CREATING A VIRTUAL ENVIRONMENT

What is a Virtual Environment?

⇨ A virtual environment is an **isolated Python workspace** that allows you to install and manage project-specific libraries without affecting the global Python installation.

❖ Why Use a Virtual Environment?

- Prevents dependency conflicts between projects
- Keeps packages project-specific
- Makes projects portable and reproducible
- Avoids permission issues from global installs

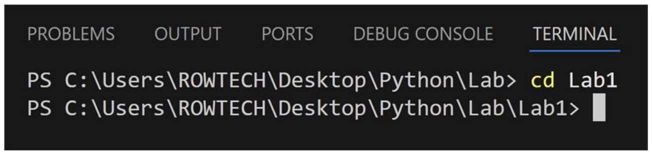
Creating a Virtual Environment

Step 1: Navigate to Project Directory

Command:

```
cd project_directory
```

Output:



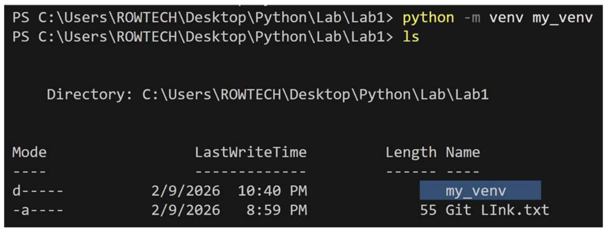
```
PS C:\Users\ROWTECH\Desktop\Python\Lab> cd Lab1
PS C:\Users\ROWTECH\Desktop\Python\Lab\Lab1>
```

Step 2: Create the Virtual Environment

Command:

```
python -m venv env_name
```

Output:



```
PS C:\Users\ROWTECH\Desktop\Python\Lab\Lab1> python -m venv my_venv
PS C:\Users\ROWTECH\Desktop\Python\Lab\Lab1> ls


    Directory: C:\Users\ROWTECH\Desktop\Python\Lab\Lab1


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----          2/9/2026  10:40 PM                my_venv
-a----          2/9/2026   8:59 PM             55 Git LInk.txt
```

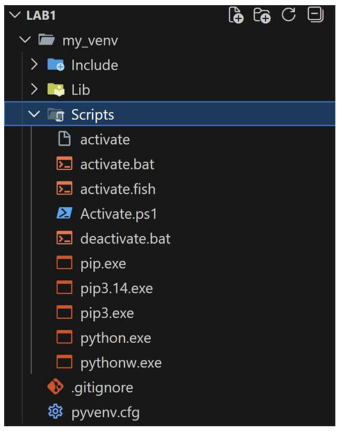
❖ python -m venv → invokes the built-in venv module
❖ env_name → user-defined environment name (e.g., venv, env, myenv)

Virtual Environment Directory Structure

➢ **After creation, the environment contains:**

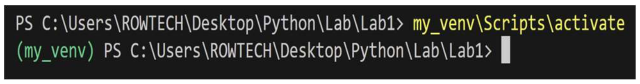
- Scripts/ (Windows) or bin/ (Linux/macOS) → activation scripts
- Lib/ → installed site-packages
- pyvenv.cfg → Python environment configuration

❖ **Example:**



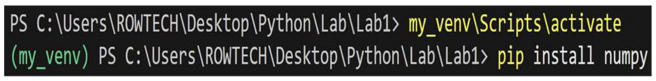
```
LAB1
  my_venv
    Include
    Lib
    Scripts
      activate
      activate.bat
      activate.fish
      Activate.ps1
      deactivate.bat
      pip.exe
      pip3.14.exe
      pip3.exe
      python.exe
      pythonw.exe
    .gitignore
    pyvenv.cfg
```

Activating the Virtual Environment

Windows:	
Command:	**Output:**
`env_name\Scripts\activate`	 `PS C:\Users\ROWTECH\Desktop\Python\Lab\Lab1> my_venv\Scripts\activate` `(my_venv) PS C:\Users\ROWTECH\Desktop\Python\Lab\Lab1>`
Linux / macOS	
Command:	
`source env_name/bin/activate`	
❖ **Activation Indicator:** ➢ The command prompt shows: (env_name)	

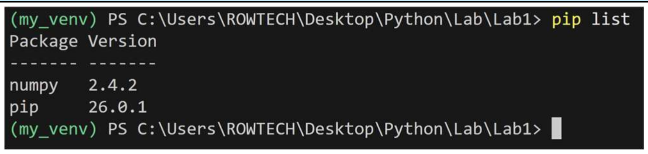
Package Management Inside Virtual Environment

❖ **Install a Package?**

Command:	Example:
`pip install package_name`	

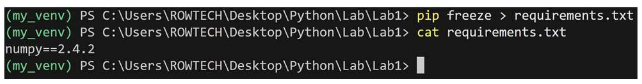
```
PS C:\Users\ROWTECH\Desktop\Python\Lab\Lab1> my_venv\Scripts\activate
(my_venv) PS C:\Users\ROWTECH\Desktop\Python\Lab\Lab1> pip install numpy
```

❖ **View Installed Packages**

Command:	Example:
`pip list`	

```
(my_venv) PS C:\Users\ROWTECH\Desktop\Python\Lab\Lab1> pip list
Package Version
------- -------
numpy   2.4.2
pip     26.0.1
(my_venv) PS C:\Users\ROWTECH\Desktop\Python\Lab\Lab1>
```

❖ **Freeze Dependencies**

Command:	Example:
`pip freeze > requirements.txt`	

```
(my_venv) PS C:\Users\ROWTECH\Desktop\Python\Lab\Lab1> pip freeze > requirements.txt
(my_venv) PS C:\Users\ROWTECH\Desktop\Python\Lab\Lab1> cat requirements.txt
numpy==2.4.2
(my_venv) PS C:\Users\ROWTECH\Desktop\Python\Lab\Lab1>
```

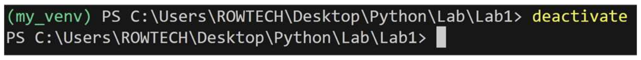
Using requirements.txt

❖ **To install all dependencies on another system:** This ensures identical project environments.

Command:
`pip install -r requirements.txt`

Using requirements.txt

❖ **This returns the shell to the global Python environment**.

Command:	Example:
`Deactivate`	

```
(my_venv) PS C:\Users\ROWTECH\Desktop\Python\Lab\Lab1> deactivate
PS C:\Users\ROWTECH\Desktop\Python\Lab\Lab1>
```