Lab 4 – Advanced Python

Prior to commencing with the tasks, ensure that you:

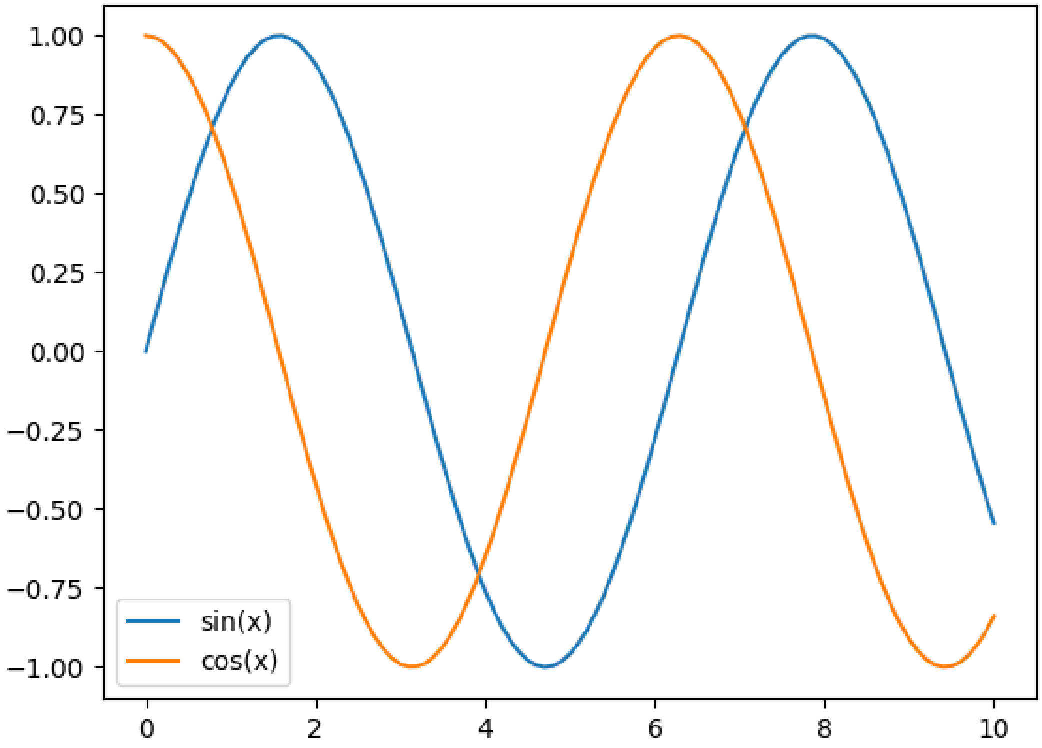
1. Synchronise the time on the Pi, either by connecting to a different network, i.e. a Hotspot, or by using: `sudo date -s 'YYYY-MM-DD hh:mm:ss`
2. Clone the Lab4 Repository from GitHub Classroom into the ee347 folder, using this link.
3. Commit and Push to GitHub after each task. Each task should be completed in the individual taskX.py/taskX.ipynb scripts provided.

Tasks

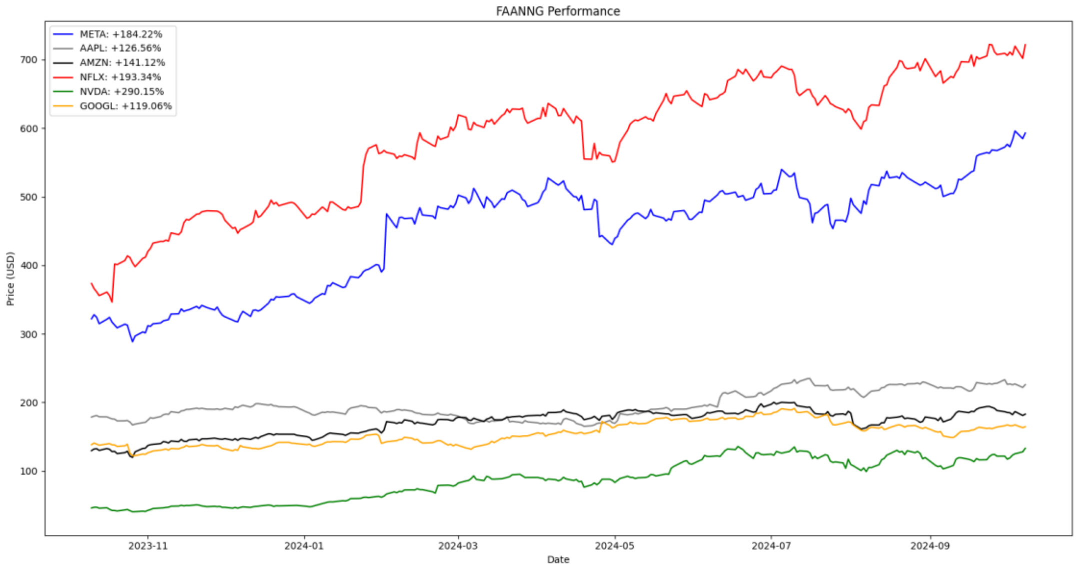
1. Open task1.ipynb. This is a Jupyter Notebook, which can be useful for prototyping and debugging complicated code, as cells can be executed individually while retaining access to each other's variables. Add a cell that prints 'Hello World!' to the screen. Run the cell and observe the output. You may need to select the kernel. In this case, select the virtual environment provided in the ee347 folder.
2. task2.py contains the outline of a `RandomNumberGeneratorClass()`. Using the random package already imported, finish the code and run the file.
3. Edit task3.py to ask the user their name, and append that name to task3.txt. Add 3 names to the file
4. Edit task4.py to read each name from the file and print them to the screen. Verify the names match those provided as inputs in task 3.
5. Using the csv package, ask the user for names until they type quit. Write each name as a separate row in task5.csv, before reading the names from the file and printing to the screen.
6. Numpy is an incredibly powerful Python package, particularly useful for array manipulation. Using the numpy array provided, add code to replicate the output shown below:

```
Before:
[[1. 1. 1. 1. 1. 1. 1. 1.]
 [1. 1. 1. 1. 1. 1. 1. 1.]
 [1. 1. 1. 1. 1. 1. 1. 1.]
 [1. 1. 1. 1. 1. 1. 1. 1.]
 [1. 1. 1. 1. 1. 1. 1. 1.]
 [1. 1. 1. 1. 1. 1. 1. 1.]
 [1. 1. 1. 1. 1. 1. 1. 1.]
 [1. 1. 1. 1. 1. 1. 1. 1.]]
After:
[[1. 1. 1. 1. 1. 1. 1. 1.]
 [1. 0. 0. 0. 0. 0. 0. 1.]
 [1. 0. 0. 0. 0. 0. 0. 1.]
 [1. 0. 0. 0. 0. 0. 0. 1.]
 [1. 0. 0. 0. 0. 0. 0. 1.]
 [1. 0. 0. 0. 0. 0. 0. 1.]
 [1. 0. 0. 0. 0. 0. 0. 1.]
 [1. 1. 1. 1. 1. 1. 1. 1.]]
```

7. Numpy is also useful for saving large amounts of data. Use numpy to generate sine and cosine waves of the given x values, and use `np.save()` to save them as task7_sin.npy and task7_cos.npy respectively.
8. Load the .npy files saved in task 7 and plot them using matplotlib, another useful package. Your output should look as below:



9. Load task9.csv, and plot the data it contains with matplotlib.
10. Tidy up the graphs from task 9, and match colours, etc as below:


FAANNG Performance
META: +184.22%
AAPL: +126.56%
AMZN: +141.12%
NFLX: +193.34%
NVDA: +290.15%
GOOGL: +119.06%
Price (USD)
Date