CSCI 4140 Laboratory One

Software Setup

Introduction

The purpose of this laboratory is to setup the software required for subsequent labs and make sure that all of it is working correctly. The software that we will be using runs on Windows, Linux and MacOS, once the software is set up all three platforms will work in basically the same way. There is a slight difference in the setup process between the platforms, but its quite straight forward.

Anaconda

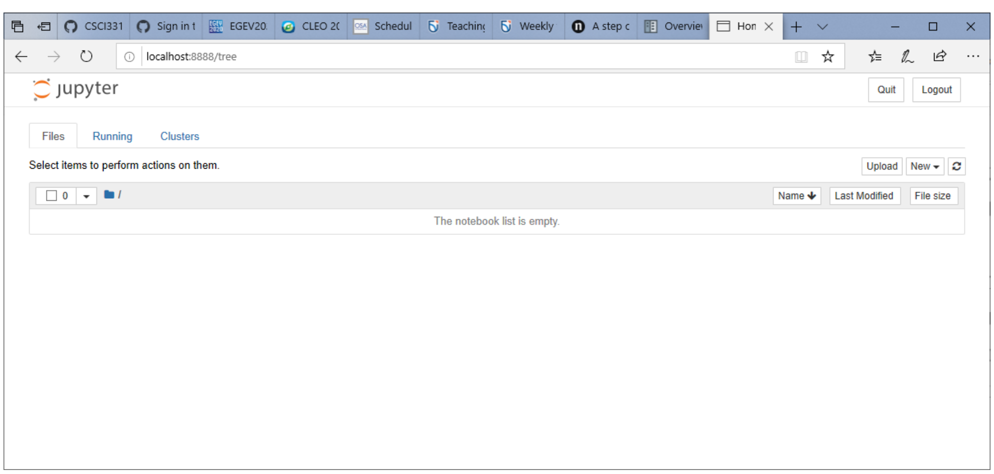
We will be using Python with Jupyter notebook for our programming. The easiest way to install this, and some of the other tools we will need is through Anaconda. Start the process by going to https://www.anaconda.com/products/individual and pressing the Download button. You must do the Python 3.7 install and select the installer for the platform you are using. You should select the 64-bit installer. Use all the default settings in the installation process.

Jupyter Notebook

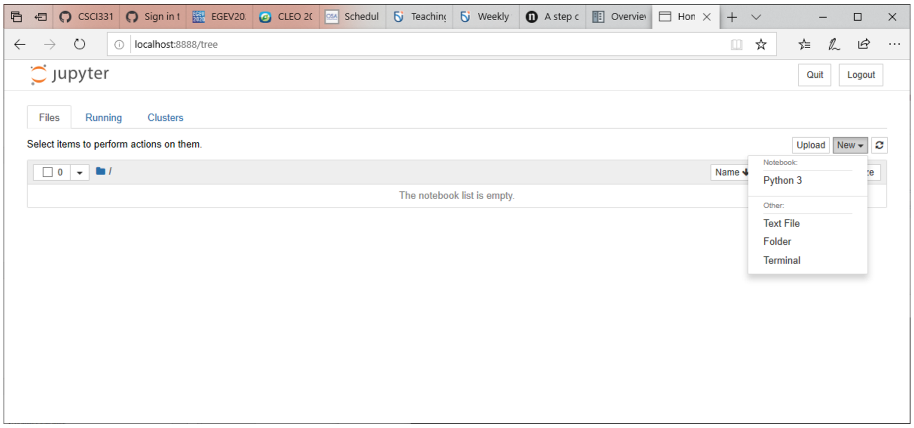
Jupyter notebooks are a way of organizing computation and text that describes them. A Jupyter notebook consists of a collection of cells, with each cell containing either Python code or text. The Python code can be edited and executed within a cell. This provides a good environment for code development as well as documenting the work that you have done. Jupyter notebook is based on web technology. You start a notebook server on your computer and then interact with it through a browser. To start a server, open a command or shell and navigate to the directory where you are storing your notebooks and then enter the following command:

```
jupyter notebook
```

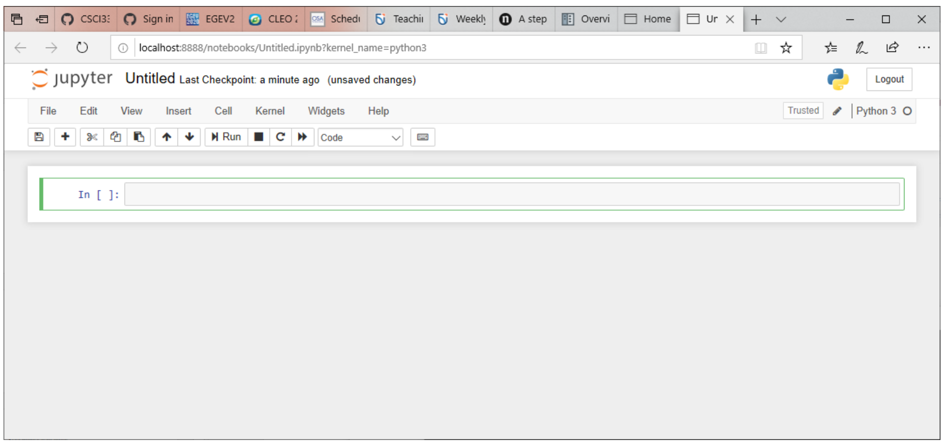
This will start the server and open a web browser with the following contents:



So far there are no notebooks, to create one select the new pull down menu and then a Python 3 notebook:



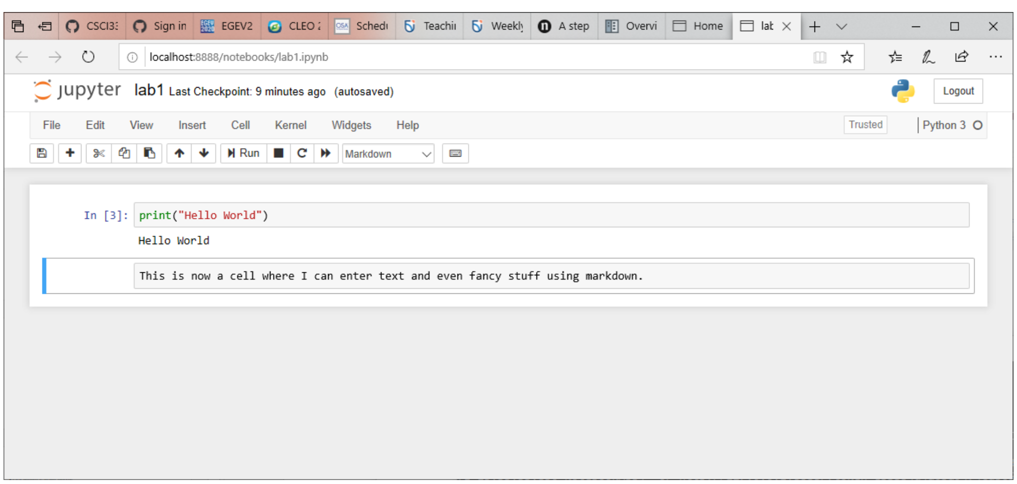
After doing this you will get a new browser window that you can start using as a notebook. It starts with a code cell where you enter Python code and execute it:



Try entering something like:

print("Hello World")

And then press the run button. You can create a new cell by pressing the + button. You can change this from a code cell to a markdown cell and enter text and HTML tags in the cell.



You can use the file menu to save the notebook. This produces an .ipynb file. This is a text file that can be submitted as lab report or assignment. More information on Jupyter notebook can be found at https://jupyter.org/documentation.

Qiskit

Qiskit is the software that we will be using for quantum computing. This is free software provide by IBM and provides access to the IBM quantum computing hardware in the cloud. Qiskit is an extension to Python and called be installed using pip. To install qiskit start an Anaconda Prompt and enter the following command:

pip install qiskit[visualization]

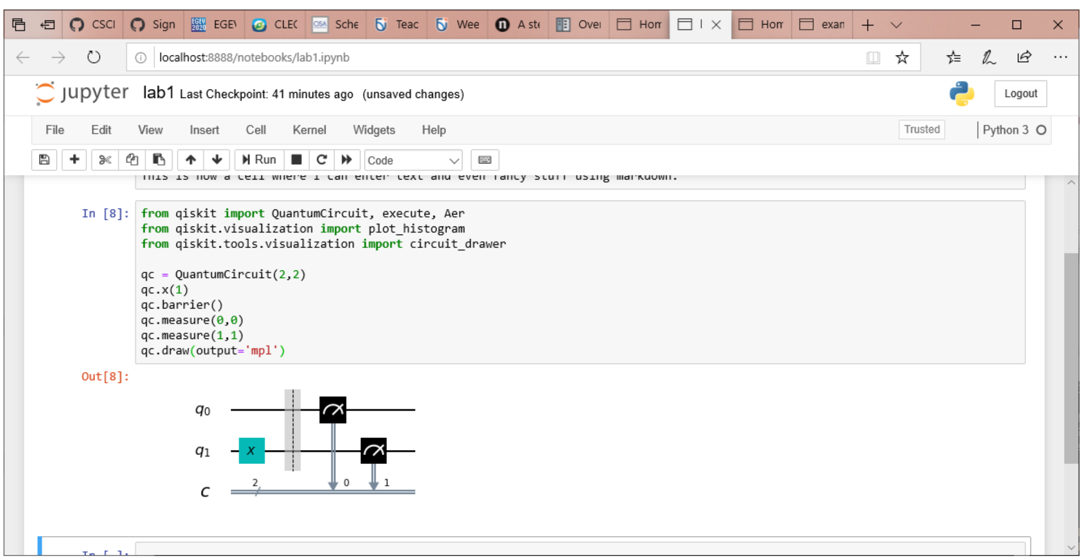
If you are using MacOS you need to enclose qiskit[visualization] in quotes.

You will also need to install the Qiskit textbook software using the following command:

python -m pip install git+https://github.com/qiskit-community/qiskit-textbook.git#subdirectory=qiskit-textbook-src

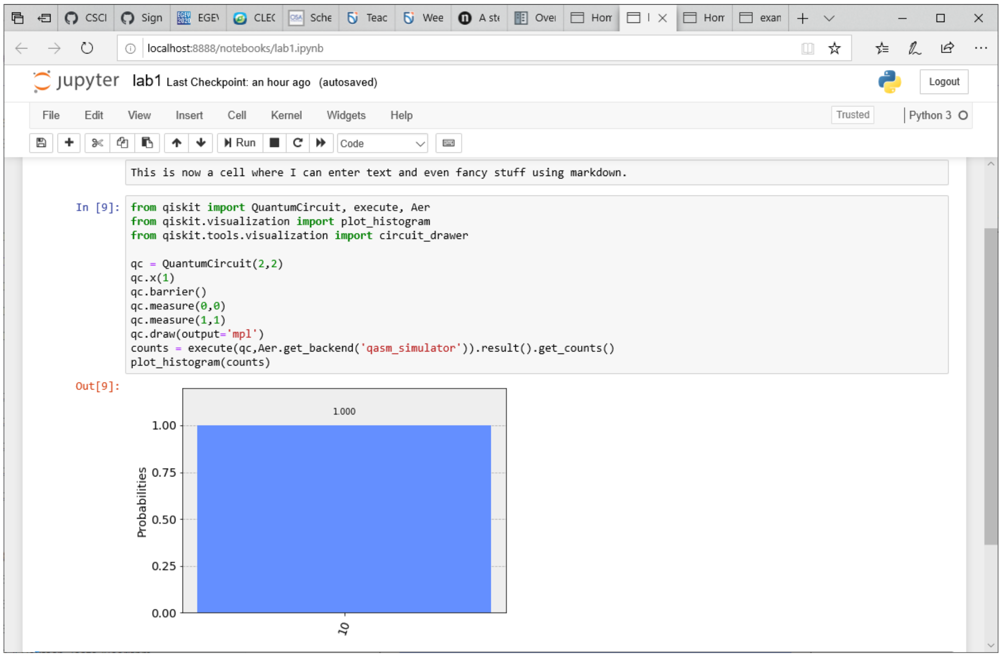
You may need to run this as administrator on Windows systems.

To see how this all works we will do a simple quantum circuit. Enter and run the following:



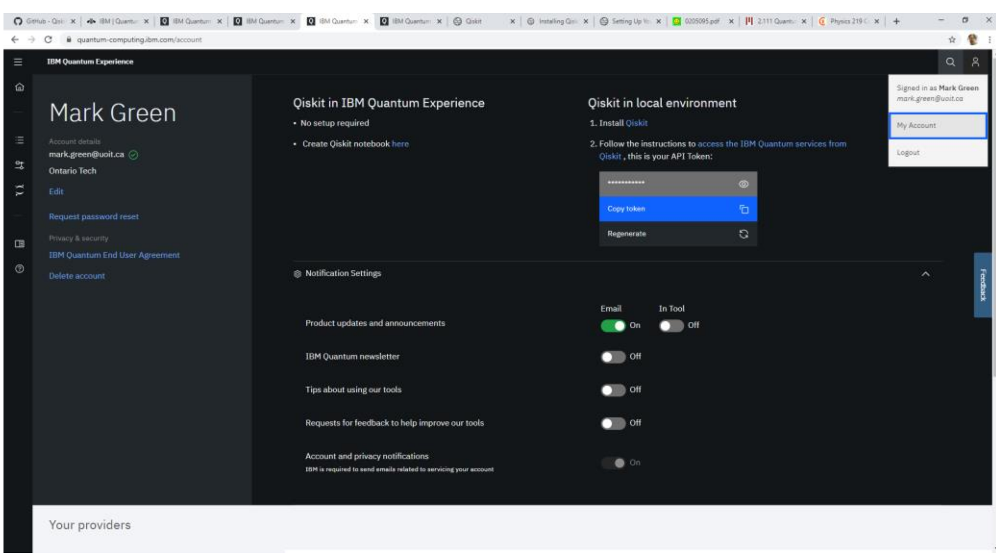
This gives us a simple circuit where the first qubit is in state |0> and the second qubit is in state |1>. This X get changes |0> to |1>.

Now we are ready to run this circuit. There are several ways that we could do this. To keep things simple, we will simulate the circuit on our computer. We could run this on real hardware, but this circuit is really simple, so there is no reason to do that. Qiskit provides us with a number of simulators, we will use one of the simpler runs that counts the number of times that particular values are measured for each qubit. Recall that quantum circuits are statistical in nature, so we need to run our circuit a number of times to get reasonable results. That's why we plot our results as a histogram. In this case there is no randomness in the circuit, so all the values come out the same.



IBM Quantum Experience

If you want to run your programs on a real quantum computer you will need an account at IBM. This is free, you just need to go through a registration process. Start by going to https://quantum-computing.ibm.com/ and then select "Create an IBMid account". Once you have created your account and logged in you need to find your account information. Press the person icon in the top right corner to access your account information. The page should look like this:



You will need to press Copy token to get the token you need to run on the quantum computers. In a window running python type the following:

```
from qiskit import IBMQ

IBMQ.save_account('TOKEN')
```

Replace Token by the token you just copied from your IBM account. Note that the token must be enclosed in ''. To check that everything worked enter the following:

```
IBMQ.providers()
```

You should get a single line with hub='ibm-q' in it. Now try the following

```
provider = IBMQ.get_provider(hub='ibm-q')

provider.backends()
```

This should produce about 8 lines of output with the quantum computers that are available to you. If you don't get any output you may need to enter IBMQ.load_account() just after you save it. This will store your account information on your local disk so you won't need to go through this process again.

Laboratory Report

For this laboratory just make a simple change to the circuit that we created above. Move the X gate from the second qubit to the first qubit. Run the example again, take a screen shot of the resulting Jupyter notebook and upload it for the laboratory.