

Development Environment

1. Virtual Environments

1.1 Why?

- 3rd-party software, namely libraries, has been widely used.
- Programs may require different versions of a certain library. Each version will have one or multiple substantial binary files in your system. However, those files can have the same name. Thus, it is difficult to sustain multiple versions of the same library in the system (i.e., you may have to carefully maintain a complex dependency graph of library files).
- Place programs that require the same libraries into the same environment.

1.2 Installation

- `conda` is the most popular virtual environment management tool.
- Either `anaconda` or `miniconda` will work.
- Check out more details here: <https://conda.io/projects/conda/en/latest/user-guide/install/index.html>

1.3 Create a Virtual Environment with Python

- Typically, the command looks like this

```
conda create -n myenv_py312 python=3.12
```

- You can specify the name of your virtual environment and the version of Python.
- Check out more details: <https://conda.io/projects/conda/en/latest/user-guide/tasks/manage-environments.html#creating-an-environment-with-commands>
- There is a default virtual environment named `base` .

1.4 Activate a Virtual Environment

- Typically, the command looks like this

```
conda activate myenv_py312
```

1.5 Deactivate the Current Virtual Environment

- Typically, the command looks like this

```
conda deactivate
```

1.6 Remove a Virtual Environment

- First, you need to deactivate from the virtual environment you want to remove if you are in it.
- Typically, the command looks like this

```
conda env remove -n myenv_py312
```

1.7 Check All Existing Virtual Environments

- Typically, the command looks like this

```
conda env list
```

1.8 Install Libraries Using `conda`

- `conda` can be considered as a big repository with many libraries stored there.
- `conda` is like a farmer's market, and different booths may offer same products. In `conda`, such a booth is called a "*channel*". `conda` does provide a default channel `pkgs/main`, and you do not need to specify this channel if you want to install libraries therefrom.
- Before you install a library, you'd better search it in channels. For example, I would like to install `numpy`, and after searching on Google, I found that it is also provided in the channel `conda-forge` (which is another very popular channel). Then, I will search `numpy` in `conda` repo like this

```
conda search -c conda-forge numpy
```

And, the results will show all available versions of `numpy` in both `pkgs/main` and `conda-forge`.

- If you have no idea about the official name of a library in `conda`, simply search on Google.
- Then, you choose the specific version you need, and install it from a specific channel. For example, we will install `numpy` from `conda-forge`.

```
conda install -c conda-forge numpy
```

This is also the official installation guide provided by NumPy: <https://numpy.org/install/>

- For convenience, if a channel will be frequently used, we can add this channel into our `conda` configuration. For example, I will add `conda-forge` this channel into my `conda` configuration.

```
conda config --env --add channels conda-forge
```

1.9 Check a Library in conda

- Typically, the command looks like this

```
conda list numpy
```

- This command will show everything you have installed related to the keyword "jupyterlab".

1.10 Remove a Library from conda

- Typically, the command looks like this

```
conda remove numpy
```

1.11 Update Libraries

- You can use the following command to update all packages

```
conda update --all
```

1.12 Update conda

- You can update conda itself

```
conda update -n base conda
```

1.13 Install Packages Using pip

- In a virtual environment, you can also use pip, instead of conda, to install packages. Beforehand, make sure you have activated your virtual environment so that the pip program is the one installed inside your virtual environment.

```
pip install numpy
```

2. IDEs

2.1 PyCharm

- Free for education: <https://www.jetbrains.com/pycharm/>

2.2 Visual Studio Code

- Free: <https://code.visualstudio.com/>

2.3 Link PyCharm to Virtual Environments

- <https://www.jetbrains.com/help/pycharm/conda-support-creating-conda-virtual-environment.html#conda-requirements>

2.4 Link Visual Studio Code to Virtual Environments

- <https://code.visualstudio.com/docs/python/environments>

2.5 JupyterLab

- <https://jupyter.org/install>
- We need to start the JupyterLab server before using it.

```
jupyter lab --no-browser --ip "*" --notebook-dir [PATH TO YOUR WORK FOLDER]
```

Replace [PATH TO YOUR WORK FOLDER] with your path.

- Download and install a GUI client for Windows: <https://github.com/jupyterlab/jupyterlab-desktop>
- Use the client to connect to the server locally or remotely.