Setting Up a Python Virtual Environment

Tom Abel

2025-01-02

Table of contents

1 Setting Up a Python 3.10 Virtual Environment

For this class, we will use a dedicated Python virtual environment with Python version 3.10. A virtual environment is an isolated space where you can manage dependencies specific to a project, without interfering with other Python installations on your system. Follow these steps to set up the environment:

1.1 Step 1: Verify Python 3.10 Installation

First, check if Python 3.10 is installed on your system. Open a terminal or command prompt and run:

```
python3.10 --version
```

If this returns something like `Python 3.10.x`, you're good to go. If Python 3.10 is not installed, you can download it from the official Python website or use your system's package manager to install it.

1.2 Step 2: Create a Virtual Environment

Navigate to a directory where you'd like to store your virtual environment. If you already have a class folder called 360 you may change into it and execute the commands below:

Then run:

```
python3.10 -m venv myenv
```

This command creates a new virtual environment named `myenv` in the current directory. Replace `myenv` with a name of your choice if desired.

1.3 Step 3: Activate the Virtual Environment

To activate the virtual environment: - On **Linux/macOS**:

```
source myenv/bin/activate
```

- On **Windows**:

```
myenv\Scripts\activate
```

After activation, your terminal prompt will change to indicate that the virtual environment is active. All Python packages installed while the virtual environment is active will be isolated from your global Python installation.

1.4 Step 4: Install Required Packages

With the virtual environment activated, you can now install the required Python packages for this class. Use `pip` to install packages:

```
pip install numpy matplotlib scipy gala taichi plotly
```

1.5 Step 5: Deactivate the Virtual Environment

When you're done working in the virtual environment, deactivate it by running:

```
deactivate
```

This will return your terminal to its normal state.

1.6 Notes

1. Always activate the virtual environment before running scripts for this class to ensure the correct Python version and dependencies are used.
2. If you ever need to delete the virtual environment, simply remove the `myenv` directory.

This process ensures an organized and isolated Python development environment, making it easier to manage dependencies for your coursework.