**Python Workshop – November 2023**

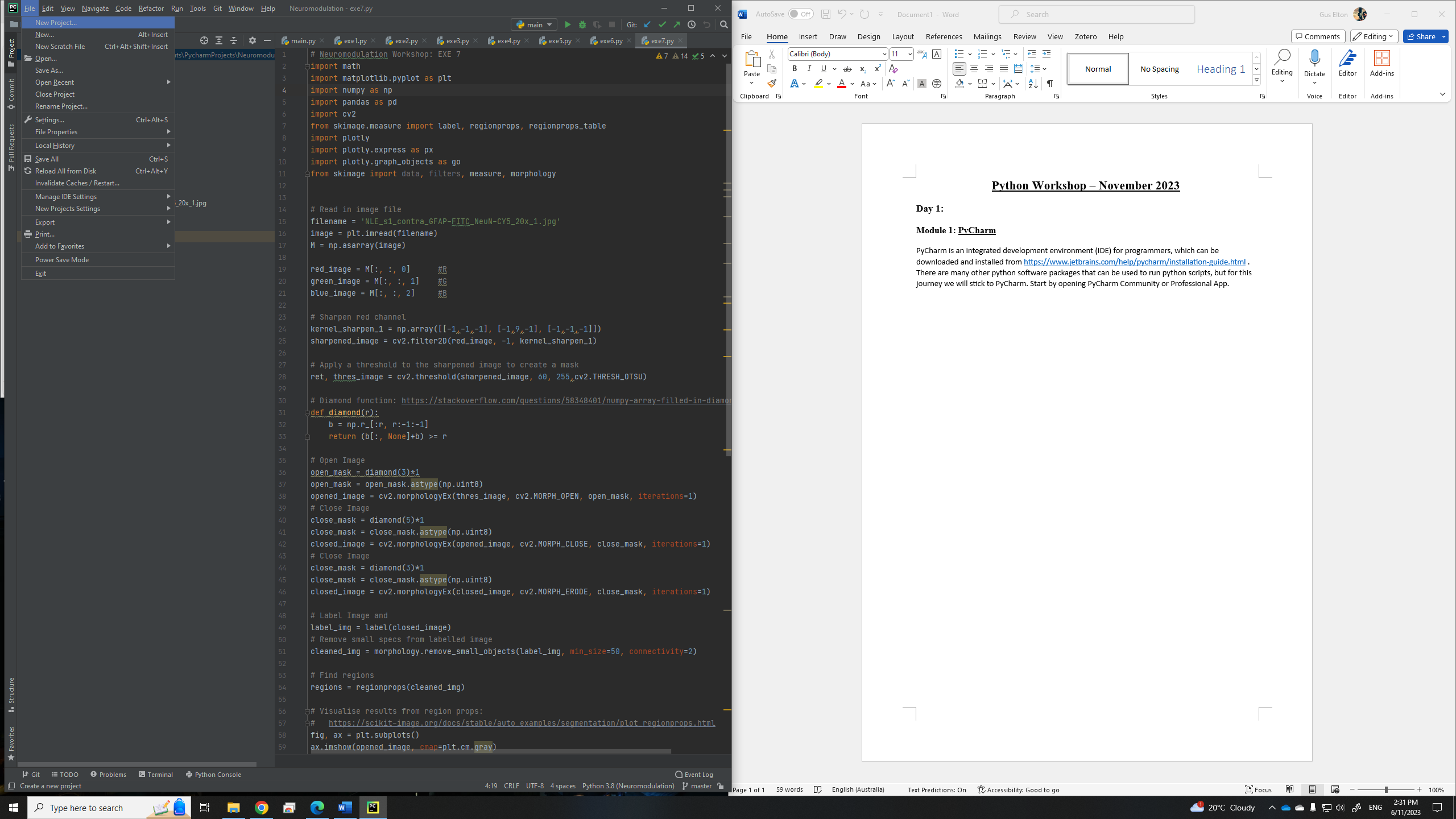
**Introduction**

Welcome to the coding section of the Neuromodulation Workshop! In two separated sessions we hope to bring you into the programming world (if you are not already familiar). We aim to step through some simple coding exercises which will lead into some traditional image processing techniques. The goal is to try and use Python (a programming language) to identify and count the number of regions of interest (ROIs) within an image obtained on a microscope so that you do not have to manually count them yourselves.

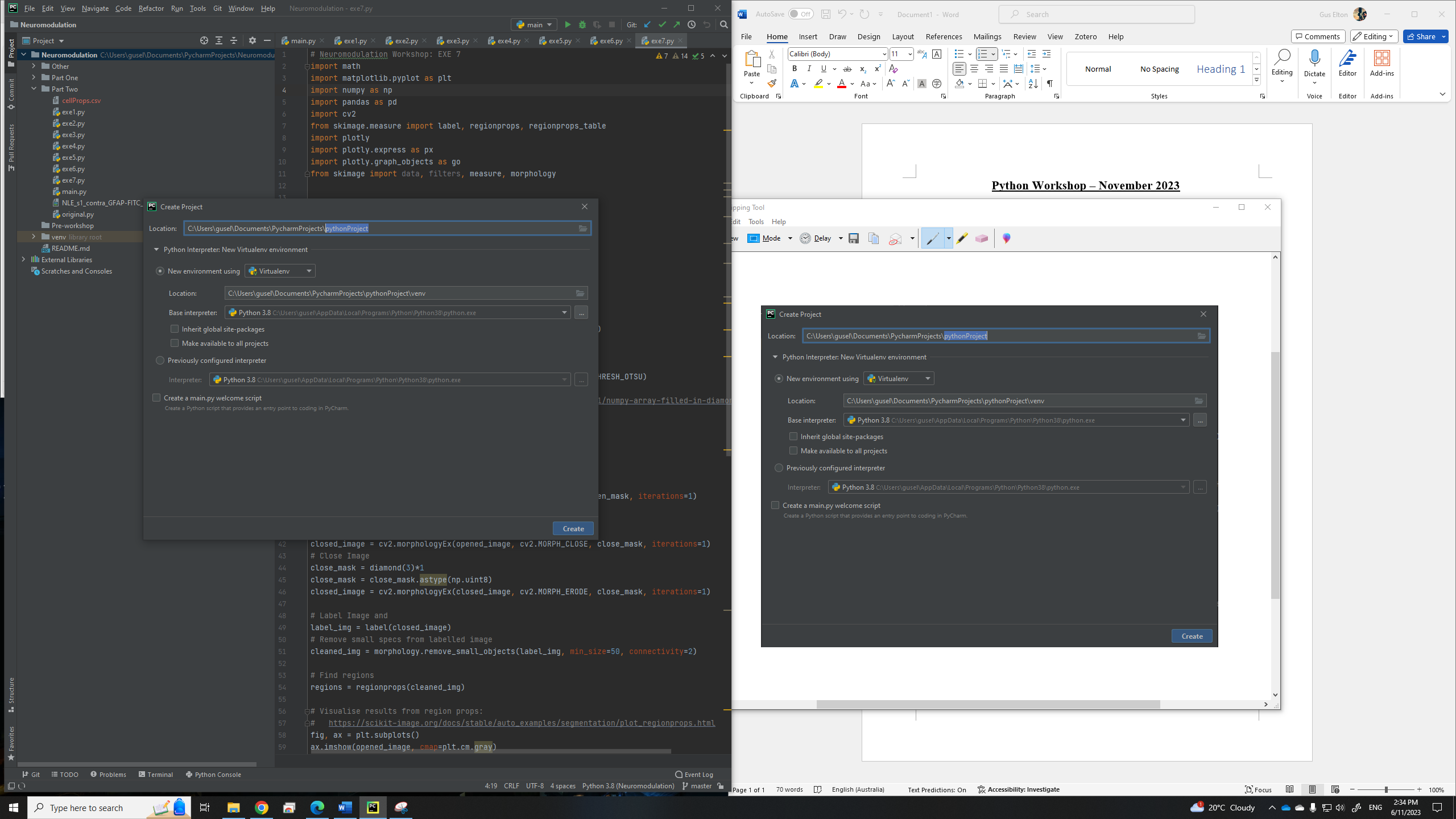
**Module 1: PyCharm**

PyCharm is an integrated development environment (IDE) for programmers, which can be downloaded and installed from <https://www.jetbrains.com/help/pycharm/installation-guide.html> . There are many other python software packages that can be used to run python scripts, but for this journey we will stick to PyCharm. Start by opening PyCharm Community or Professional App. Then

1. Create a new project: File -> New Project…



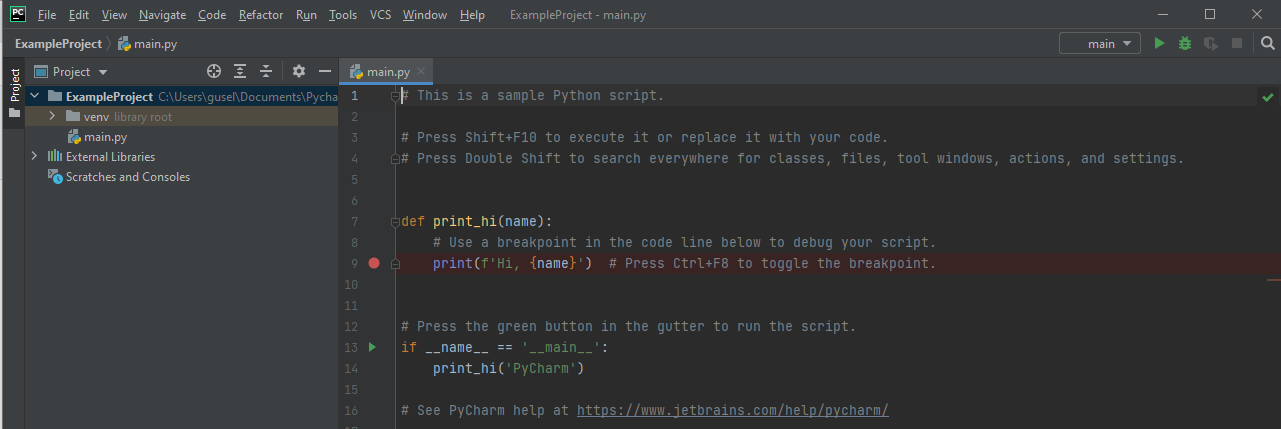
1. Rename Python Project by typing the name where the highlighted font is below.



1. Tick the ‘create a main.py welcome script’ and select create in the bottom right-hand corner of the dialogue box. Make sure you also select Python 3.x. as your base interpreter.

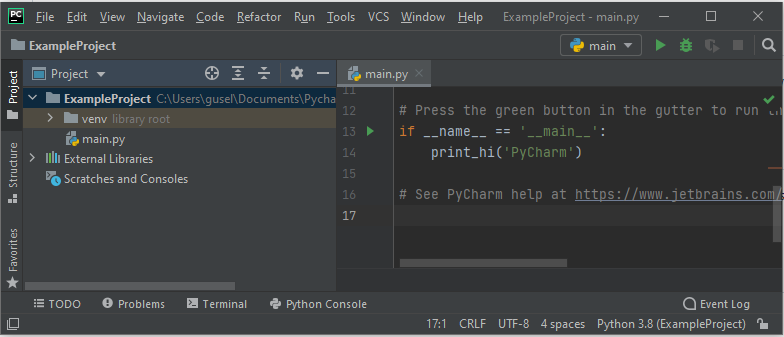


A main.py script should now appear in front of you with some example code as follows. Try running the script by clicking the play button.



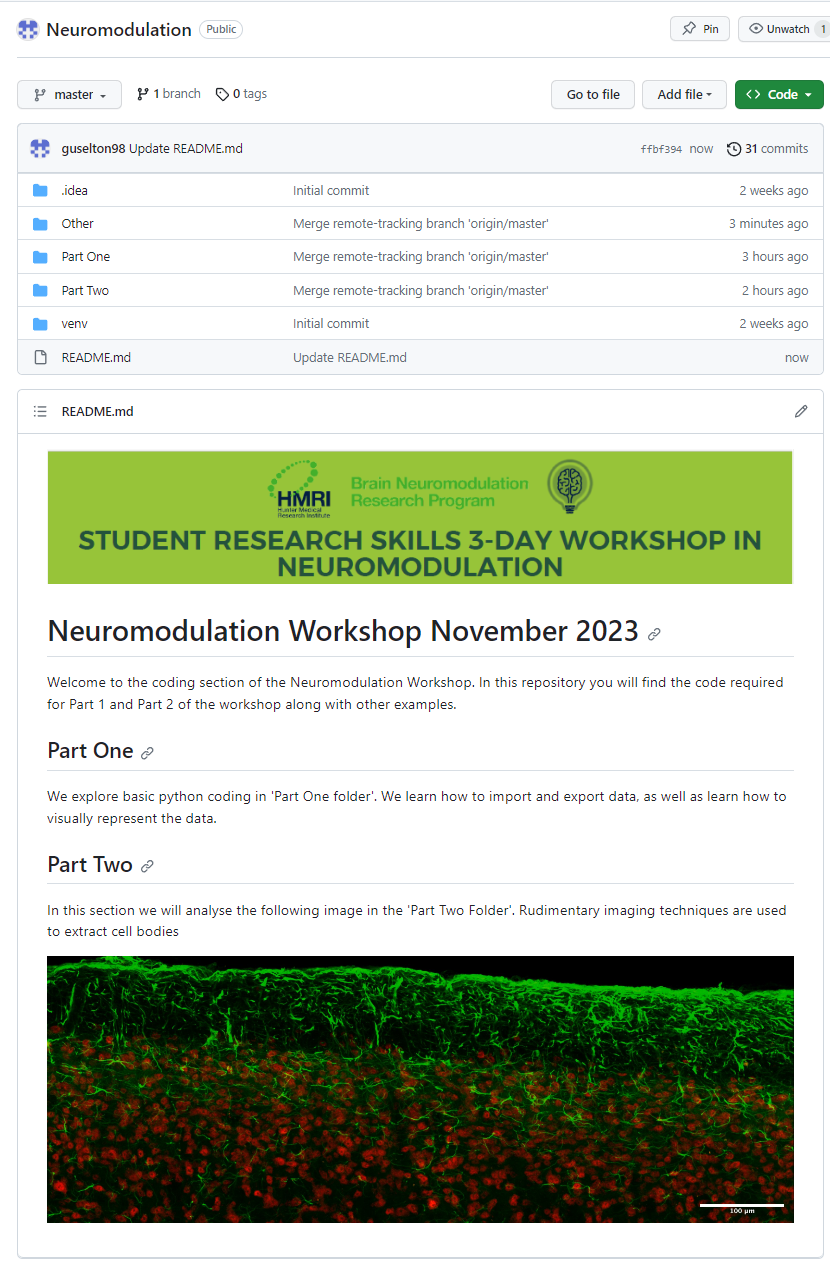
Notice that there are different Tabs (see figure below):

* Script/editor tab: where you can edit your code.
* Project tab lists the current project/working directory and files.
* Problems: when you run your code problems/warnings will appear here.
* Terminal/console: you can run commands or code directly from the computer terminal or python console (we will not be doing this)
* TODO tab: you can create items for you to complete which will be listed here.



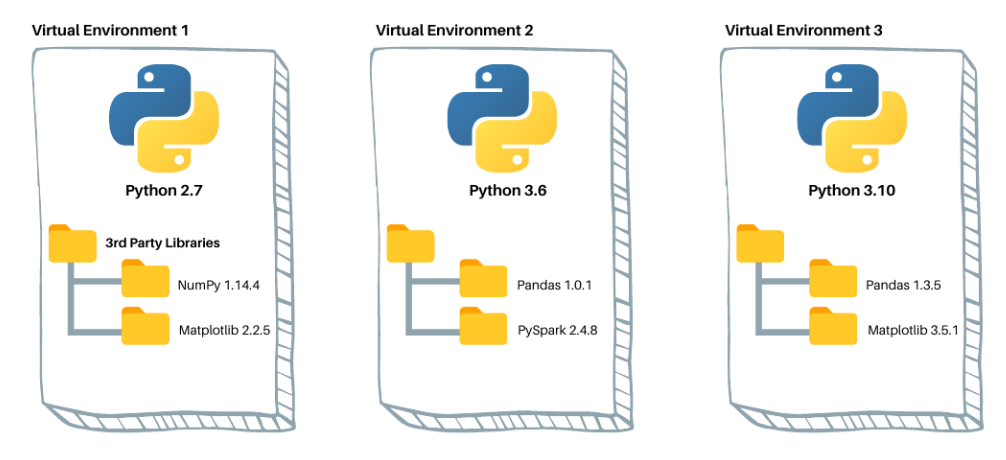
**Module 2: GitHub Repository**

In the coding sessions we will be following the examples outlined in the GitHub repository: <https://github.com/guselton98/Neuromodulation> .



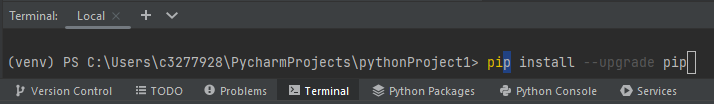
**Module 3: Setting an environment: using pip install**

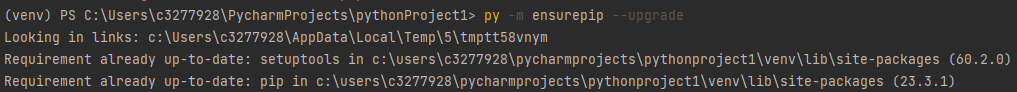
The virtual environment that you have *magically* created contains the python compiler and some pre-installed python library packages.



In this workshop we will also require a few more libraries which allows us to use tools already available. To install these packages, we use the *pip install*, and we need to make sure that the pip installer is upgraded:

1. Go to the terminal tab
2. Enter the command: pip install –upgrade pip



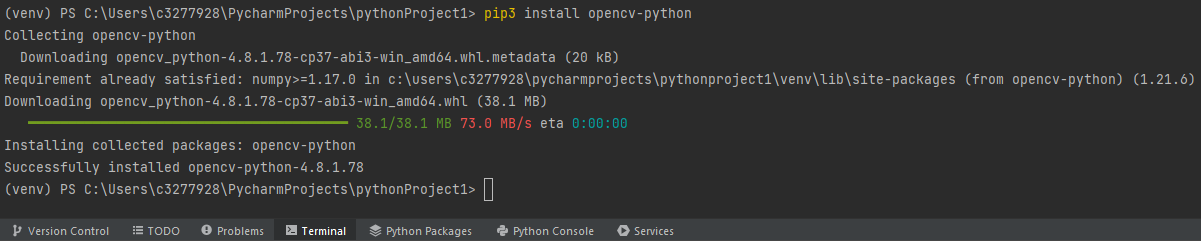
1. Enter the command: py -m ensurepip –upgrade

You should see ‘requirement already up to update with version 23.3.1 installed.’

Now install the opencv library:

1. Go to the terminal tab
2. Enter the command: pip3 install opencv-python

It should successfully collect and install the packages.

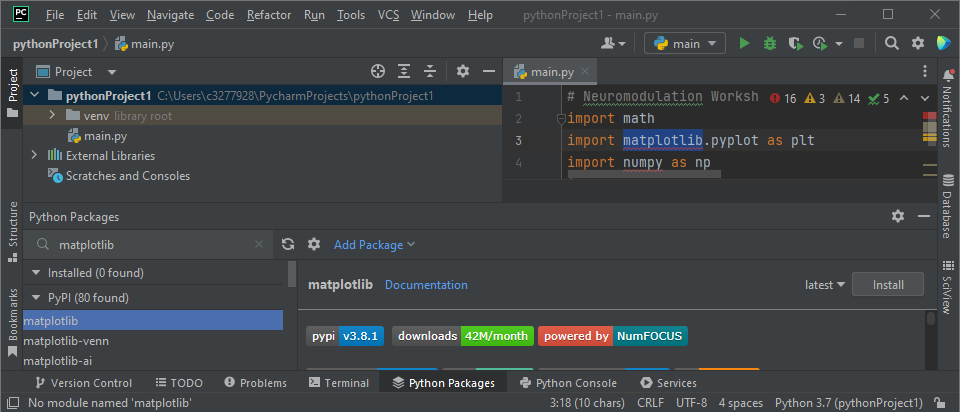


**Module 4: Installing other packages using the python packages tab**

The python packages tab can be located near the terminal and console tab. Please search for the libraries

* matplotlib,
* numpy,
* cv2,
* scikit-image,
* plotly,

and click the install button. Note that this may take a while to install.



Sometimes, **not required in this workshop**, different versions of libraries are needed to be compatible with code found on the internet. This can be done by selecting the dropdown menu next to the install button. You will often find in GitHub repositories what versions of python and libraries are required to run other peoples contributed code.

**Module 5: Coding exercises**

For the following 2 days we will be covering the exercises found on the GitHub Repository. For each exercise we hope to achieve the following outcomes:

1. Part One
   1. ‘Exercise1.py’: Simple arithmetic with numbers.
   2. ‘Exercise2.py’: Simple arithmetic with many numbers (vectors).
   3. ‘Exercise3.py’: Print to screen and write to a .txt file.
   4. ‘Exercise4.py’: read from a csv file and perform some statistical analysis.
   5. ‘Exercise5.py’: read from a csv file and plot the data.
   6. ‘Exercise6.py’: read from a csv file and create a histogram plot.
   7. ‘Exercise7.py’: read from a csv file and create a density plot.
   8. ‘Exercise8.py’: read from a csv file and re-create the original image.
2. Part Two
   1. ‘exe1.py’: display a jpg image.
   2. ‘exe2.py’: separate an RGB image.
   3. ‘exe3.py’: use preprocesses on the image to reduce noise.
   4. ‘exe4.py’: create a mask by thresholding the image.
   5. ‘exe5.py’: manipulate the mask image using morphological processes.
   6. ‘exe6.py’: label the masked image into different regions of interest (ROI).
   7. ‘exe7.py’: plot the labelled image.

**Module 5: Important libraries and functions**

A function is python is something that takes in an argument (input) and returns something back (output). A function may need many arguments and may return many outputs. Libraries contain pre-built functions that you can use. Please insert the following lines at the top of your code. This will install the libraries needed into your virtual environment.

*import math*

*import matplotlib.pyplot as plt*

*import numpy as np*

*import pandas as pd*

*import cv2*

*from skimage.measure import label, regionprops, regionprops\_table*

*import plotly*

*import plotly.express as px*

*import plotly.graph\_objects as go*

*from skimage import data, filters, measure, morphology*

Functions typically are created to perform computationally quicker or to perform tasks. In the first exercise we use the ‘print()’ function. This function takes in a string (input) and prints that string to the console (as the output). Some other useful functions (highlighted in red) are listed below, with their outputs highlighted in blue, and there arguments highlighted in green.

* average = np.mean(array): the function np.mean takes in a value or many values and returns the mean of those values in the variable average.
* data = pd.read\_csv('cell\_properties.csv'): the function pd.read\_csv takes in a string (a filename) and returns the result into a variable called data.
* image = plt.imread(filename): the function plt.imread takes in a string stored inside a variable filename and returns an image stored in the variable image

Note that the functions listed start with ‘np’, ‘pd’ or ‘plt’. This is because they come from those libraries. *For example* ***np.mean()*** *comes from the* ***numpy*** *library which we have named as* ***np****.* Also, functions may have other input and output options available. You can check them out online at the library’s documentation. For example, try: <https://numpy.org/doc/stable/reference/generated/numpy.mean.html>