**PYTHON Programming Exercises 5**

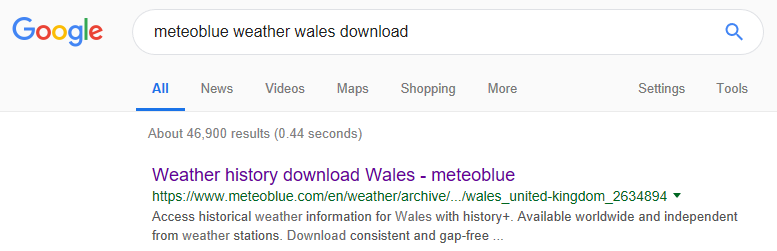
The solutions to the Exercises in the previous session’s tutorials (Python Programming Exercises 2-4) are now up on Blackboard. Take a look at these and make sure you understand the logic behind the solutions. In most cases, you can probably identify where the code can be improved? (e.g. remembering to close files!!).

**Task 1: Reading CSV Files**

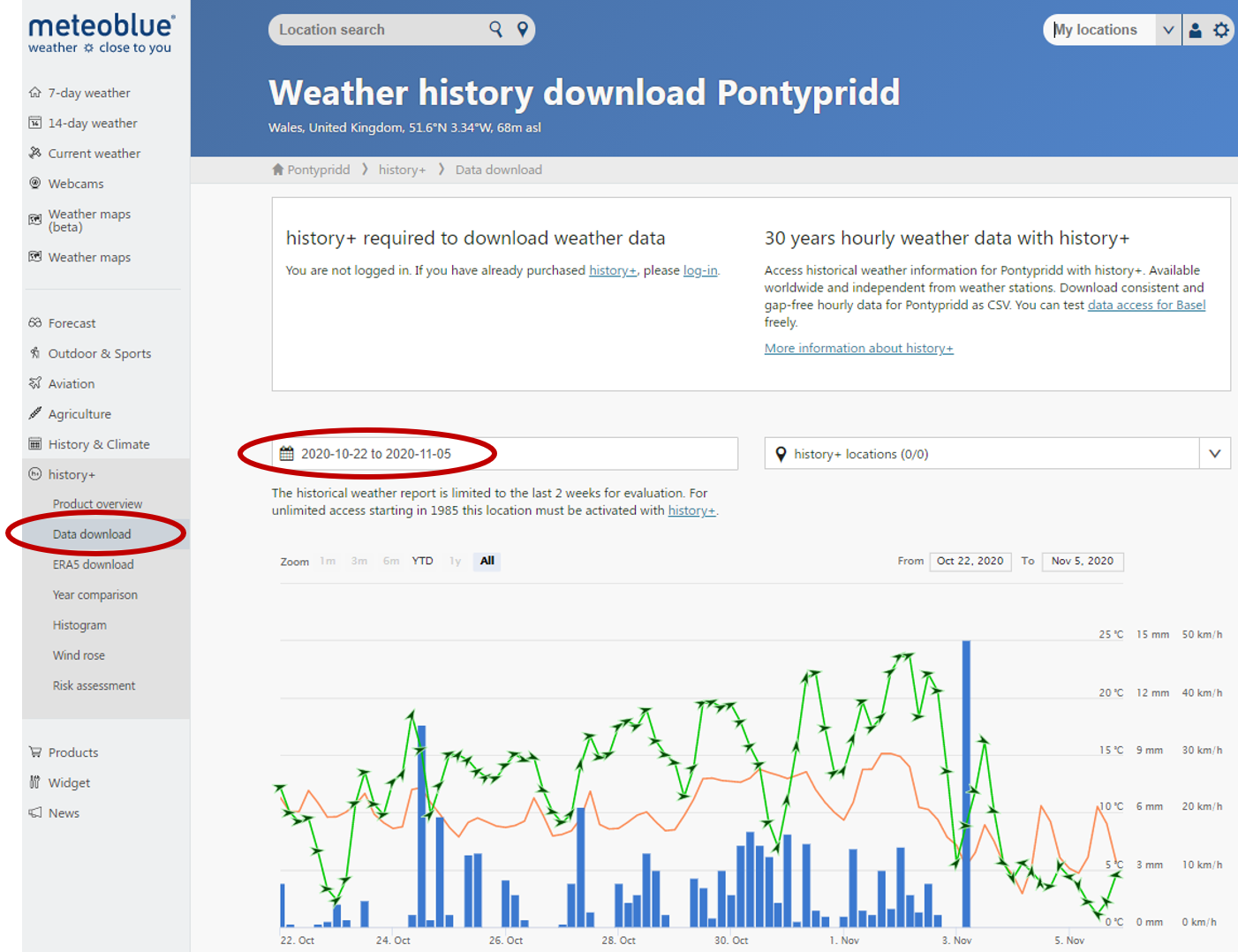
We want to download the last week’s weather data for Sunny Pontypridd and analyse this using Python. Use the following link to go to the meteoblue website:

<https://www.meteoblue.com/en/weather/archive/export/pontypridd_united-kingdom_2640104>

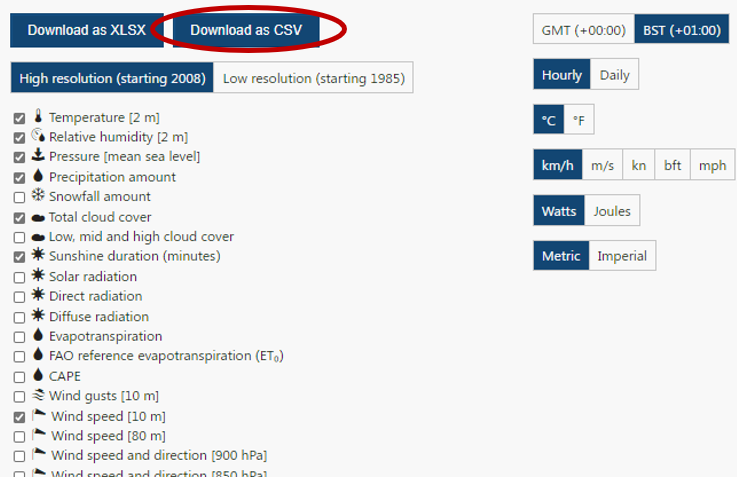
or Google meteoblue weather wales download



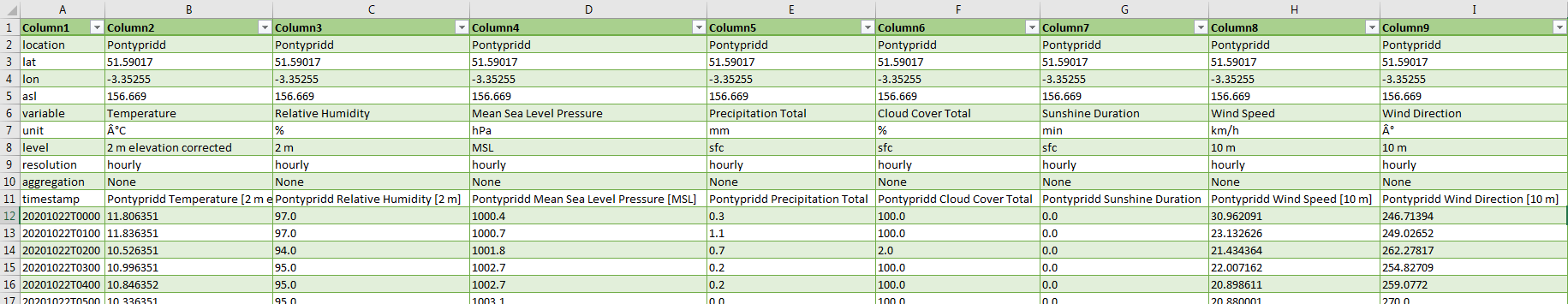
Which should take you to the following website. Type PONTYPRIDD into the search window; click the Data Download link (inside history+ on the left hand tab) and set the Date Range to the last 2 weeks:



We are going to select a subset of data fields to download. Select TEMPERATURE, RELATIVE HUMIDITY, PRESSURE, PRECIPITATION, TOTAL CLOUD COVER, SUNSHINE & WIND SPEED and download the generated as a CSV file:



Load the data into Notepad to explore what the data looks like. Try loading the data into Excel for a better look at the data. (**If Excel struggles, then go to the Data tab and select From Text/CSV; select your downloaded file and Get Data**).

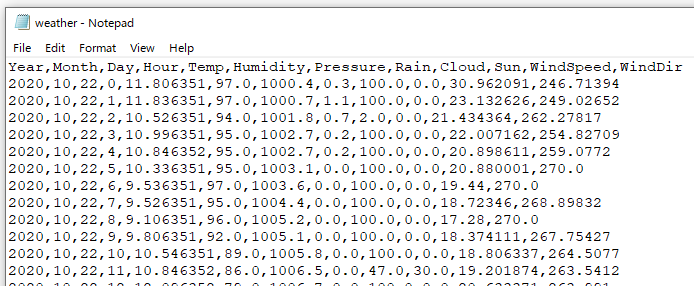


We now want to analyse this data with respect to identifying trends in the data; the total rainfall; the total sunshine; average temperature; minimum and maximum temperature; the day and time of the highest temperature; and so forth.

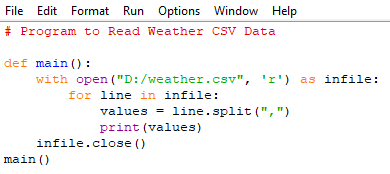
For the time being, we are going to use a simplified version of the data covering the last two weeks (up to and including midnight last night) at hourly intervals. Either:

1. remove the header records, simplify the column titles and save as a CSV file (and if you are confident with Excel, change the TimeStamp to Year, Month, Day, Hour) **OR BETTER STILL**
2. download th#e file **weather.csv** (from Blackboard to your workspace or USB stick).

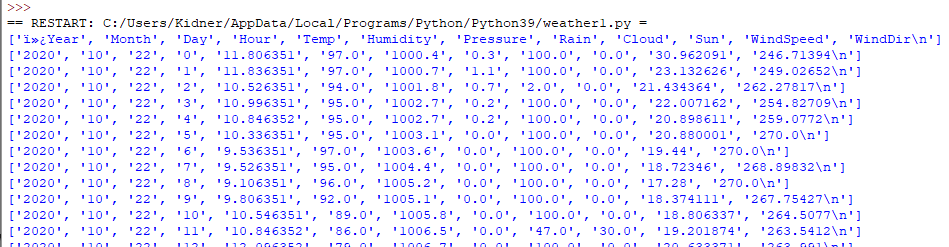
The fields include **Year**, **Month**, **Day**, **Hour** (Start Time), **Temp** (Temperature in Degrees Celsius), **Humidity** (Relative Humidity as a %), **Pressure** (at Mean Sea Level in hPa), **Rain** (Total Precipitation in mm), **Cloud** (Total Cloud Cover as a %), **Sun** (Sunshine Duration for that hour in minutes) and **WindSpeed** (in Km/h) and WindDir (Direction of Wind in Degrees).



In the Python IDLE, open a New File and type in the following code:



**Make Sure you define the path to your workspace, i.e. “D:/weather.csv” to C:/users/30xxxxxx/weather.csv”** *(where 30xxxxxx is your enrolment number or Local Drive).*Save the program as **weather1.py** and run it …

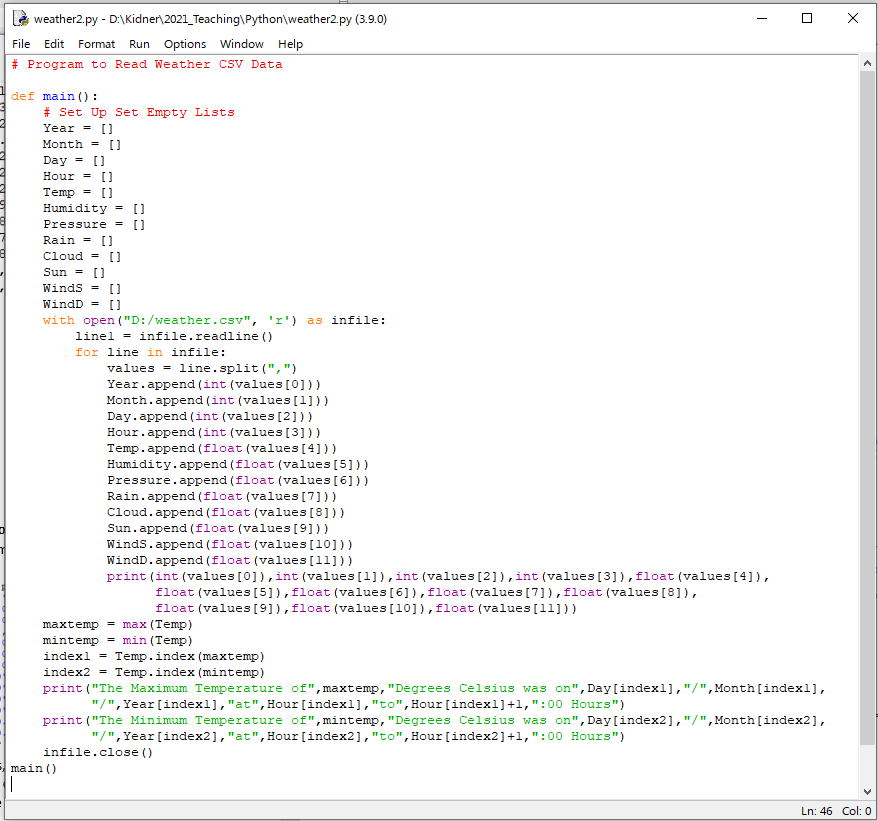


The program reads each line of the file and splits the line into a list of values, where the split or delimiter is defined as a comma. We could append each of these individual values into a list for each field. However, as the data is read in as strings, then we need to convert the values to an appropriate datatype (e.g. an Integer or Float). When the data are organised into the separate lists, then we can analyse the data.

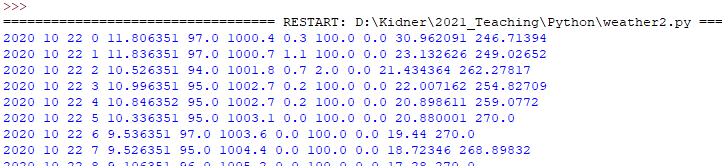
Download the file **weather2.py** (from Blackboard to your workspace or USB stick).

Edit the program to point to your workspace directory where the weather.csv is stored (i.e. NOT D:/).

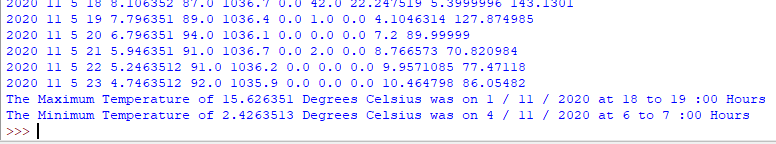
Take a few moments to understand what the program does.



RUN THE PROGRAM!



…

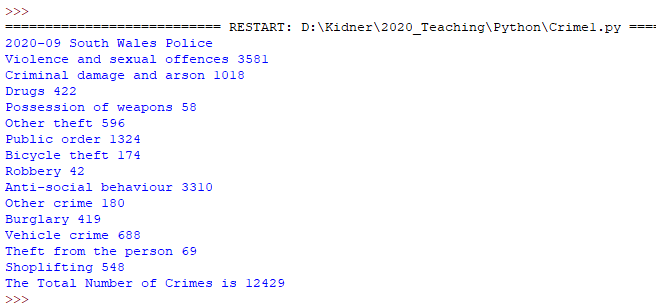


**EXERCISE 1:** Expand upon your weather2.py to also calculate:

1. The Total Rainfall over the period.
2. The Largest Amount of Rainfall in any 1-Hour Period (in mm and the Date and Time)
3. The Total Amount of Sunshine in minutes over the period.
4. The Average Temperature over the period.
5. The Largest Amount of Rainfall in any 1-Day (24 Hour Period) (Quantity and Date).
6. The Largest Amount of Sunshine in any 1-Day (Quantity and Date). (Not a trick question).

**EXERCISE 2:** If you haven’t already done so, take a look at **PYTHON CW Tutorial 1** with respect to the Police UK Crime datasets. Download as a CSV file, the South Wales Police crime data for **SEPTEMBER 2020**. Have a look at the data for yourself.

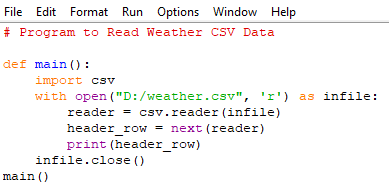
Write a Python program (**NOT using the CSV library**) to read in this dataset into a series of lists and which sums up the total number of crimes of each unique type; and the sum of all crimes. For example, the output might look like:



**THINK ABOUT YOUR ALGORITHM FIRST!** With Pen and Paper (remember them?), jot down what it is that you need to do and how you can break down the problem into achievable tasks. Use **weather2.py** as a starting point.

**Task 2: Reading CSV Files with Python’s CSV Library**

Now we are going to use Python’s csv module in the standard library to parse the lines in a CSV file. This should allow us to quickly extract the values we’re interested in. As before, let’s start by examining the first line of the file (the data headers). Save the following program as **weather3.py** and run it …

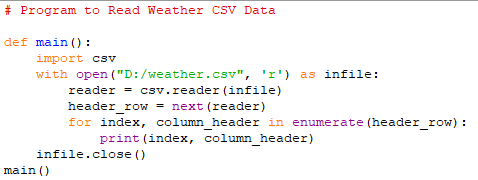




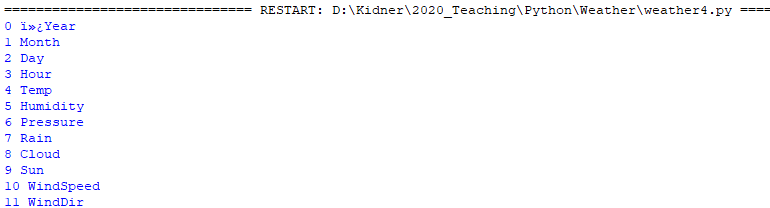
Why Not ‘Year’?

In this program, after importing the CSV module, we store the file that we’re working with as the file object **infile**. Next, we call the **csv.reader()** and pass it the file object as an argument to create a **reader** object. The csv module contains a **next()** function, which returns the next line in the file when passed the reader object. We only use this once to read the first line of data which is returned as the **header\_row**.

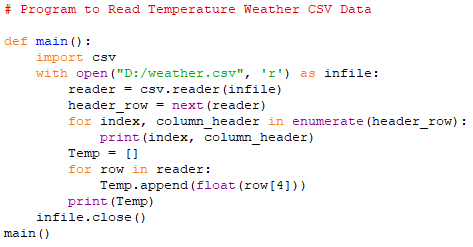
To make it easier to understand the file header data, print each header and its position in the list:



We use **enumerate()** on the list to get the index of each item, as well as the value. (Note that we’ve removed the line **print(header\_row)** in favour of this more detailed version).



Let’s assume we are only interested in the Temperature data for now (Index Position 4). Like before, we can define an empty list (Temp) to store these values:



We then loop through the remaining rows in the file. The reader object continues where it left off in the CSV file. On each pass through, we append the data from index 4 (the 5th column) and convert the value from a string into a float. The list of values is then output:



Make sure you save this file, as we will be using it again later.

**EXERCISE 3:** Following on from EXERCISE 2, use the CSV module to read the **September 2020** crime data for South Wales Police and just generate the list of Crime Types and the same output as before (i.e. total of all crimes for the month).

**EXERCISE 4:** The File **ponty2019.csv** includes the hourly weather data for Pontypridd for the whole of 2019! Take your program from Exercise 1 and adapt it to calculate the same weather statistics (together with the total snowfall) for the whole of 2019.