Environment Setup

You can use your global Python installation or create a virtual environment to run the python codes.

Global Installation

By installing a global Python and adding packages to it, you can use such environment for all your Python-based projects. To install a global Python, you can find great instructions to install on different platforms at https://www.tutorialspoint.com/python/python_environment.htm.

The downside of this approach is that every project may need different version of a package that you already have installed for previous projects and updating those packages may cause breaking those older projects.

Virtual Environment

Here we provide instructions to set up a virtual environment through python *virtualenv* or *conda*. We specify the target platform (Windows, Linux, MacOs) when applicable.

I. Set up python virtual environment (Virtualenv):

1. Open a terminal command line and install Virtualenv package (if not installed before):

```
pip install virtualenv
```

2. Create a folder (e.g., *my_project*) for your project and change directory to it

```
mkdir my_project
cd my_project
```

3. Create your virtual environment (e.g., your_env_name)

```
virtualenv your_env_name
```

This will create a folder named your_*env_name* inside your *my_project* folder.

4. Activate your virtual environment

linux/Mac:

```
source your_env_name/bin/activate
```

Windows:

```
your_env_name\scripts\activate
```

The name of your environment inside a parenthesis will appear at the beginning of command prompt; e.g., **`(your_env_name)`** will appear for the above example.

4. Install the required packages (given that you have a file named *requirements.txt* in you current directory that includes packages to be installed)

```
pip install -r requirements.txt
```

5. Deactivate your environment (After you finish working on your project)

```
deactivate
```

II. Set up Conda Virtual Environment

Alternatively, you may also create your virtual environment through conda.

1. Check conda is installed in your machine

   `conda -V`
   * If conda is installed you should see something like `conda 3.7.0`.

2. Check conda is up to date

   `conda update conda`

   * Update any packages if necessary by typing `y` to proceed.

3. Create a virtual environment for your project given ***your_env_name*** is the name you want to call your environment

   `$ conda create -n your_env_name python=3.6 anaconda`

   1. Replace 3.6 with the Python version you wish to use. (To see a list of available python versions first, type `conda search "^python$"` and press enter.)
   2. Press `y` to proceed. This will install the Python version and all the associated anaconda packaged libraries at “path_to_your_anaconda_location/anaconda/envs/your_env_name”

4. Activate your virtual environment.

   `source activate your_env_name`

   * Activating a conda environment modifies the PATH and shell variables to point to the specific isolated Python set-up you created. The command prompt will change to indicate which conda environment you are currently in by prepending **(*your_env_name*)**. To see a list of all your environments, use the command `conda info -e`.

5. Install additional Python packages to a virtual environment. You need to specify each package name `[package]` individually.

```
conda install -n your_env_name [package]
```

* If you do not specify "*-n your_env_name*", conda will install the package to the root Python installation.

6. Deactivate your virtual environment.

```
source deactivate
```

* There is no need to specify the environment name. Whichever is currently active will be deactivated, and the PATH and shell variables will be returned to normal.

7. Delete a no longer needed virtual environment

```
conda remove -n your_env_name -all
```