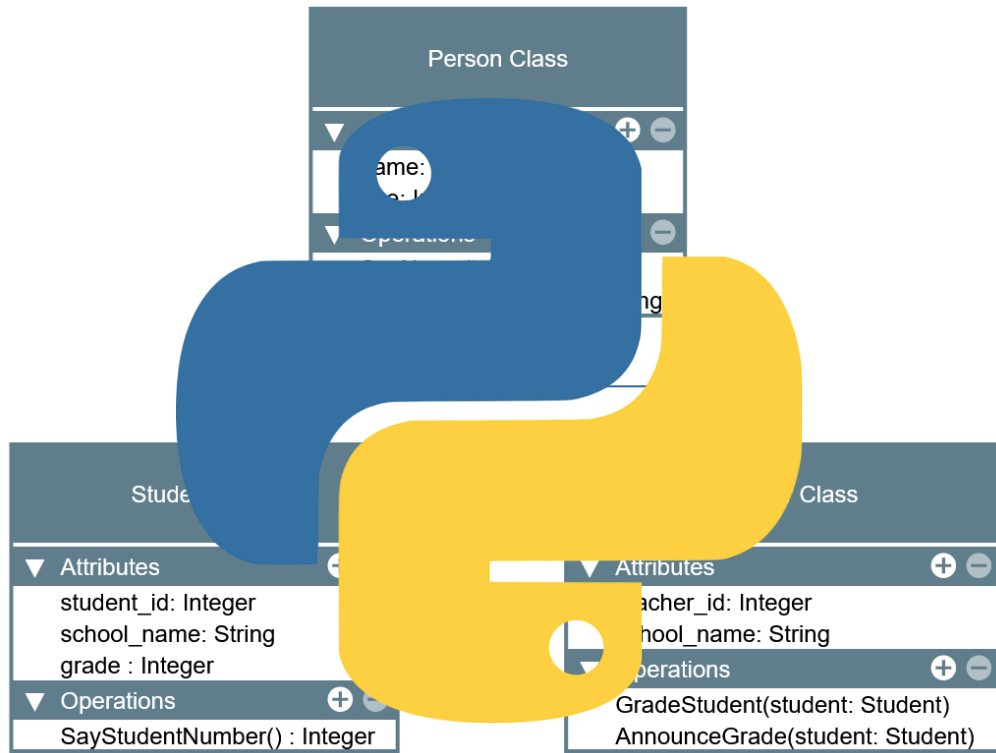**TECHNOLOGICAL INSTITUTE OF THE PHILIPPINES**
COMPUTER ENGINEERING DEPARTMENT
Quezon City

LABORATORY MANUAL

Object-Oriented Programming (CPE 009)

Prepared by:

Engr. Royce Chua

Activity No. 1
Working with the Python Environment

Course Code: CPE009	**Program:** BSCPE
Course Title: Object-Oriented Programming	**Date Performed:**
Section:	**Date Submitted:**
Name:	**Instructor:**

1. Objective(s):

This activity aims to familiarize students in the various ways how a Python program can be written and executed on the terminal and through an Integrated Development Environment (IDE).

2. Intended Learning Outcomes (ILOs):

The students should be able to:

2.1 Install and use Anaconda Python, and an Integrated Development Environment (IDE) on their machines.

2.2 Execute Python Scripts through the terminal, the Spyder IDE, and the Jupyter Notebook.

3. Discussion:

The **Python** programming language is an interpreted, high-level general-purpose programming language developed and released by Guido Van Rossum in 1991. The Python programming language is used in different kinds of applications some of which are:

- Software and Web development
- Data Science and Data Analysis (Scientific Computing)
- Artificial Intelligence (AI)
- Systems and Network Administration
- Scripting and Automation
- Electronics / Embedded Systems / Robotics

The Python programming language is easy to learn with its English like syntax and at the same time powerful performance in implementing complex logic in readable short lines of code. Being an interpreted language means that Python executes code directly at runtime unlike compiled languages that needs to be compiled before being able to run. Because of this Python offers a feature called **Python Shell** which can immediately interpret Python code. It is called directly by finding and running python.exe either through the Windows GUI or through the terminal (command prompt). This can be used by developers to quickly implement or test a program or algorithm without having to write a program, save, then execute.


```
Python 3.6 (64-bit)

Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 17:00:18) [MSC v.1900 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> print("Welcome to Python")
Welcome to Python
>>> 1+1
2
>>>
```

Python has two versions: The first is Python 2.x which is the first version of Python to be released that is about to reach its end of life date or EOL in 2020 meaning that it will no longer be maintained by its creators and problems/errors encountered while

using a library/module written in Python 2 will most likely remain broken and unfixed. The latest version is Python 3.x (Python 3.6, 3.7, 3.8) which provides support for newer libraries/modules as well as improvements to backend processes that make the newer Python program's performance better. Therefore, it is recommended to always download and use Python 3.x for any programs that will be developed at the start of this laboratory.

Python Distributions

Python can be downloaded as a stand-alone program from their website (https://www.python.org) or through various distributions made by the developers in the Python community.

Two common distributions that are used are: **Anaconda** and **WinPython**.
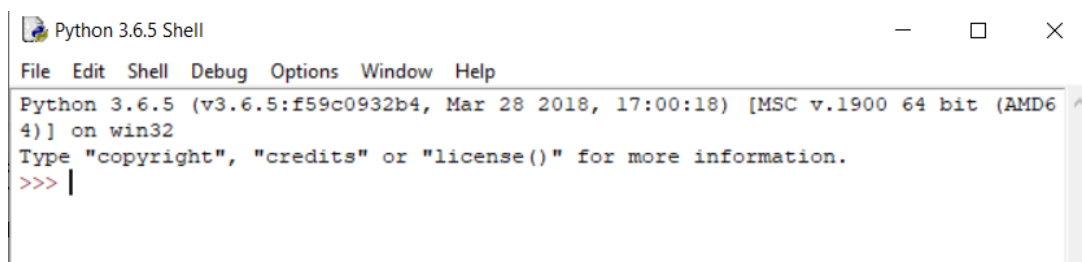
Anaconda is a free and open-source Python and R distribution for Windows and Linux that comes with its own installation and package management tools (https://www.anaconda.com/). It comes pre-packaged with programs such as the Anaconda Navigator, Jupyter Notebook, Spyder and the common libraries/modules used for Data Science and Scientific Computing in general as these libraries are many and installing these modules one-by-one can be a tiring and time-consuming task. The laboratory computers are already installed with Anaconda for use in engineering data analysis, data science and programming courses. Anaconda is the Python distribution that will be installed and used throughout the laboratory activities.

WinPython is also a free open-source distribution of the Python programming language for Windows 7/8/10 and scientific and educational usage (https://winpython.github.io/). While Anaconda needs to be installed before it can be used. WinPython only extracts the Python interpreter and its packages (additional programs and modules) on to a folder making it a Portable Python distribution which has the same modules as Anaconda for data science. Being portable means it is not needed to be installed for use. However, unlike Anaconda, WinPython cannot be directly integrated with the Windows terminal making it slightly more challenging to use than Anaconda.
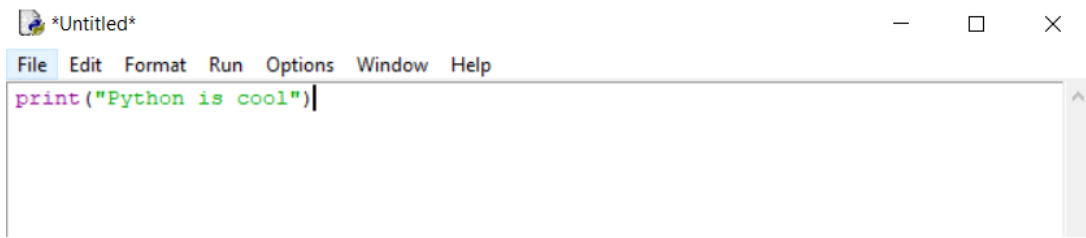
Integrated Development Environments (IDEs) for Python

Integrated Development Environments are simply programs optimized for the developers to help them write, run, and debug programs more efficiently through integrated tools in the program such as the Syntax Highlighter which colors the keywords in a programming language. Python code can be written on the Notepad and the Command Prompt if no IDEs are present on the machine but the different distributions of Python do come with their own IDE.

The base Python distribution comes with the **Python IDLE** which stands for Python Integrated Development and Learning Environment. This is a basic editor for developing Python programs as well as being able to run them through the IDLE. The Python IDLE also includes shell window which allows developers to use the Python shell.

The IDLE shell window

```
*Untitled*                                        —   □   ×
File  Edit  Format  Run  Options  Window  Help
print("Python is cool")
```

The IDLE editor window

The Anaconda distribution of Python comes pre-packaged with two editors that developers, scientists, and engineers can use: The **Spyder IDE** and the **Jupyter Notebook**.

The **Spyder IDE** is a powerful development environment for writing Python code that provides support for data science and scientific computation related tasks through various tools integrated with the IDE.

The **Jupyter Notebook** is not an IDE but is a web-based interactive computational environment for creating Python programs for data science related tasks. Although it is most commonly used for engineering and science-related work, it is also used as a platform for education and teaching programming. Jupyter notebooks can be opened and performed also using Google's Collaboratory from your Google Drive.

These two editors will be explored further in the laboratory activity.

4. Materials and Equipment:

Desktop Computer
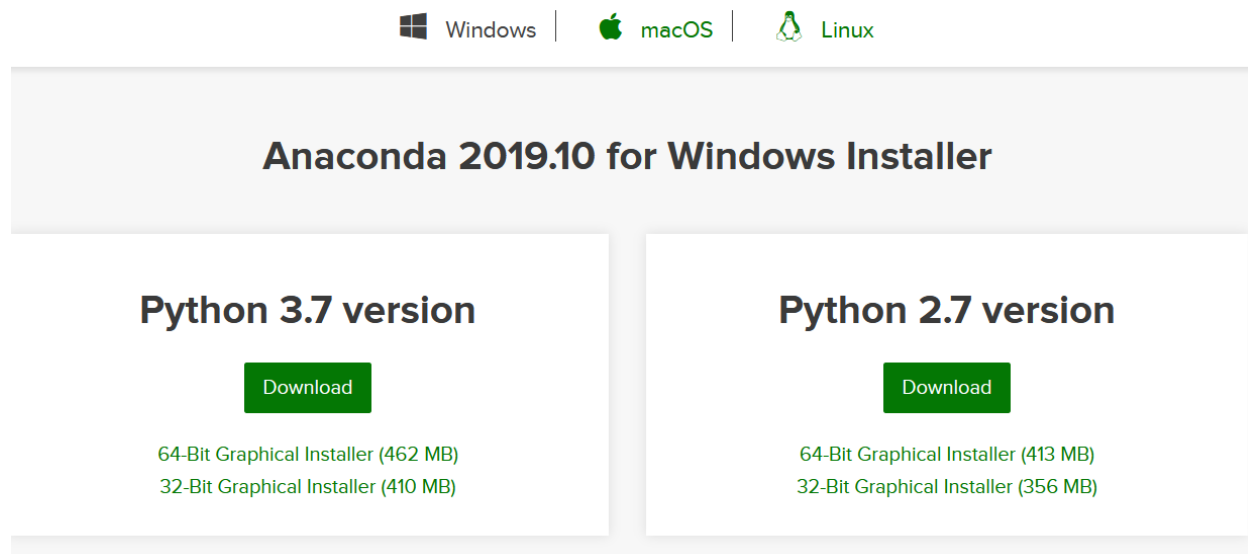Windows Operating System
Internet Connection

5. Procedure:

Downloading and Installing Anaconda on Windows 7/8/10

1. Go to the Anaconda download website through this link: https://www.anaconda.com/distribution/
   The page that should load should look similar to the one below.
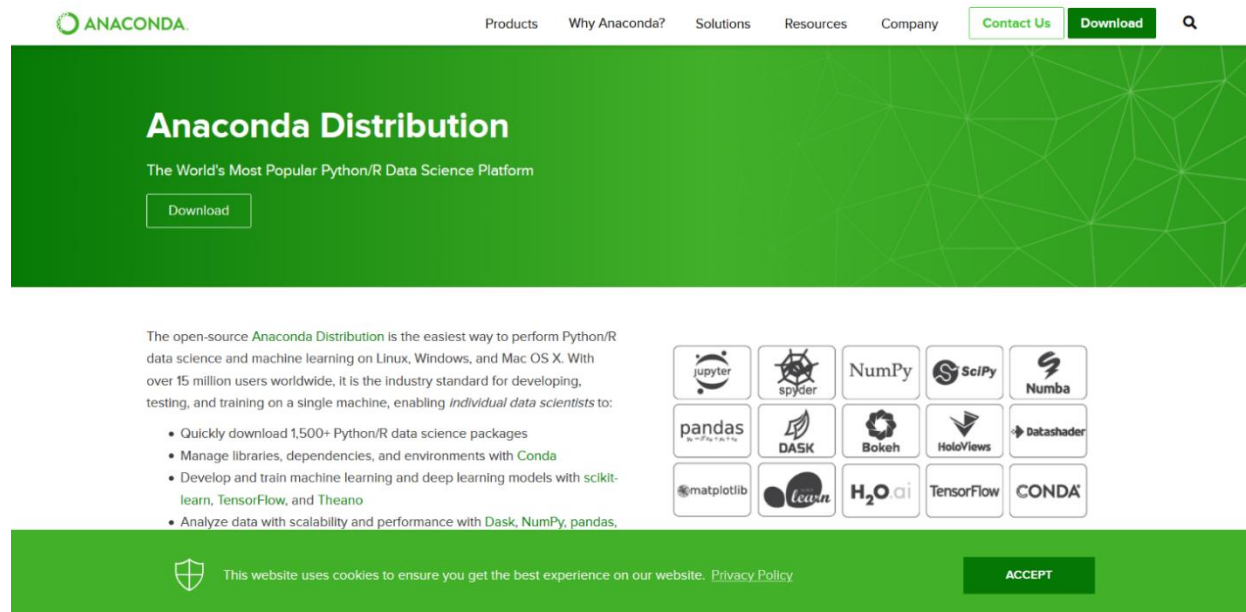
2. Click download. Select the appropriate bit-version of your machine under Python 3.7.

3. Save the file. Then run the installer once it has finished downloading.
4. Follow through the installation and do not change any of the default settings. In the installation type it selecting Just Me maybe the only option for your machine if you do not have Administrator Privileges.
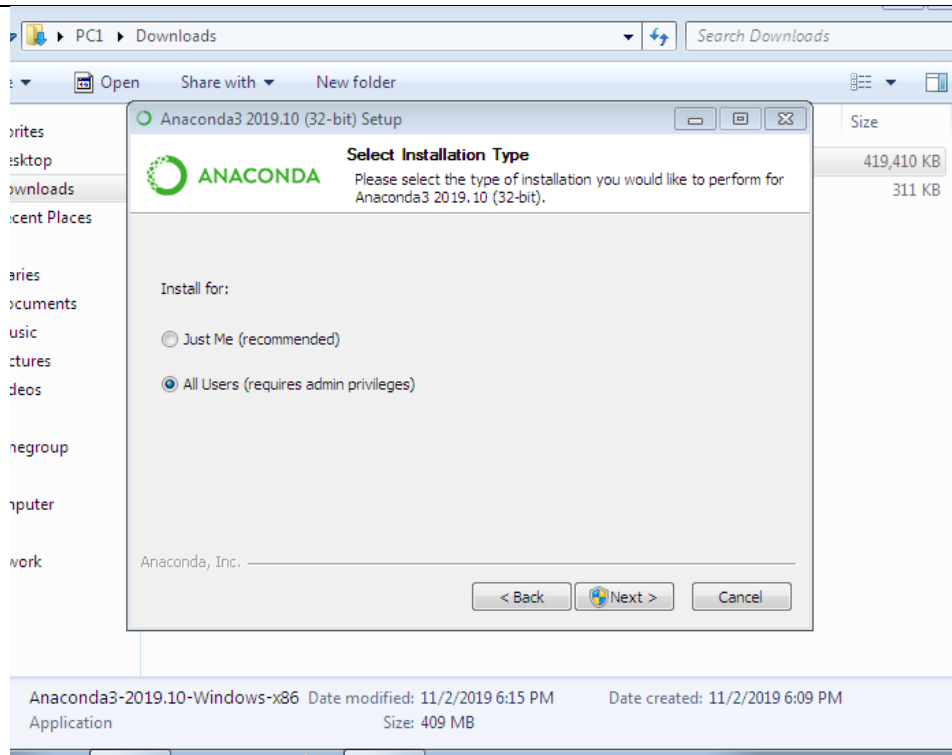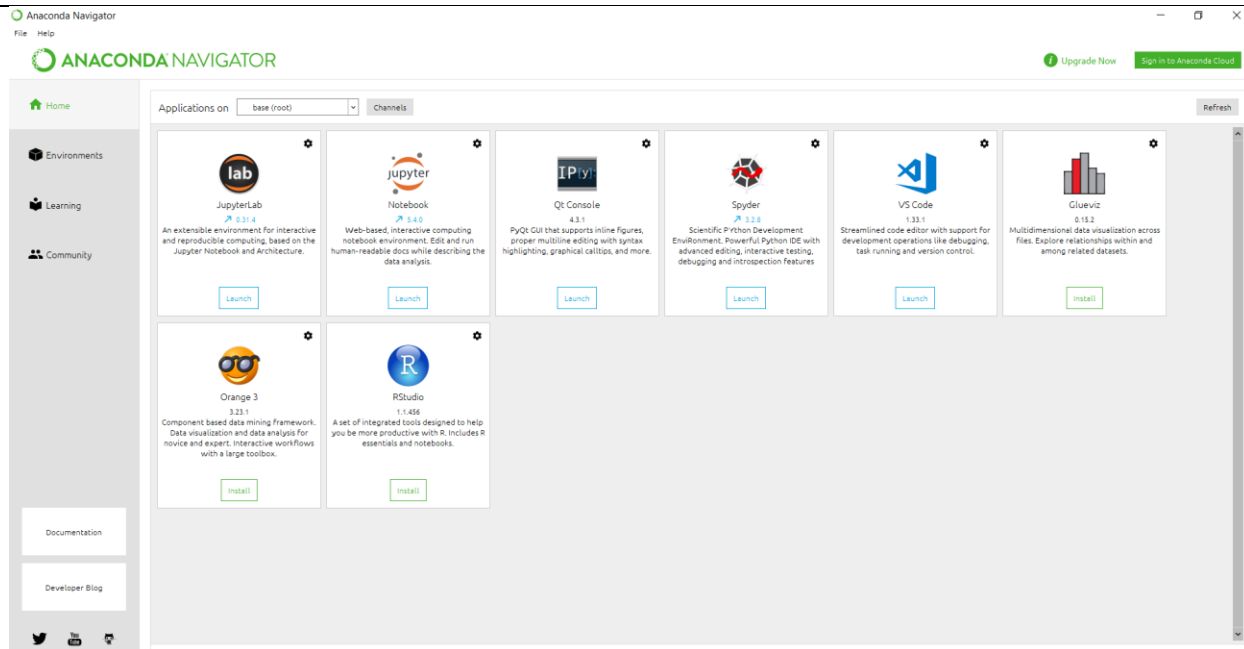
5. Click next through the different steps. In the last step do not modify anything and just click Install. Wait for the installation to finish.
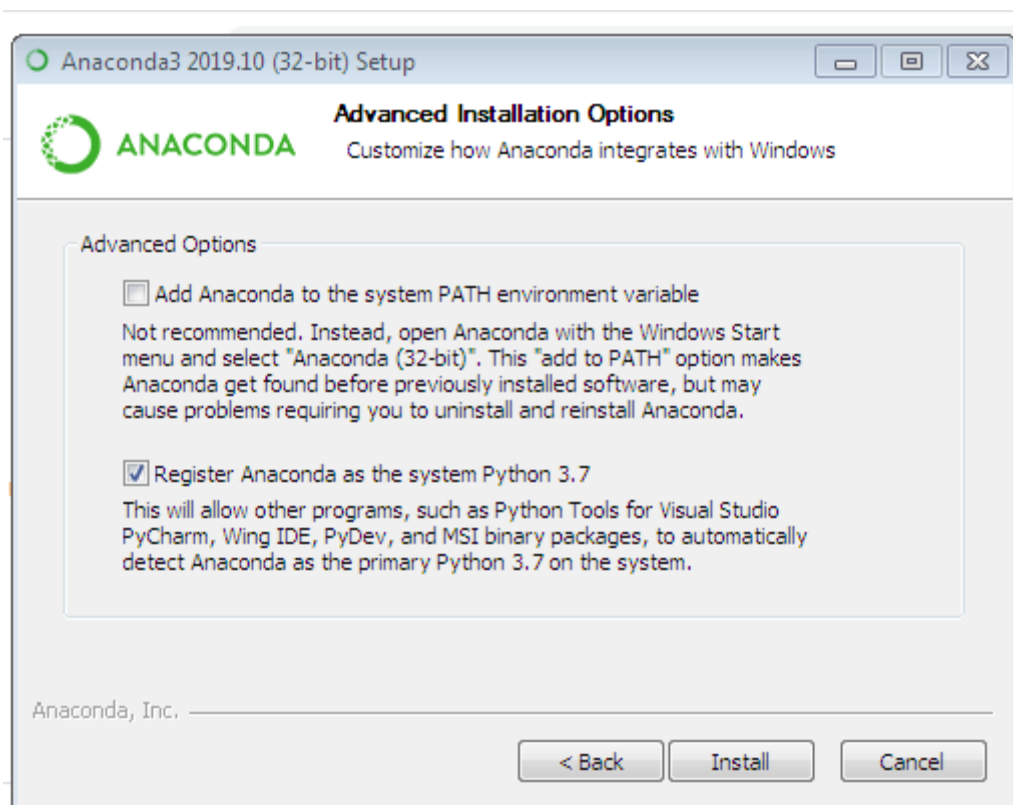
6. After the installation, press the start menu or the start button and type "Anaconda Navigator".
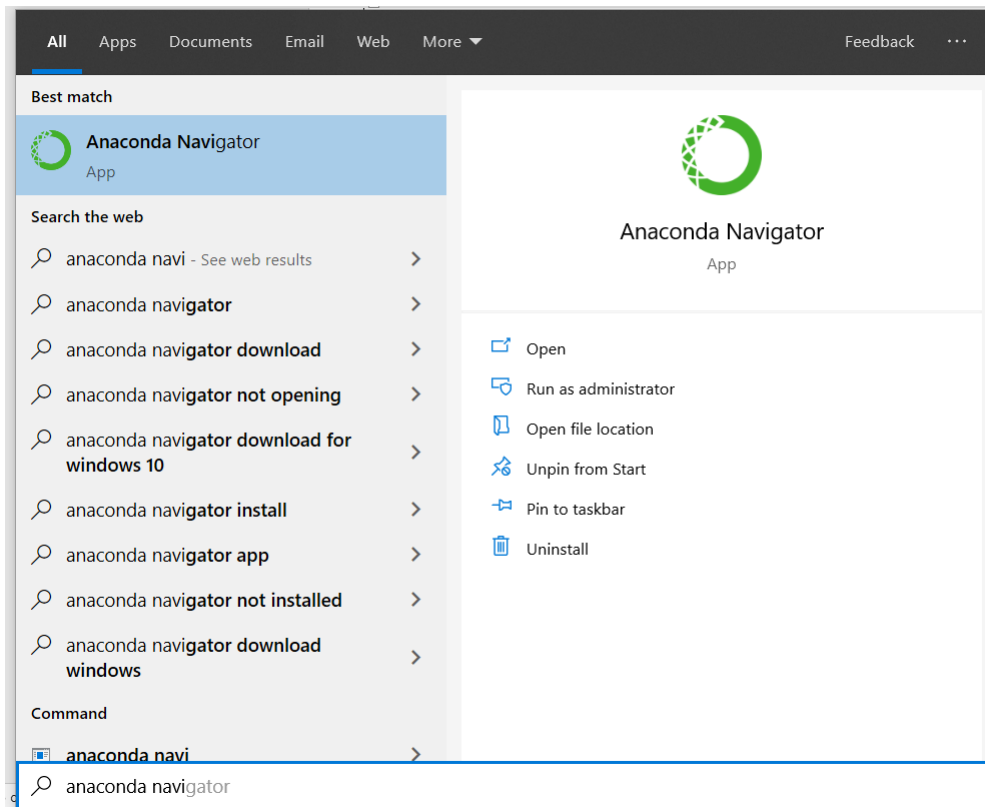
7. The Anaconda Navigator should look similar to the image below. This indicates a successful installation of Anaconda.

Running Python Programs through the Terminal and IDE

The next set of activities will show the different ways on how to execute/run Python programs. Study and observe the output of the procedures carefully and what each section shows on how to execute Python programs.
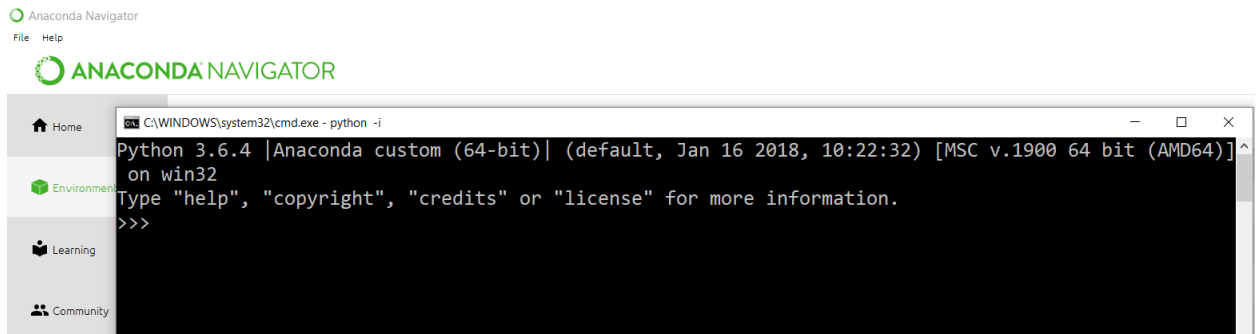
Using the Python Shell

There may be some instances where you may find that directly running the python script through the interpreter maybe more convenient than waiting for the IDE or Jupyter to load. In these instances, you may just want to test a small code logic or a small program. The Python Shell maybe more convenient to use in this case.
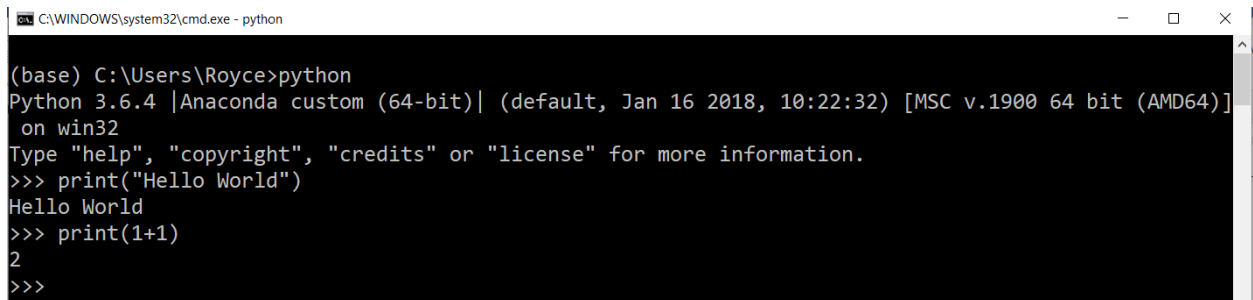
1. In the Anaconda Navigator, click Environments at the left side under the Home. Then click the icon at the right of base. Select *Open with Python*.
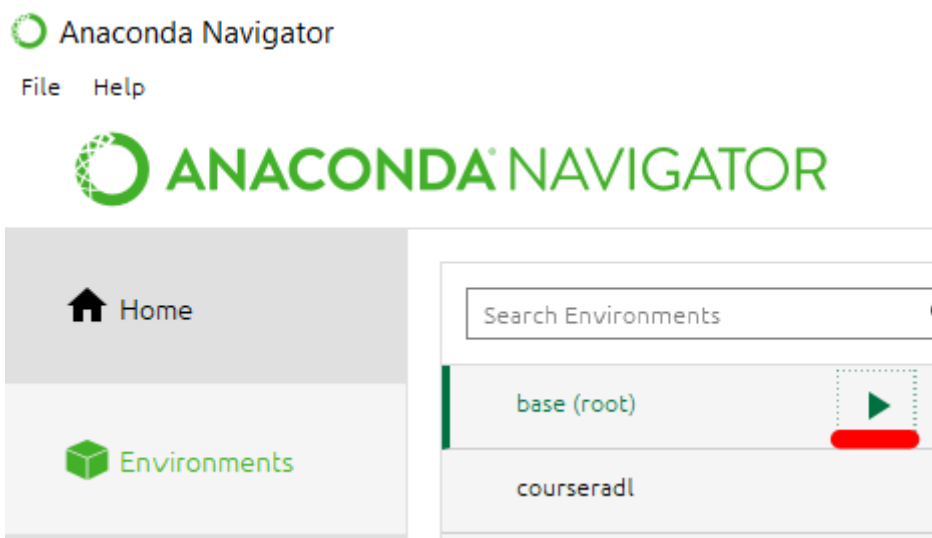
This is the window that should appear after clicking Open with Python.

2. Type the command print("Hello World") and/or print(1+1) on the shell. Observe how the shell interprets the code. The **print()** function is the main built-in function in Python used to display data on to the screen/terminal.

Using the terminal to run Python Programs
Besides using the Python shell, developers may also want to run the Python program directly through the command prompt.

1. In the Anaconda Navigator, click Environments at the left side under the Home. Then click the icon at the right of base. Select Open Terminal.
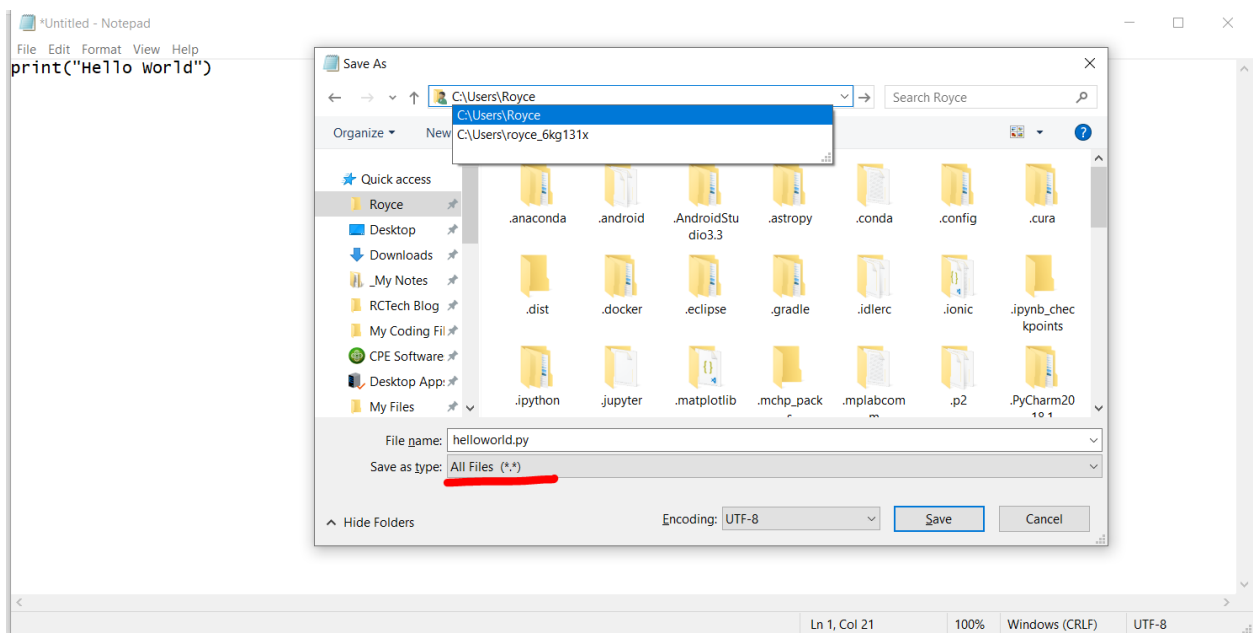
C:\WINDOWS\system32\cmd.exe

(base) C:\Users\Royce>
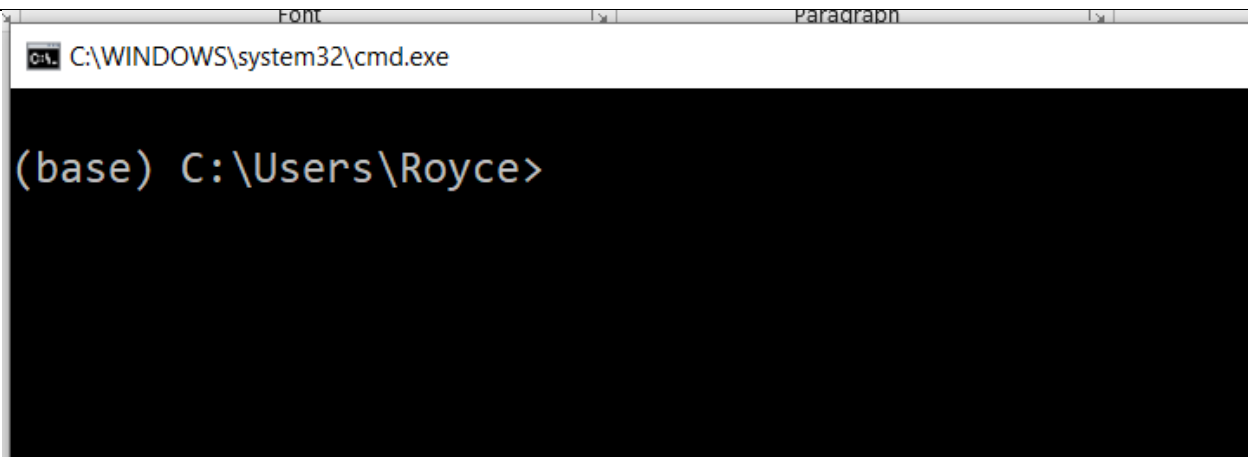
2. You now have an option to directly execute the program through the command prompt by running the command
python filename.py

Note: We will need to create a Python program first using the notepad.

3. Create a Python program by opening notepad and typing the python code.
print("Hello World")

Then Save the program in the directory you see in the command prompt by pressing File->Save or through the shortcut Ctrl+S. Make sure to change the file type to All Files (*) and set your filename to helloworld.py

4. Go back to the terminal and type the command

python helloworld.py

5. The output should look similar to the image below. You have successfully run your Python program through the terminal.
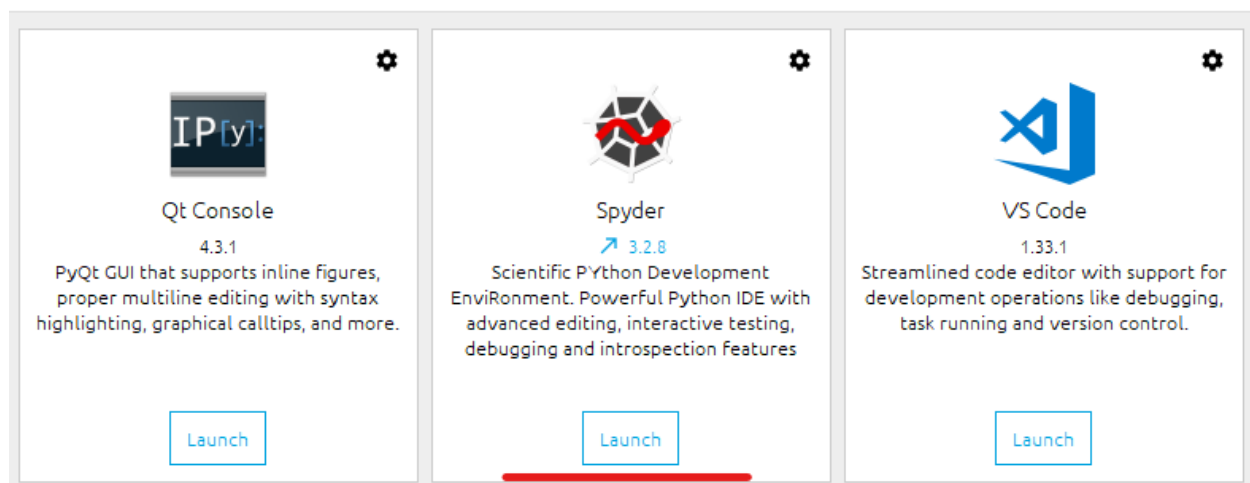
C:\WINDOWS\system32\cmd.exe

```
(base) C:\Users\Royce>python helloworld.py
Hello World

(base) C:\Users\Royce>_
```

Do not close this terminal until you have finished the Supplementary Activity.

**Using the Spyder IDE for running Python Programs**
1. Go to the Home Anaconda Navigator, click the Launch Button under the Spyder section of the window. Launching Spyder should be visible at the bottom left of the Navigator.

IP[y]:	Spyder icon	VS Code icon
Qt Console	**Spyder**	**VS Code**
4.3.1	⬈ 3.2.8	1.33.1
PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.	Scientific PYthon Development EnviRonment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features	Streamlined code editor with support for development operations like debugging, task running and version control.
Launch	Launch	Launch

2. Wait for the Spyder IDE to load. The IDE should look similar to the image below.
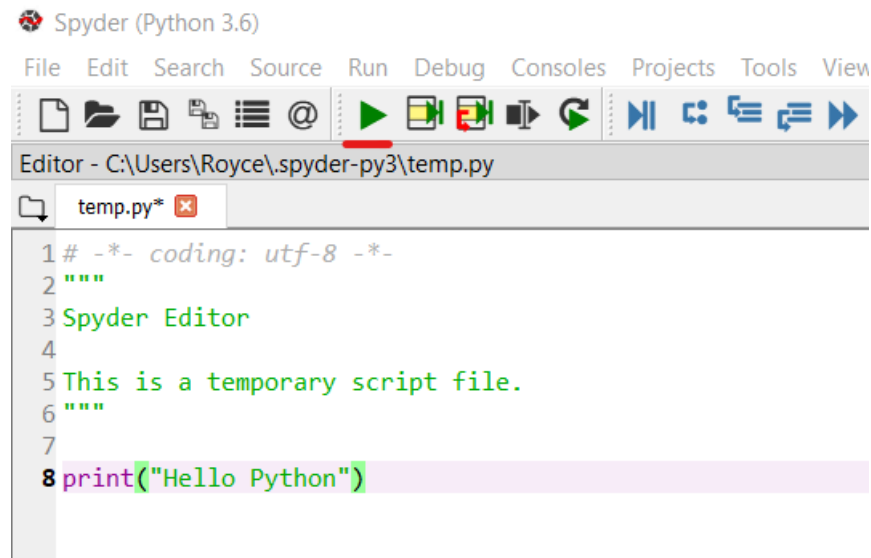   Note: It may take a couple of minutes depending on the machine for the initial setup.
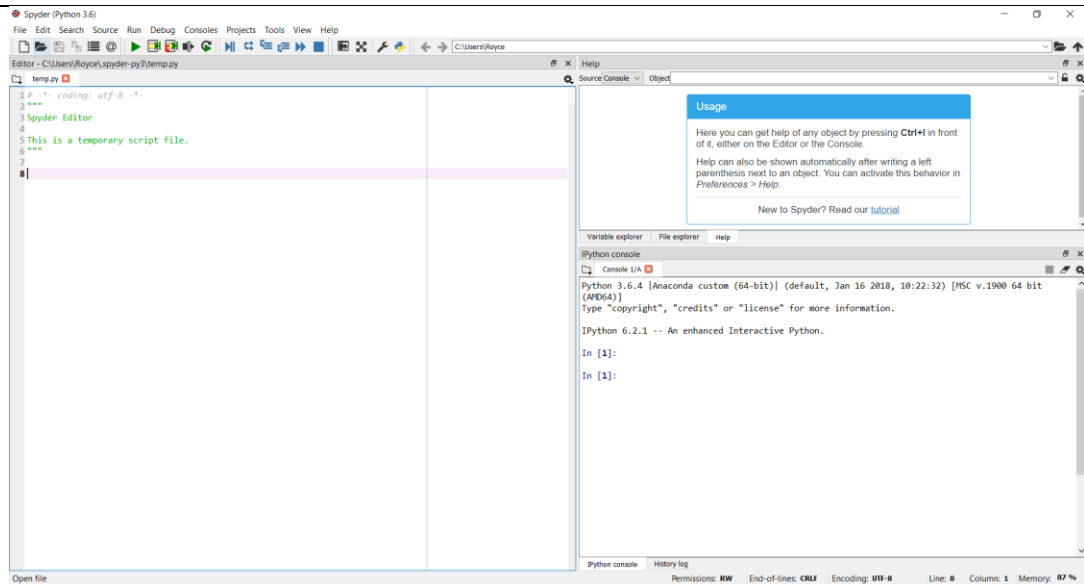
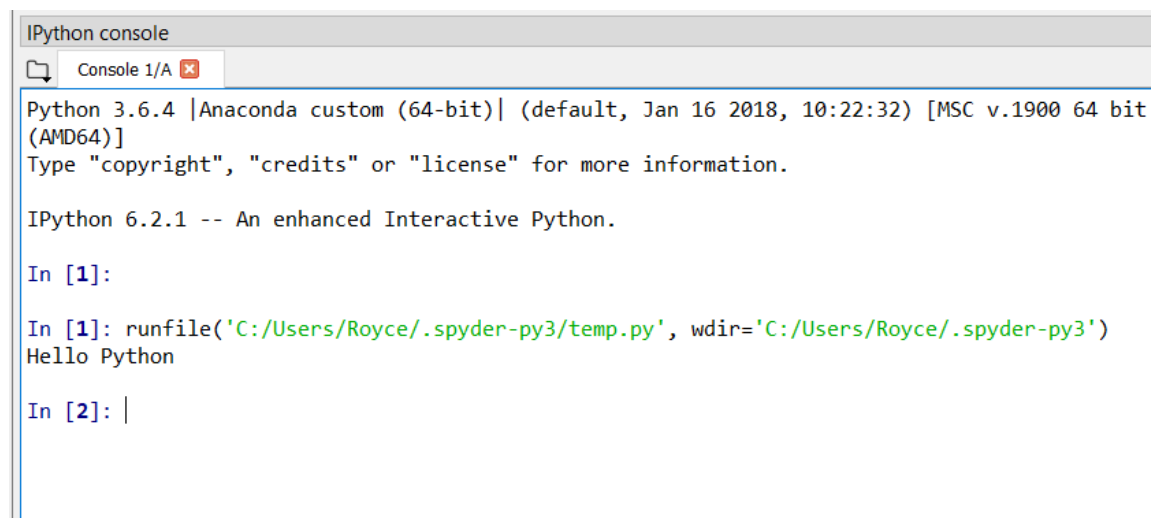3. Type the following command on the left side.
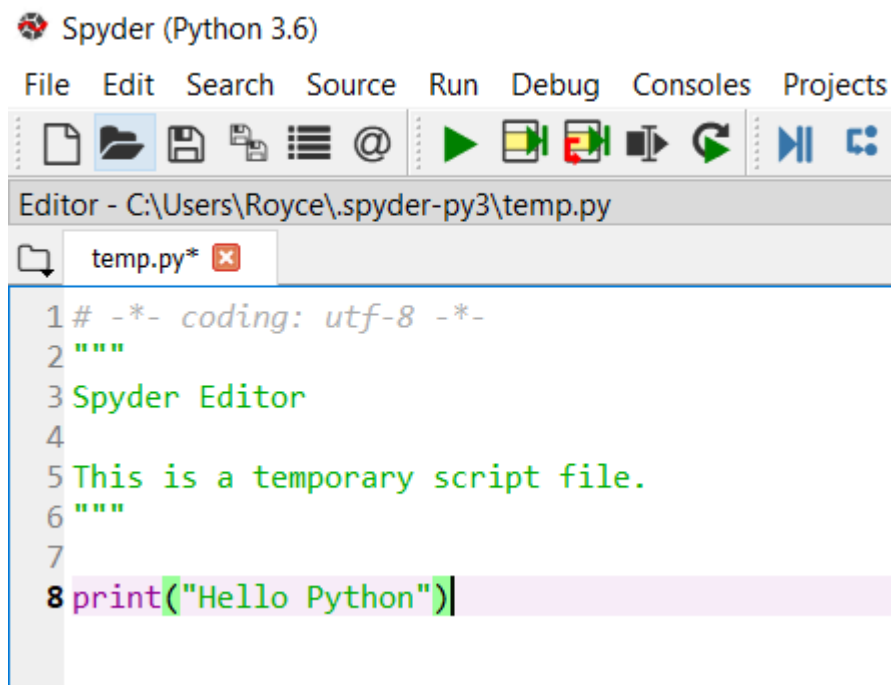
Note: The text inside """ """ can be disregarded. You can either remove it or leave it as is as it represents a multi-line comment which Python does not interpret.

4. Click Run or press F5.

```
Spyder (Python 3.6)
File  Edit  Search  Source  Run  Debug  Consoles  Projects  Tools  View

Editor - C:\Users\Royce\.spyder-py3\temp.py

temp.py*

1 # -*- coding: utf-8 -*-
2 """
3 Spyder Editor
4
5 This is a temporary script file.
6 """
7
8 print("Hello Python")
```

5. The output should be displayed at the right side of the Spyder IDE in the IPython Console.

```
IPython console
Console 1/A

Python 3.6.4 |Anaconda custom (64-bit)| (default, Jan 16 2018, 10:22:32) [MSC v.1900 64 bit
(AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 6.2.1 -- An enhanced Interactive Python.

In [1]:

In [1]: runfile('C:/Users/Royce/.spyder-py3/temp.py', wdir='C:/Users/Royce/.spyder-py3')
Hello Python

In [2]:
```

You have now successfully run your first Python program in the Spyder IDE. The next section will show how to run a Python Program using the Jupyter Notebook. Do not close the Spyder IDE until you have finished the Supplementary Activity.
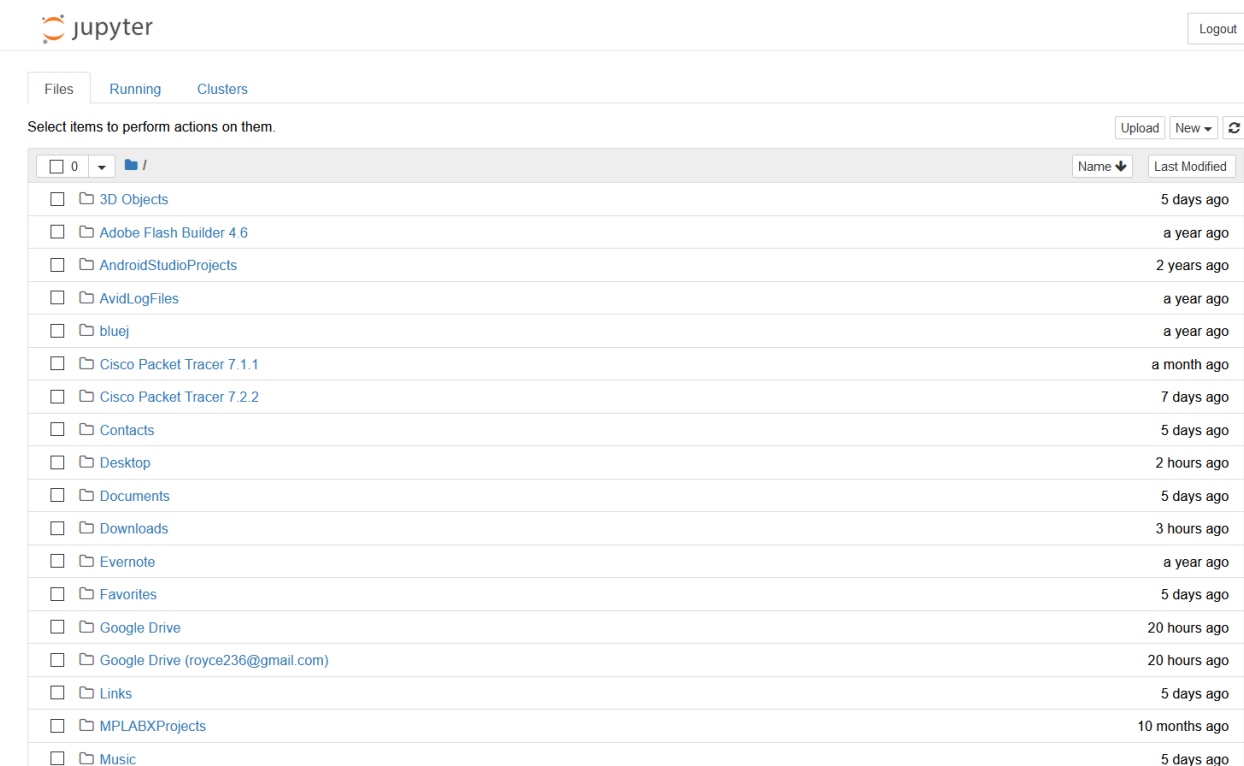
Using the Jupyter Notebook to run Python Programs

1. Navigate back to the Anaconda Navigator and launch the Jupyter Notebook. A new tab should appear in your browser with the URL: localhost:8888/tree
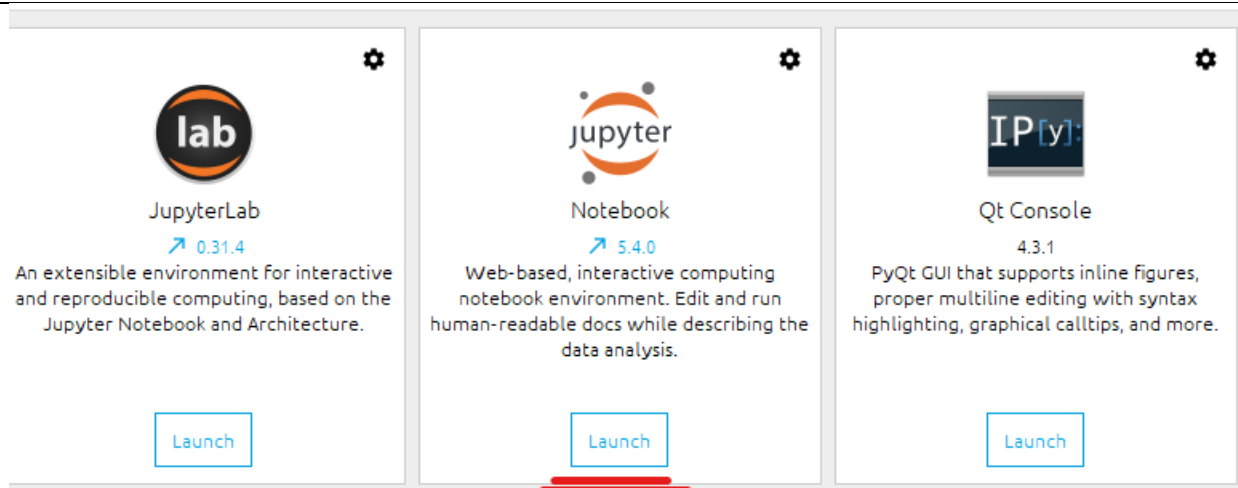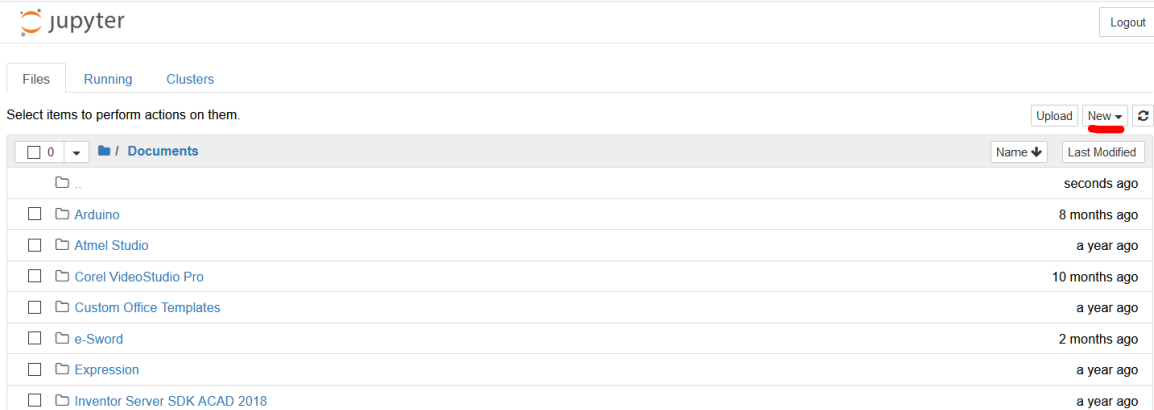
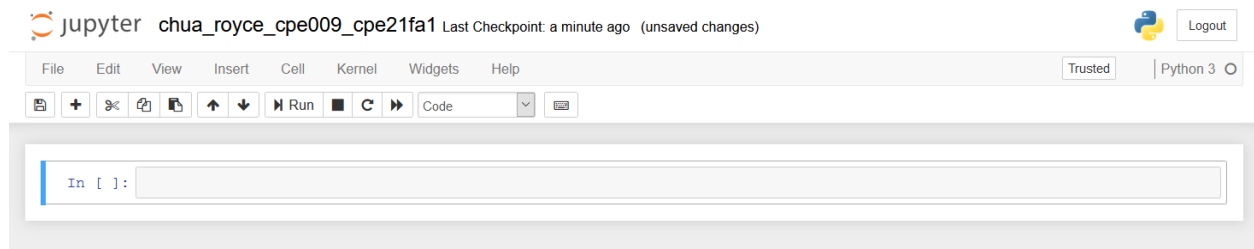**Note:** Jupyter Notebook is different from Jupyter Lab.

2. This should be the window that appears after Jupyter Notebook has been launched. Click **Documents**.
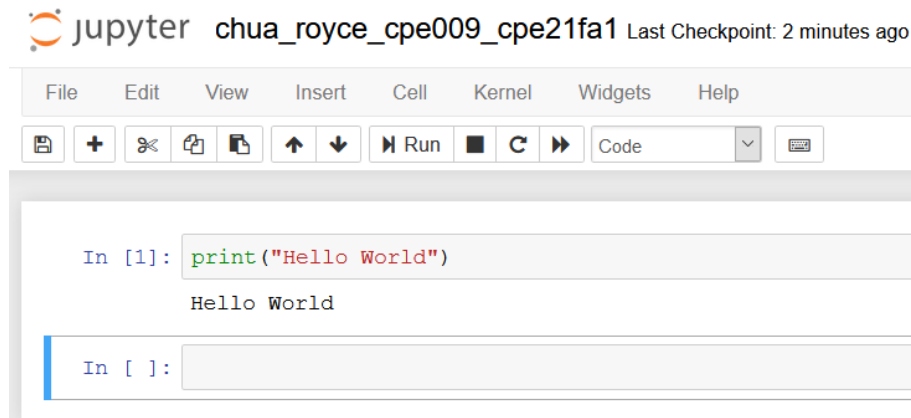

3. Create a new file by clicking New then Python 3. A new tab should appear.

4. Click on the Untitled at the Top portion of the new window then rename it to lastname_firstname_cpe009_section similar to the image below.

5. Click the text box at the right side of the In [ ]: and type print("Hello World") then click Run.

Do not close the Jupyter Notebook until you have finished the Supplementary Activity.
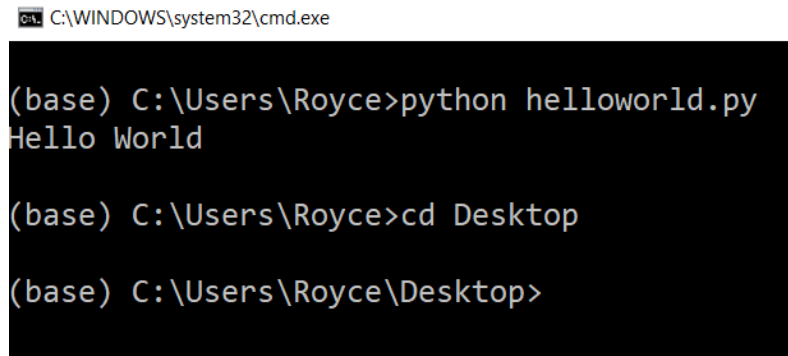
6. Supplementary Activity:

Tasks

1. Using the notepad create a new Python program named lastname_firstname_program1.py with the code

print("Python is everywhere")

and save it on your Desktop folder. Execute the program using the anaconda prompt and terminal.

Hint: Refer to the cd command in the Windows terminal.

C:\WINDOWS\system32\cmd.exe

```
(base) C:\Users\Royce>python helloworld.py
Hello World

(base) C:\Users\Royce>cd Desktop

(base) C:\Users\Royce\Desktop>
```

2. Using the Spyder IDE, create a new program with the code

> print("Python is everywhere!")

Run the program using Spyder IDE.

3. Using the Jupyter Notebook, create a new file/notebook with the code:

> print("Python is everywhere!")

Run the program using Jupyter.

Questions:
1. **How does Python differ with the previous programming language you have used in terms of syntax and execution?**
   _____
   _____

2. **In the Open with Terminal option in Anaconda, what happens when you type the command *python*? Based from the discussion, what do you think is the advantage of this feature?**
   _____
   _____

3. **In the same window in number 2, type the command *printf("That's not Python"). What is the output? Did the command prompt stop working after the output?**
   _____
   _____

4. **What happens if you type python file1.py without creating such a file? Why do you think that was the output?**

   _____

   _____

5. **In the Jupyter Notebook, after the first code has been run a new block or cell has been created try putting the code** _print("Your second block of code!")_ **and then run the notebook. What do you think is the difference between the Jupyter Notebook and Spyder IDE in terms of making Python programs?**

   _____

   _____

7. Conclusion:

8. Assessment Rubric: