Install Anaconda or Miniconda and start the following procedure

Creating virtual environment: (name of environment is pytorch)

```
conda create -n pytorch python=3
activate pytorch
conda list
conda install numpy pandas matplotlib
```

Installing packages:

To install a package type following command:

```
conda install PACKAGE_NAME
```

For example, to install numpy, type `conda install numpy`.

You can install multiple packages at the same time.

```
conda install numpy scipy pandas
```

you can also specify which version of package you want to install:

`conda install numpy=1.10`

Removing packages:

```
conda remove PACKAGE_NAME
```

Update packages:

```
conda update package_name
```

To update all packages:

`conda update --all`

To see all package's, you have installed:

conda list

MANAGING ENVIRONMENTS

To create environment:

```
conda create -n env_name [python=3.7] [LIST_OF_PACKAGES]
```

```
conda create -n my_env python=3.7 numpy Keras
```

Entering environment:

```
conda activate my_env
```

```
activate my_env
```

To see the installed packages:

Conda list

Deactivation an environment:

Conda deactivate

MORE ENVIRONMENT ACTIONS:

Saving and loading environments:

A really useful feature is sharing environments so others can install all the packages used in your code, with the correct versions. Let's see all the package-names, including the Python version present in the current environment, using the command:

```
conda env export
```

you can save all seen information to YAML file.

```
conda env export > environment.yaml
```

Now you can share this file to others.

Create an environment

To create an environment from an environment file, use the following command:

```
conda env create -f environment.yaml
```

If you forgot your environment:

```
conda env list
conda info --envs
```

To view the list of packages, run the following command in your terminal / Anaconda Prompt,:

```
# If the environment is not activated
conda list -n env_name

# If the environment is activated
conda list

# To see if a specific package, say `scipy` is installed in an environment
conda list -n env_name scipy
```

Removing an environment

```
conda env remove -n env_name
```

Sharing Environments

```
conda env export > environment.yaml
```

Share the List of Dependencies:

For users not using conda, you may want to share the list of packages installed in the current environment. You can use `pip` to generate such a list as `requirements.txt` file using:

```
pip freeze > requirements.txt
```

you can install all the packages mentioned in the `requirements.txt` file using:

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

NOTE: You can use pip along side conda to install packages