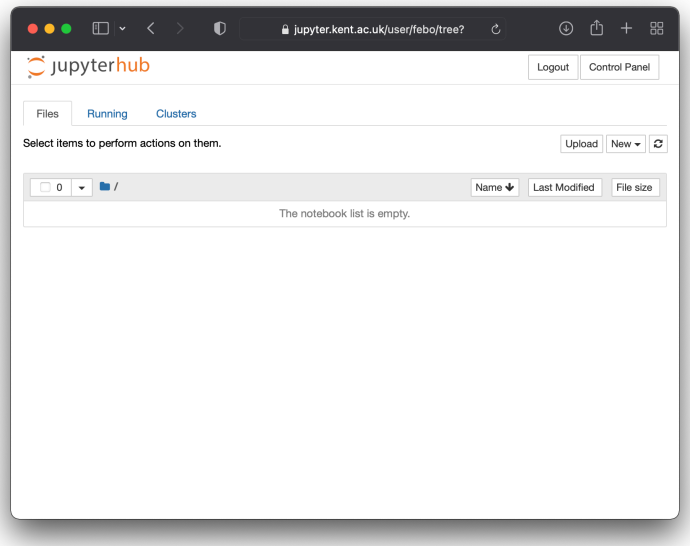
COMP8270
Programming for Artificial Intelligence

Class 1

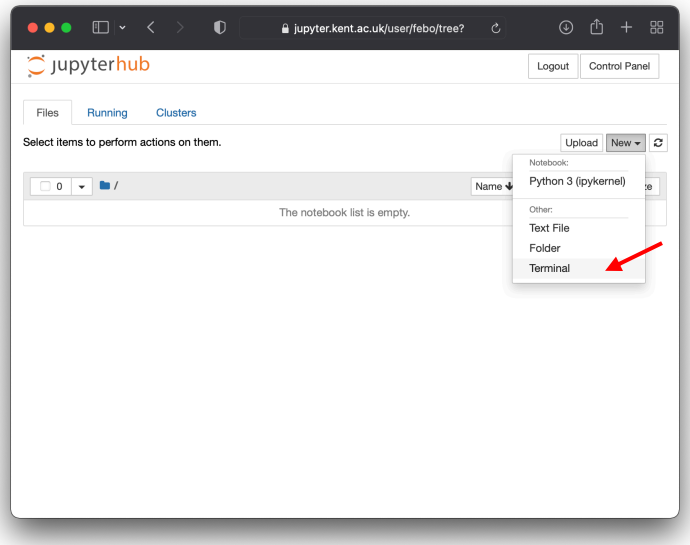
The aim of this class is to familiarise ourselves with the Jupyter environment and write python code in a Jupyter Notebook.

As a first step, you will need to log in into: https://jupyter.kent.ac.uk – this is the URL for the School of Computing Jupyter server. After you log in, you will see your 'Dashboard' section.

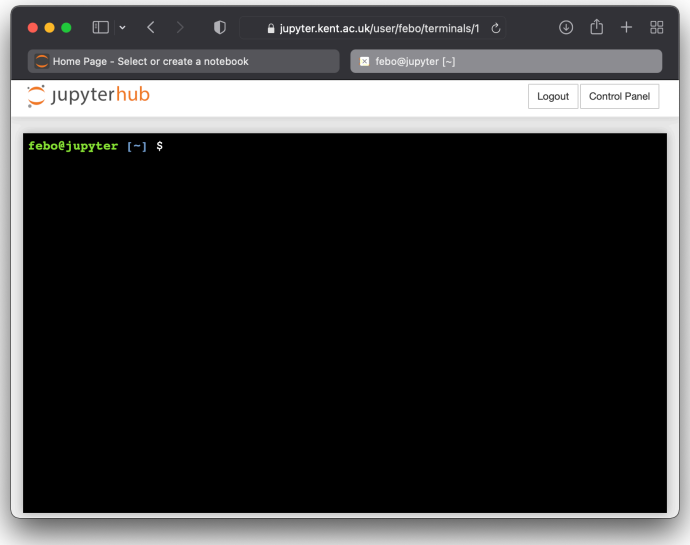


Task 1 – Setup the Quickstart guide

On the *Dashboard* page, click on New → Terminal:



This will open a terminal window on the server:



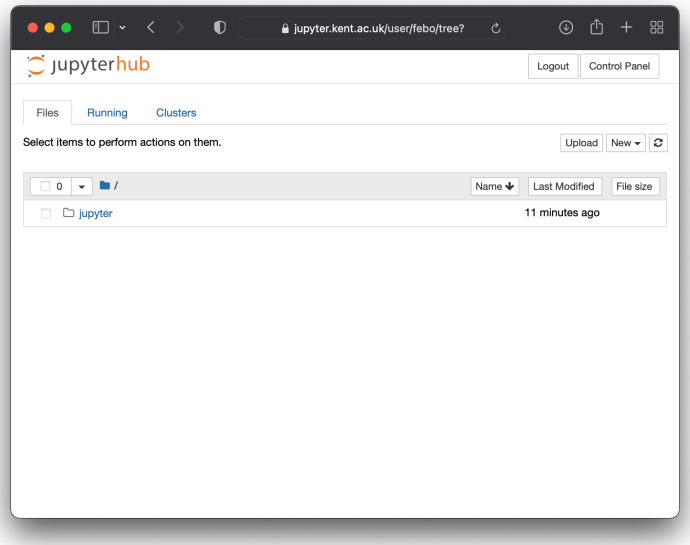
On the terminal, execute the following command:

```
$ git clone https://git.cs.kent.ac.uk/febo/jupyter.git
```

You will need to enter your Kent login details. The execution of the command will generate an output similar to:

```
Cloning into 'jupyter'...
Username for 'https://git.cs.kent.ac.uk': febo
Password for 'https://febo@git.cs.kent.ac.uk':
remote: Enumerating objects: 285, done.
remote: Counting objects: 100% (285/285), done.
remote: Compressing objects: 100% (247/247), done.
remote: Total 285 (delta 31), reused 285 (delta 31), pack-reused 0
Receiving objects: 100% (285/285), 5.76 MiB | 44.38 MiB/s, done.
Resolving deltas: 100% (31/31), done.
```

You can now close this tab and go back to the *Dashboard* page.



Click on the “jupyter” folder and open the notebook “Index.ipynb.” From this point onwards, explore the information available on the quickstart guide.

Writing Python code

1. The Fibonacci numbers form a sequence where each number is the sum of the two preceding ones:

   $F_0 = 0, F_1 = 1,$

   and

   $F_n = F_{n-1} + F_{n-2}$

   The Java code below prints the Fibonacci numbers smaller than 22. Your task is to create a Jupyter Notebook representing a Python version of the code.

```
public class Fibonacci {
    public static void main(String[] args) {
        int n1 = 0;
        int n2 = 1;

        System.out.println(n1);

        while (n2 < 22) {
            System.out.println(n2);
            int n3 = n1 + n2;
            n1 = n2;
            n2 = n3;
        }
    }
}
```

2. Write a Python code to create the multiplication table (from 1 to 10) of a number. The expected output is (for the number 6):

```
6 x 1 = 6
6 x 2 = 12
6 x 3 = 18
6 x 4 = 24
6 x 5 = 30
6 x 6 = 36
6 x 7 = 42
6 x 8 = 48
6 x 9 = 54
6 x 10 = 60
```

3. Go over the lectures slides and test out other examples of Python code (e.g., ternary operator, type casting). Try different variations to see what happens.