2IMV20 Course. Python Installation Guide

In this guide, we are going to set up our developing environment for the first assignment of the 2IMV20 course. Let's start by installing the IDE that we will be using during the first assignment.

Installing Pycharm

Windows and MacOS

Go to the following link (https://www.jetbrains.com/pycharm/download/) and download the executable for either Pro or Community Edition version. If you choose Pro version, you will need to get a license from Jetbrains with your TUe account.

Once you have the installation file, just run it and follow the steps indicated.

Linux

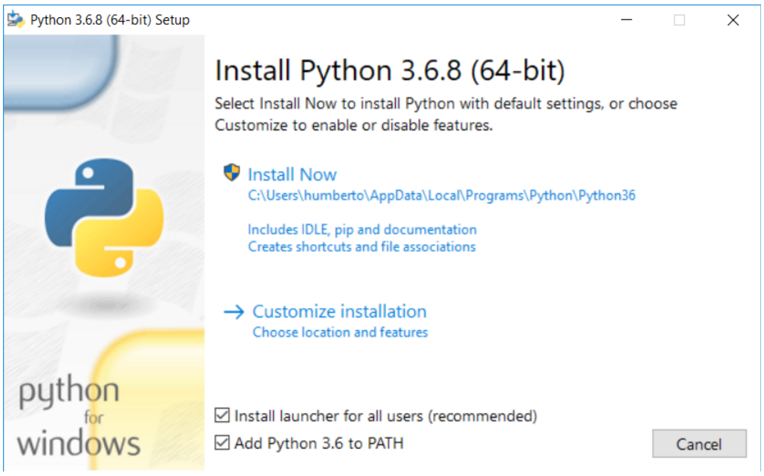
For linux users you can either download the installation files from https://www.jetbrains.com/pycharm/download/, or use the package manager of your distribution if Pycharm is included in the repositories (e.g., Ubuntu).

Installing Python

We need python in our system in order to interpret our code. We will use **Python 3.6**.8. It might be that higher versions of Python work just fine, but through this guide we will stick to version 3.6.8.

Windows and MacOS

Go to https://www.python.org/downloads/release/python-368 and download the installer. Execute the file. You will get a window similar to the one below. Make sure you mark the option "Add Python 3.6 to PATH". Then click "Install now".



Install Python 3.6.8 (64-bit)

Select Install Now to install Python with default settings, or choose Customize to enable or disable features.

Install Now
C:\Users\humberto\AppData\Local\Programs\Python\Python36

Includes IDLE, pip and documentation
Creates shortcuts and file associations

→ Customize installation
Choose location and features

☑ Install launcher for all users (recommended)
☑ Add Python 3.6 to PATH

Linux

Usually, linux distributions already have a version of python installed. If that is not your case, simply use the package manager of your distribution to install python (e.g., *sudo apt-get install python3.6*).

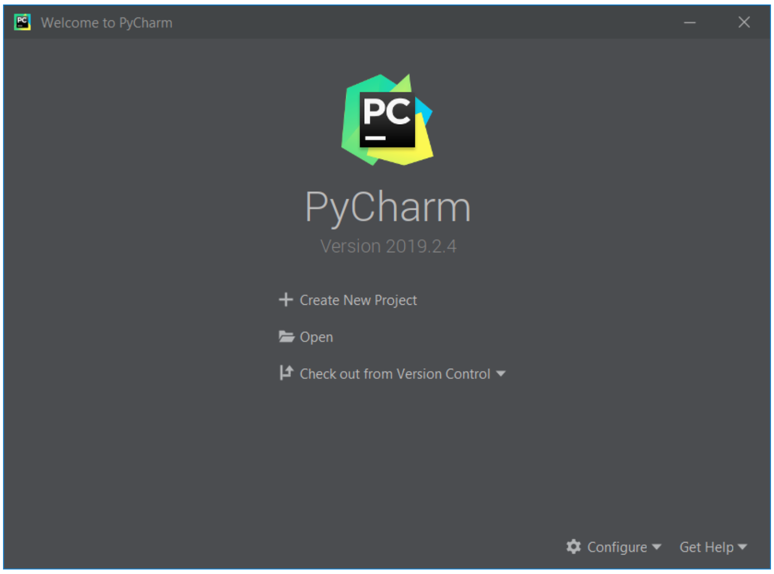
Downloading the source code

It is time to download the source code from canvas. Just go to the Canvas website of the course and download the ZIP file. Once finished, unzip the content into a folder in your computer.

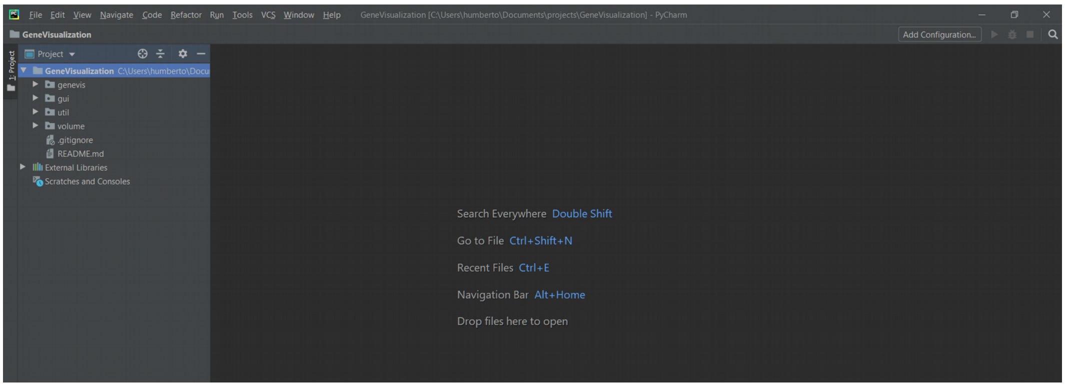
Creating a virtual environment

Virtual environments are useful to encapsulate the binaries that are necessary for a project, keeping all the changes and modules local to your project. To create our virtual environment, we will have to follow these steps:

1. Open PyCharm.
2. If this is the first time you open it, you will see some windows to configure the IDE. Just follow the steps.
3. When you have finished, you will see a window like this:



4. Click Open.
5. Find the folder with the source code that you downloaded previously.
6. Open it. You will get something like this:



GeneVisualization [C:\Users\humberto\Documents\projects\GeneVisualization] - PyCharm

GeneVisualization

GeneVisualization C:\Users\humberto\Docu

Search Everywhere Double Shift

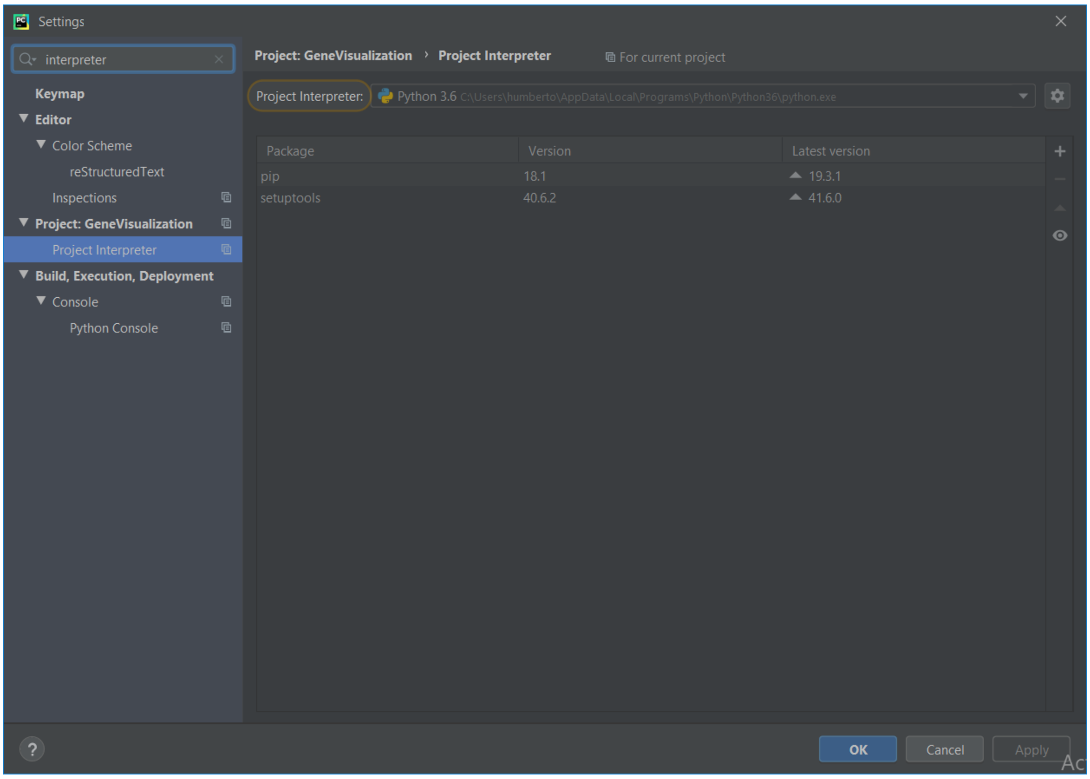
Go to File Ctrl+Shift+N

Recent Files Ctrl+E

Navigation Bar Alt+Home

Drop files here to open

7. Now we have to create the virtual environment for our project. Go to File > Settings. In the new window, just type *project interpreter* and you will see something like this:



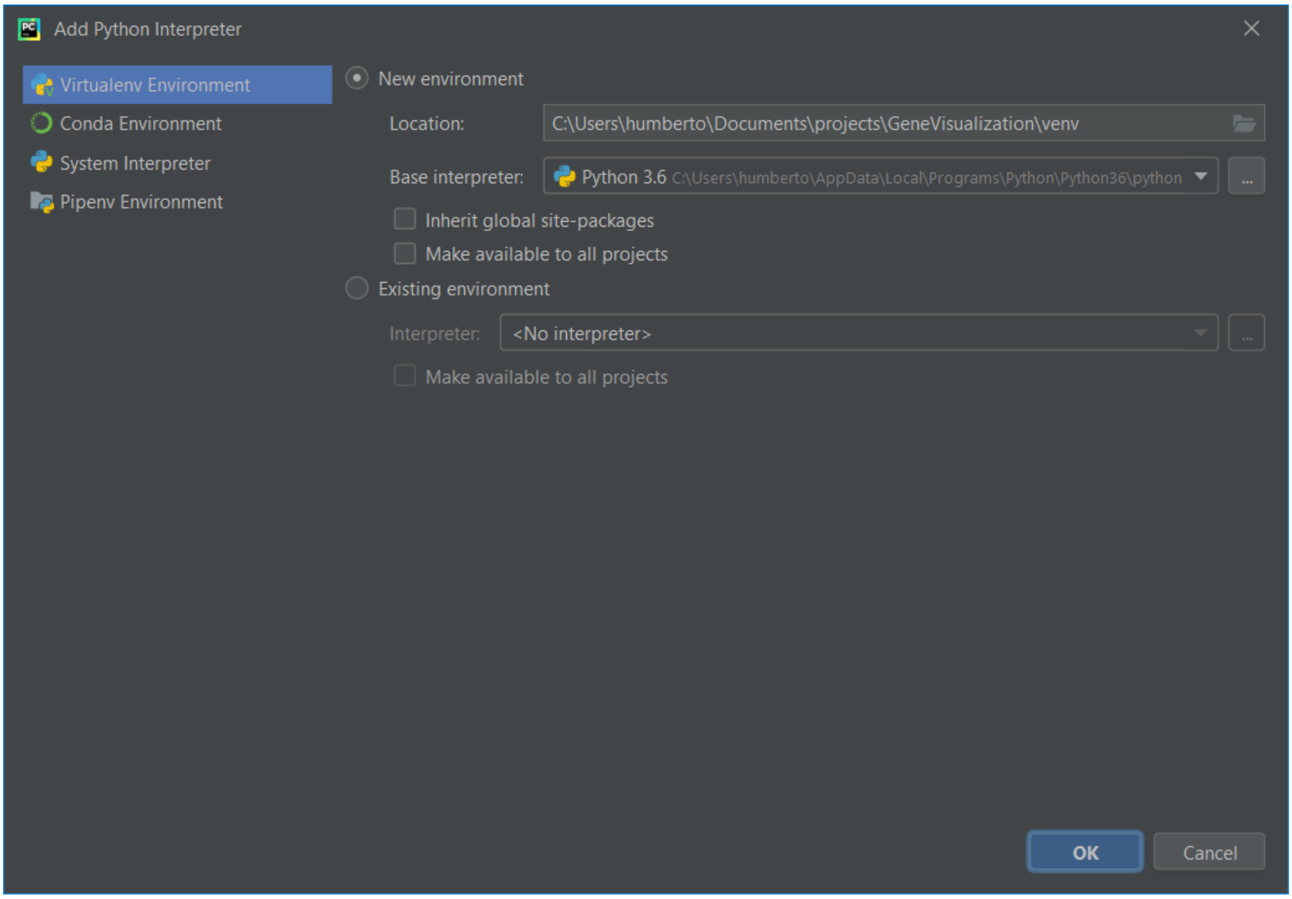
interpreter

Project: GeneVisualization › Project Interpreter For current project

Project Interpreter: Python 3.6 C:\Users\humberto\AppData\Local\Programs\Python\Python36\python.exe

Package	Version	Latest version
pip	18.1	▲ 19.3.1
setuptools	40.6.2	▲ 41.6.0

8. Right now, as can be seen, the project is configured to use the global python interpreter, thus we need to change it. Click over the gear icon that is at the right edge of the window and then “Add...”. It will pop up another window like this:



● New environment

Location: C:\Users\humberto\Documents\projects\GeneVisualization\venv

Base interpreter: Python 3.6 C:\Users\humberto\AppData\Local\Programs\Python\Python36\python

□ Inherit global site-packages

□ Make available to all projects

○ Existing environment

Interpreter: <No interpreter>

□ Make available to all projects

9. As you can see, it already gives the Virtualenv Environment as the default option. Just make sure that the location matches the path of your project (plus the folder for the virtual environment, in this case *venv*), and that the base interpreter is correctly set to Python 3.6. Just click OK and you will be done. Now you have your virtual environment.

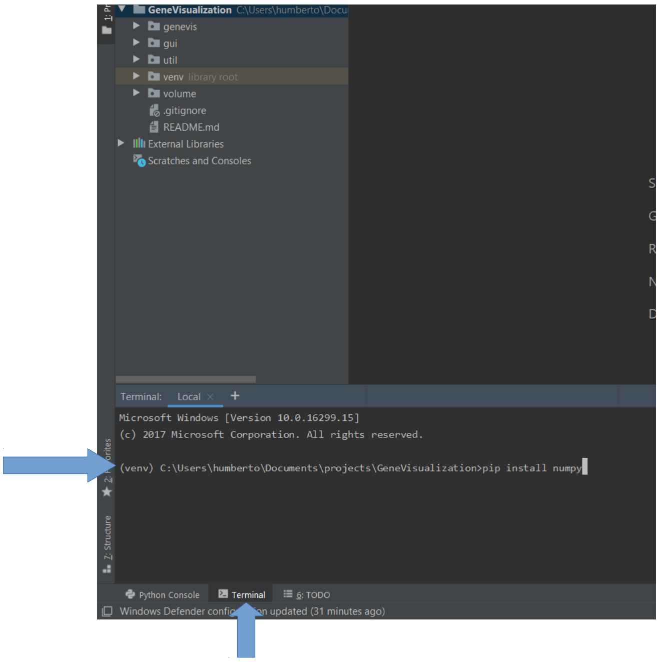
Installing Project Dependencies

Once we have our pycharm and python working, we can start to install the dependencies of our project. Python uses *pip* as a package manager and it can be run from the terminal that is . We need to install the following dependencies:

- Numpy
- PyOpenGL
- wxPython
- itk

Numpy

Within Pycharm IDE, go to the tab “Terminal” at the bottom of the window. Make sure you are with your virtual environment active. This is done automatically by the IDE and you should notice a (venv) prefixing the prompt of your terminal, like in the picture:



```
Microsoft Windows [Version 10.0.16299.15]
(c) 2017 Microsoft Corporation. All rights reserved.

(venv) C:\Users\humberto\Documents\projects\GeneVisualization>pip install numpy
```

Then type “*pip install numpy*” without the quote marks, and press enter. It will then install Numpy into your virtual environment.

PyOpenGL

Repeat the same process as before, but now type “*pip install PyOpenGL==3.1.3rc1*”.

NOTE: If by any chance you get an error when installing it on Windows, you can try with a pre-compiled version of PyOpenGL. To do so, go to this website https://www.lfd.uci.edu/~gohlke/pythonlibs/#pyopengl and download the corresponding version of PyOpenGL. Note that you must download the file that matches your Python version (cp36-cp36m) and architecture (32 or 64 bits). Once downloaded, just type “*pip install PATH_TO_THE_FILE*” where PATH_TO_THE_FILE indicates the route in your file system to the downloaded file (e.g., C:\Users\humberto\Downloads\PyOpenGL-3.1.3rc1-cp36-cp36m-win_amd64.whl).

wxPython

Windows and MacOS

Just repeat the same process as before and type “*pip install wxPython*”.

Linux

In Linux you could also type “*pip install wxPython*”. However, this would require some external libraries in order to compile the binaries. Therefore, it is better to download the corresponding wheel for your system (linux distribution) and Python version (cp36-cp36m). You can download it from https://extras.wxpython.org/wxPython4/extras/linux/gtk3/ and then type “*pip install PATH_TO_THE_FILE*”, where PATH_TO_THE_FILE indicates the route in your file system to the downloaded file (e.g., /home/humberto/Downloads/wxPython-4.0.7-cp36-cp36m-linux_x86_64.whl).

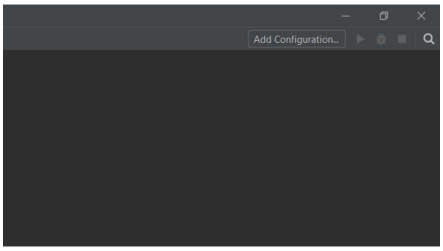
ITK

For installing ITK just type “*pip install itk*”.

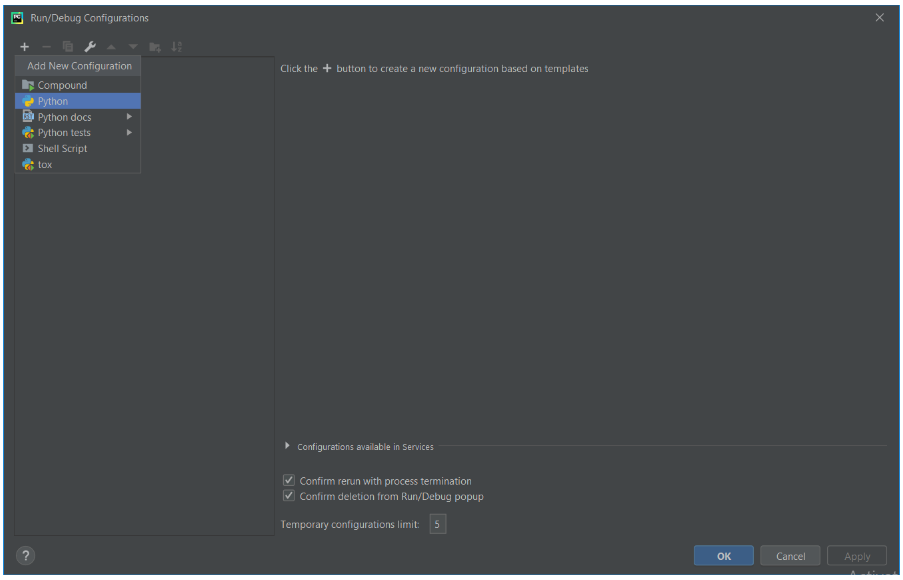
Running the project

To run the project you will have to create a configuration. This is very simple and can be done following these steps:

1. Click “Add Configuration...” at the top-right corner of the window



2. It will pop up another window. Click the “+” button and then select Python:



Click the + button to create a new configuration based on templates

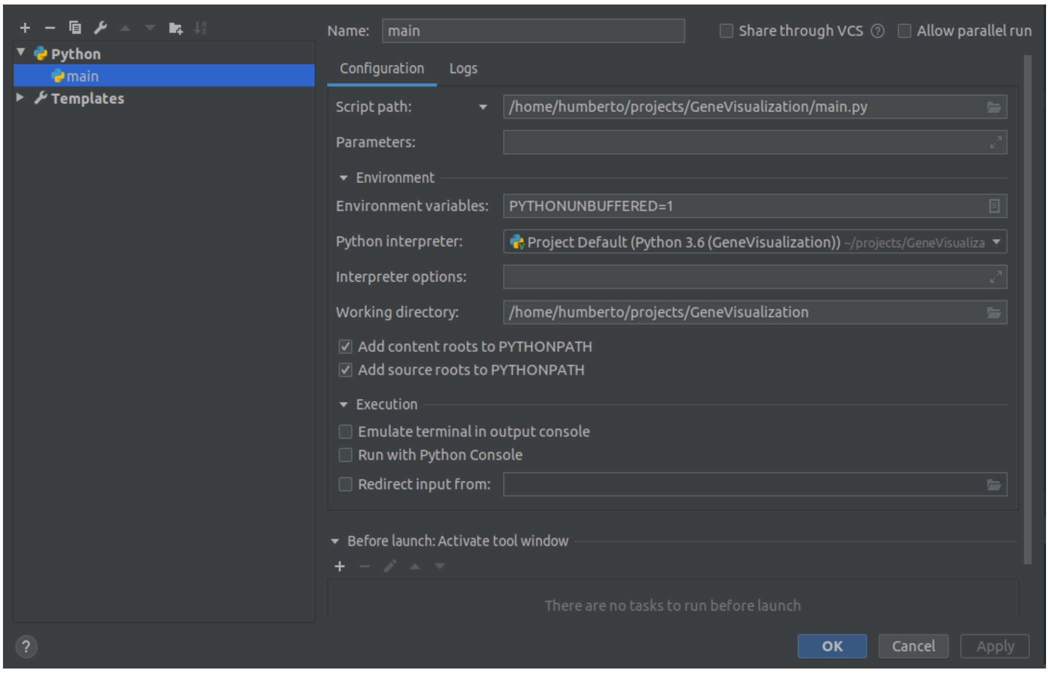
Configurations available in Services

☑ Confirm rerun with process termination

☑ Confirm deletion from Run/Debug popup

Temporary configurations limit: 5

3. Now, in Script path click the folder icon and select, within your project, the file gui/Application.py. Fill some name for the configuration and make sure that Python interpreter is the one located in your virtual environment (*venv* folder):



Name: main

☐ Share through VCS ☐ Allow parallel run

Script path: /home/humberto/projects/GeneVisualization/main.py

Parameters:

Environment

Environment variables: PYTHONUNBUFFERED=1

Python interpreter: Project Default (Python 3.6 (GeneVisualization)) ~/projects/GeneVisualiza

Interpreter options:

Working directory: /home/humberto/projects/GeneVisualization

☑ Add content roots to PYTHONPATH

☑ Add source roots to PYTHONPATH

Execution

☐ Emulate terminal in output console

☐ Run with Python Console

☐ Redirect input from:

Before launch: Activate tool window

There are no tasks to run before launch

4. Click OK. Now your project is ready to be run by clicking the Play button at the top-right corner.