

Understanding Virtual Environments in Python

Lecture Focus:

This lecture explores virtual environments in Python: what they are, why they're important, and how to create, activate, use, and manage them using Anaconda (conda) and pip.

1. What Are Virtual Environments?

Definition:

A virtual environment is an isolated Python workspace with its own interpreter and dependencies, used to keep projects independent of each other.

Why Important?

- To avoid conflicts between packages required by different projects:
 - Project A: Needs NumPy 1.3
 - Project B: Needs NumPy 2.3
 - Installing both in the same environment causes conflicts
 - Virtual environments prevent this by isolating packages per project

2. Creating a Virtual Environment

- Open Anaconda Prompt
- Navigate to your project folder:
 - `cd Documents\Python\Python_Programming`
- Create the environment:
 - `conda create -n firstenv python=3.10`
- Confirm installation when prompted (type y)

3. Activating and Using a Virtual Environment

- Activate:
 - `conda activate firstenv`
- Install packages:
 - `pip install numpy==2.3`
 - `conda install pandas`
- List installed packages:
 - `conda list`

4. Deactivating the Environment

- `conda deactivate`

5. Managing Multiple Environments

- List all environments:
 - `conda info --envs`
- Delete an environment:
 - `conda remove --name helloenv --all`

6. Key Points to Remember

Command	Description
conda create -n <name> python=<version>	Create a virtual environment
conda activate <name>	Activate it
conda deactivate	Deactivate it
conda list	List installed packages
conda info --envs	List all environments
conda remove --name <name> --all	Delete an environment