Virtual Environments in Python

Virtual Environments in Python

This tutorial explains how to use **virtual environments** in Python to manage different versions of packages for multiple projects on a single machine.

What is a Virtual Environment?

A virtual environment creates an **isolated space** for a Python project. This prevents conflicts when different projects require different versions of the same package.

How to Use a Virtual Environment

Code Example: Creation and Activation

```
# Creating a new virtual environment
python -m venv my_project_env

# Activating the environment on Windows (PowerShell)
.\my_project_env\Scripts\Activate.ps1

# Activating the environment on macOS/Linux
source my_project_env/bin/activate
```

Managing Dependencies

Code Example: `requirements.txt`

```
# Generate a requirements file
pip freeze > requirements.txt

# Install packages from the file
pip install -r requirements.txt
```