Python Installation - Windows + Linux

Prerequisites: You should have WSL enabled with a corresponding Linux distribution if you are on Windows — most likely Ubuntu. Different Linux distributions will have different package managers (e.g., `yum` for Red Hat instead of `apt-get`), but this should not be an issue if you followed the setup from EECS 280/281. EECS 445 requires **Python 3.11**.

First, check if Python is already installed and if so, what version it is.

```
$ python --version
```

- If you have **Python 2**: install Python 3.11, and run all **`python`** commands with **`python3`**
- If you have an older version of Python 3, install Python 3.11, and run all **`python`** commands with **`python3.11`**

Install Python 3.11:

```
$ sudo apt-get install software-properties-common
$ sudo add-apt-repository ppa:deadsnakes/ppa
$ sudo apt-get update
$ sudo apt-get install python3.11
$ sudo apt-get install python3.11-venv
```

Or, download Python (3.11) from https://www.python.org/downloads/ and follow the instructions. There are many ways to do this, but the above should work for most.

If you prefer, you can install Python from source code files. Download the Python 3.11 source files from https://www.python.org/downloads/source/ and follow these instructions for installing from source on WSL/Linux (**note:** the tutorial uses Python 3.8, but you should instead use Python 3.11).

Python Installation - MacOS

Prerequisites for Mac:

- Have homebrew installed (if you took EECS 280, this should be true)

First, check if Python is already installed and if so, what version it is.

```
$ python --version
```

- If you have **Python 2**: install Python 3.11, and run all **`python`** commands with **`python3`**
- If you have an older version of Python 3, install Python 3.11, and run all **`python`** commands with **`python3.11`**

To install Python 3:

```
$ brew install python3
```

Or, download Python (3.11) from https://www.python.org/downloads/ and follow the instructions.

Running a script (all OS)

Suppose you want to run the Python file named `test.py`. In your terminal, type:

```
$ python test.py
```

and it should run! Remember to be in the right directory.

NOTE: If you installed Python 3.11 as either **`python3`** or **`python3.11`** in previous steps, make sure to use those commands instead of **`python`**.

To create a virtual environment (all OS)

When programming in Python, we often want to use a different set of packages (third-party code) for different projects - this is especially true for assignments in EECS 445.

To achieve this, Python allows us to create **virtual environments** - a folder containing a fresh copy of Python + all the libraries we want for our current project. The command below creates a new virtual environment in a folder called “env” in your working directory.

```
$ python -m venv env
```

To activate the virtual environment, run:

```
$ source env/bin/activate
```

This tells your terminal to use the version of Python installed in your virtual environment.

Once you’re done working on the current project, deactivate the virtual environment by running:

```
$ deactivate
```

NOTE: install all packages and do your work inside the (activated) virtual environment — this will make your life a lot easier since you will not have to keep track of outside dependencies! Use the following command to install packages.

```
$ pip install [package_name]
```

Installing Jupyter (all OS)

The coding portion of your homeworks will be in Jupyter notebooks. Activate your virtual environment and run the following commands.

```
$ pip install jupyter notebook
$ jupyter notebook
```

NOTE: Copy/paste the URL into your browser if Jupyter Notebook doesn't automatically open.

Using an IDE:

Using an IDE has many advantages, including

- ***automatic virtual environments***,
- ***debugging support***,
- easier graphing and plotting,
- git integration,
- better code autocomplete,
- and Jupyter Notebook integration.

We highly recommend that you use an IDE, and recommend one of the following:

- **VSCode.** A standard IDE which is used for many languages. To work with Python you should install the Python extensions and follow their quick start guide. If you're WSL, you may need to follow this WSL tutorial as well.
- **PyCharm.** An IDE with more Python-specific features than VSCode. You can get a free student license and then follow their quick start guide. If you're using WSL, you may need to follow this WSL tutorial as well.

If you have issues, both IDEs have very extensive documentation.