`Virtualenv`

Virtual environment is a Python environment which will help you to install different versions of same Python library across single system.

- For example, consider `requests` library.
  - For a project named “ABC”, you may be using version `requests==2.28.1`
  - And for another project named “XYZ”, you may be using older version of requests library `requests==2.28.0`
  - Now the question is how can we deploy same library / package on the single server?
  - **The solution:** we can create independent virtual environments for each of the project and under respective project folder. Like -
    - `venv1` (for project `ABC` with `requests==2.28.1`)
    - `venv2` (for project `XYZ` with `requests==2.28.0`)

Steps to create

- To create virtual environment, we need a tool called `virtualenv`. Install it using `pip` (package manager)

```
pip install virtualenv
```

- To create a virtual environment -

```
python3 -m venv virt1
```

You can name your virtual environment anything you want. Also, at whatever path you run the command to create virtual environment a directory structure with the same name will be created.

- In order to “**activate**” virtual environment -

```
# NOTE - replace `venv` with whatever is the name of your own virtual environment.
# NOTE - You should execute this command from the same path where venv exists.

# Windows ::
.\venv\Scripts\activate

# Unix ::
source venv/bin/activate
```

- In order to “**deactivate**” virtual environment -

```
# NOTE - replace `venv` with whatever is the name of your own virtual environment.
# NOTE - You should execute this command from the same path where venv exists.

# Windows ::
.\venv\Scripts\deactivate

# Unix ::
deactivate
```

You can install any external packages under `venv` and in order to check the list of packages installed (including their versions), run command `pip freeze` under that particular active virtual environment.

Why do we use `virtualenv`?

- This way you can have many different environments for all of your development needs.
- Always remember to create virtual environments while developing new applications.
- This will help you to isolate projects and their require library versions.