**CE/CZ1104 Linear Algebra for Computing**

**Lab 0 – Jupyter Notebook and Python**

This lab must be completed by the end of Week 3 during your own time. The aim of this lab is to familiarize yourself (i) with Jupyter notebook, which is the environment in which you will code, as well as (ii) with Python, the language that you will use to code in this course and with the IPython programming environment. All subsequent labs will involve coding in Python within Jupyter.

**I. Jupyter Notebook**

A notebook integrates code and its output into a single document that combines visualizations, narrative text, mathematical equations, and other rich media. This intuitive workflow promotes iterative and rapid development, making notebooks an increasingly popular choice at the heart of contemporary data science, analysis, and increasingly science at large. (from <https://www.dataquest.io/blog/jupyter-notebook-tutorial/>).

The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text. Uses include: data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning, and much more. (from [jupyter.org](https://jupyter.org))

***Installing Jupyter notebook***

While Jupyter runs code in many programming languages, **Python** is a requirement (Python 3.3 or greater) for installing the Jupyter Notebook.

We will use the [Anaconda](https://www.anaconda.com/download) distribution to install Python and Jupyter. Anaconda conveniently installs Python, the Jupyter Notebook, and other commonly used packages for scientific computing and data science.

Use the following installation steps:

1. Download [Anaconda](https://www.anaconda.com/download). Download Anaconda’s latest Python 3 version (currently Python 3.8 as of 30 Jul 2021). For Windows, choose 64-bit/32-bit version according to your laptop/desktop.
2. Install the version of Anaconda which you downloaded, following the instructions on the download page.
3. Congratulations, you have installed both Python and Jupyter Notebook.

***Running Jupyter notebook***

Start the notebook server from the command line (using Terminal on Mac/Linux, Command Prompt on Windows) by running: A screenshot of a cell phone

Description automatically generated

More details at <https://jupyter.readthedocs.io/en/latest/running.html#running>.

**II. IPython**

IPython is an interactive shell environment to execute python code in REPL (Read Eval Print Loop). It makes it more interactive by adding features like syntax highlighting, code completion etc.

A **read–eval–print loop** (**REPL**), also termed an **interactive toplevel** or **language shell**, is a simple, interactive computer programming environment that takes single user inputs (i.e., single expressions), evaluates (executes) them, and returns the result to the user; a program written in a REPL environment is executed piecewise. A REPL can become an essential part of learning a new language as it gives quick feedback to the novice.

**Preparation for the course**

This lab is not about learning the Python language. Programming in Python is discussed in the course CE/CZ 1103 Introduction to Computational Thinking & Programming. You would have already taken that course or might be taking it this semester. Therefore, you should spare some time to revise or pick up the main points in Python.

If you are new to Python, there are many resources available to learn the language. You can also get a quick snapshot of Python that is relevant to Linear Algebra [here](https://cs231n.github.io/python-numpy-tutorial/) (please ignore the discussions on Colab). You are encouraged to run the code snippets available there in your Jupyter notebook. Additional information is available [here](https://www.math.ubc.ca/~pwalls/math-python/linear-algebra/linear-algebra-scipy/) and [here](https://s3.amazonaws.com/assets.datacamp.com/blog_assets/Python_SciPy_Cheat_Sheet_Linear_Algebra.pdf).