🧾 Lecture 03: Virtual Environments in Python

📓 1. What Are Virtual Environments?

✅ Definition:

A **virtual environment** is an **isolated workspace** in Python that allows you to:

- Keep **dependencies separate** for each project
- Avoid **conflicts** between different versions of libraries or Python

🧠 Why They Matter:

❌ Problem:

- Project A needs NumPy **v1.3**
- Project B needs NumPy **v2.3**
- Installing both in the same environment causes **errors** and conflicts

✅ Solution:

- Use a **virtual environment** for each project:
  - Each gets its **own Python + libraries**
  - No interference with other projects

⚙️ 2. How to Create a Virtual Environment

💻 Steps:

1. **Open Anaconda Prompt:**
   - Press **Windows key** → search **Anaconda Prompt** → open
2. **Navigate to Project Folder:**

```
bash
CopyEdit
cd Documents
cd Python
cd Python\ Programming
```

◈ *Use the `Tab` key to auto-complete folder names*

3. **Create the Environment:**

```
bash
CopyEdit
conda create -n firstenv python=3.10
```

   - `-n`: name of the environment → `firstenv`
   - `python=3.10`: Python version to use

4. **Confirm Installation:**
   - Type `y` and press Enter when prompted

🔄 3. Activating & Using a Virtual Environment

▶️ Activate Environment:

```
bash
CopyEdit
conda activate firstenv
```

- When activated: `(firstenv)` appears before the command line

📦 Installing Packages:

Tool	Command	Example
`pip`	Install specific version	`pip install numpy==2.3`
`conda`	Install general package	`conda install pandas`

📋 View Installed Packages:

```
bash
CopyEdit
conda list
```

⏹️ 4. Deactivating a Virtual Environment

```
bash
CopyEdit
```

```
conda deactivate
```

- Brings you back to base environment

🧠 5. Managing Multiple Environments

📜 List All Environments:

```
bash
CopyEdit
conda info --envs
```

Example:

```
mathematica
CopyEdit
# conda environments:
base         *  C:\Users\...
firstenv        C:\Users\...\firstenv
helloenv        C:\Users\...\helloenv
```

❌ Delete an Environment:

```
bash
CopyEdit
conda remove --name helloenv --all
```

- Confirm deletion by typing `y`

🧠 6. Key Points to Remember

Feature	Purpose
`conda create -n <name> python=<ver>`	Create new environment
`conda activate <name>`	Activate the environment
`conda deactivate`	Exit current environment
`conda list`	View installed packages

Feature	Purpose
`conda info --envs`	List all environments
`conda remove --name <name> --all`	Delete environment

□ 7. Practical Example

Scenario:

- Project A: Python 3.10 + NumPy 1.3
- Project B: Python 3.11 + NumPy 2.3

Steps:

1. **Create environments:**

```
bash
CopyEdit
conda create -n projectA python=3.10
conda create -n projectB python=3.11
```

2. **Install packages:**

```
bash
CopyEdit
conda activate projectA
pip install numpy==1.3

conda activate projectB
pip install numpy==2.3
```

3. **Switch between environments as needed:**

```
bash
CopyEdit
conda activate projectA
```

🔗 8. Extra Tips & Troubleshooting

Topic	Explanation
pip vs conda	`pip`: best for Python-only packages. `conda`: good for both Python + non-Python dependencies

Topic	Explanation
Check Python Version	`python --version`
Error Handling	If a version isn't found, use suggested alternatives

❗ Example Error:

```
arduino
CopyEdit
ERROR: Could not find a version that satisfies numpy==2.3
```

✔ Solution: Use a **different version**, like:

```
bash
CopyEdit
pip install numpy==1.24
```

✅ 9. Conclusion

- Virtual environments help **organize** and **separate** projects
- Prevent library **conflicts**
- Essential for professional development

🚀 **Practice Tip:** Try creating, activating, and deleting environments on your system!

📓 Difficult Terms Explained

Term	Meaning
Dependency	A software package your code needs to run (e.g., NumPy, Pandas)
Environment	A self-contained space where Python and packages live
Interpreter	The program that runs Python code

❓ Questions to Review

1. What is a virtual environment in Python?
2. Why are virtual environments important?
3. How do you create a virtual environment using `conda`?
4. How can you switch between environments?
5. What command lists all installed packages in the active environment?
6. What to do if a specific package version is not available?