#5_Python_Virtual_Environments

13/05/2025

- Virtual environments:

  ① -> manage project dependencies

  ② -> isolates environments from each other

-> virtual environments ensure that dependencies do not conflict and projects remain organised.

-> virtual environments do not affect the global environment

-> avoid conflicts between different projects.

-> allow for clean and portable environment.

-> ensure consistency across different machines

Overview

- Can run different versions of Python ~~as~~ in each environment, including associated packages.

enva :	Python 3.9
envb :	Python 3.10 & Flask 2.2.1
envc :	Python 3.11 & Flask 2.3.1

Setting Up Virtual Environments:

- all virtual environments should be set up in a central directory.

```
~/.venv
```

```
python3.9 -m venv ~/.venv/enva
python3.11 -m venv ~/.venv/envb
```

virtual environment envb

Python version

module

directory where envb is setup.

tells python to run venv module

activating virtual environment:

13/01/2025

```
source ~/.venv/envb/bin/activate
```

this' prompt activates the virtual environment

```
pip install flask==2.3.1
```

installs Flask.
may need to use pip3

Flask versions older than 2.2.0 require the install of Werkzeug

```
pip install werkzeug==2.2.2 flask==2.1.2
```

```
deactivate
```

deactivates the virtual environment.

Using Virtual Environments

```
source ~/.venv/envb/bin/activate   # Step 1
python b.py                        # Step 2
deactivate                         # Step 3
```

Step 1: activates the environment envb
Step 2: runs the application b.py
Step 3: deactivates envb