**Working with Files in Python**

*Using files allow you to permanently store data when your programme is not running*

**Year 10 Work – text files**

1. Read through the ***files*** document in the Python examples\5 – File Handling folder.
2. Look at the ***Simple Text File*** Python script. Save it into your own folder and run it. Experiment, writing and reading data to a text file. Make sure you understand it and can:

* Write data to a file
* Read data from a file
* Writing separate lines of data to a file
* Reading separate lines of data back

1. Make sure you have looked at ***NestedListsBasics.py*** on O:

**Year 10 Work – csv files**

**Nested lists in Python represent data stored in a grid (or 2D array). CSV files use the same grid format, so nested lists and CSV files are naturally related. Data read from a CSV file into Python is placed into a nested list. Data from a nested list in Python can be easily written into a CSV file.**

1. Examine both of the ***SimpleCSV.py, CSV handler.py,*** Python scripts (Python examples folder 5). Save them into your own folder and run them. Experiment writing and reading data to a csv file. Make sure you understand these code examples and can:

* Write data to a csv file
* Read data from a csv file

1. Examine ***NestedListsBasics.py*** (Python examples folder 4). Save it into your own folder and run it. This gives you examples of how to control nested lists in Python.
2. Examine ***ListOperations.py*** (Python examples folder 4). Save it into your own folder and run it. This gives you examples of how to control nested lists within Python and read/write them from/to CSV files. Advanced programmers should look at ***list comprehension*** as a powerful way of creating lists (see **pythonforbeginners.com** for examples).

**CSV Task**

Create a Python script that contains/does the following:

* Generate a nested list equivalent to a 10x10 multiplication square.
* Write the multiplication square data to a csv file
* Read the multiplication square data back from the csv file. Printing the data as a tidy grid with column headers (Col 1, Col 2, etc.) and row headers.

***NestedListsBasics.py –*** use this to get the printing perfect. Use the example beginning with the comment: ***#without the print end trick #using string formatting***

***ListOperations.py & CSV handler.py –*** use these for examples of how to write data to a csv file and read it back.