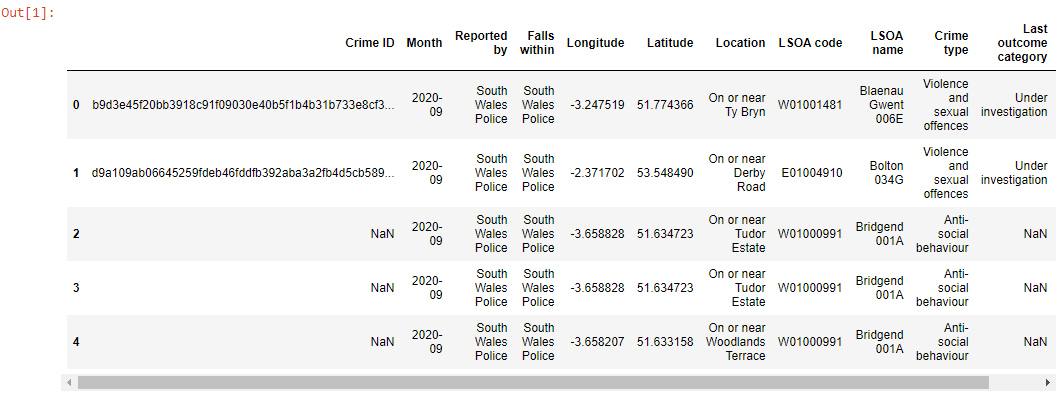
**Task 3: Open a Crime Dataset, Store it, Manipulate It, Visualise It as a Bar Graph & Map It!**

For this exercise, we will assume that you have downloaded ONE Crime dataset into your current working directory (see pervious exercise for changing directories, etc.). In this example, the dataset is renamed as **Sep2020.csv** and stored in the D: folder. Change the code accordingly for your path directory and filename.

In the first instance we will be reading the dataset into a DataFrame called crime and print out the first few lines …

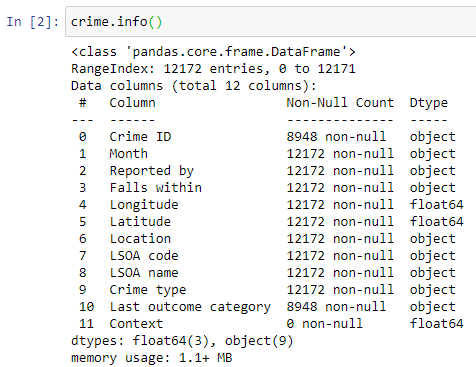


And click Run …

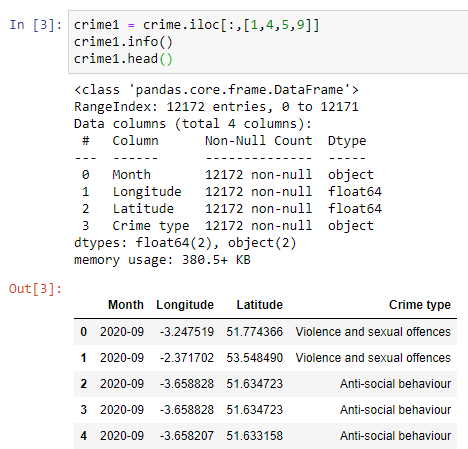


You will probably prefer to use your existing code from previous weeks! (Or use Pandas?).

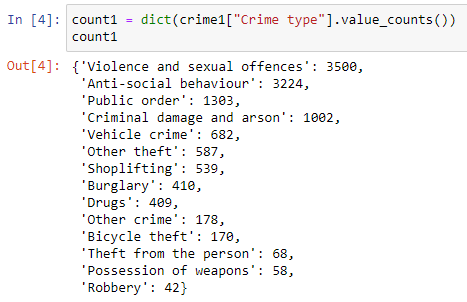
Type in> crime.info() & Run. What does this tell us?



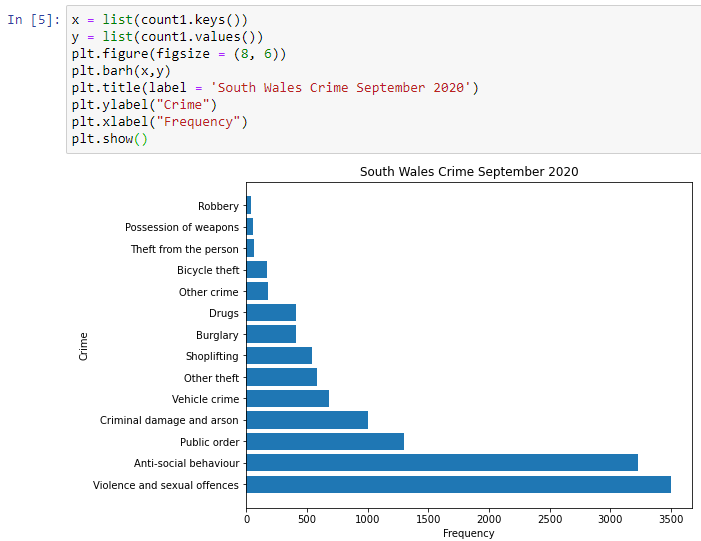
As in the previous tutorial exercises, we might only be interested in using certain columns of this data, such as Month (if we are using 12 or 36-months of data), Longitude and Latitude (for Location), and the Crime type. These are in column locations 1, 4, 5 and 9.



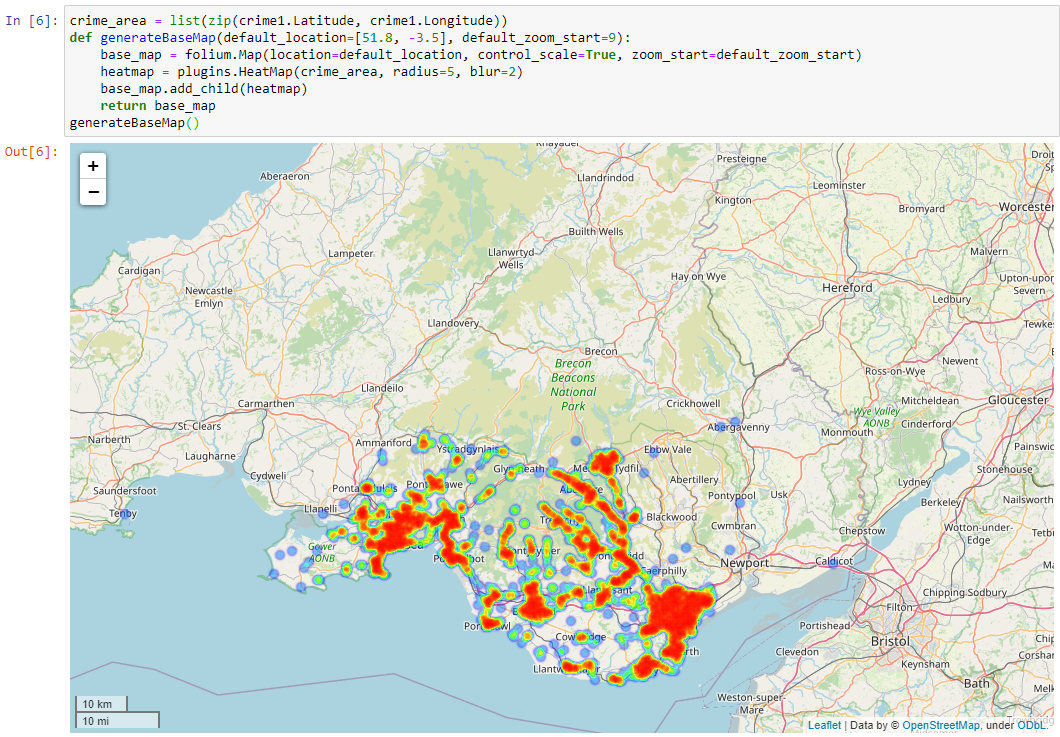
OK, that’s probably a bit easier than manipulating our Python lists, but how do we go about counting up Unque Crime Types …



Let’s visualise this data now in MatPlotLib (as we did last time), but this time we’ll use a bar / column chart …



… and finally let’s map our crime locations as a Heat Map. (Please take your time to find out what the code does and how you can change some of the arguments). Your Location Data may have Null Lat/Long values, so you will need to find a way of stripping these out beforehand …



**SAVE YOUR JUPYTER NOTEBOOK AND EXPORT IT TO A PDF AS WELL!!!**