Creating and Using Virtual Environments in Python

What is a Virtual Environment?

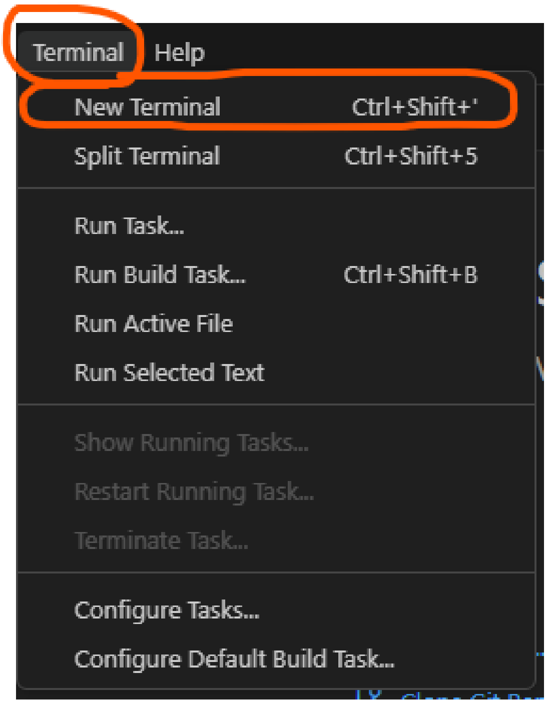
Python virtual environments are isolated environments for Python projects. They allow you to install project-specific packages and dependencies without affecting the global Python installation on your system. You can create a virtual environment using the **venv** module and activate it to install packages using **pip**. Virtual environments are essential for managing dependencies and ensuring that your code works as expected across different environments.

How to create a Virtual Environment in Python

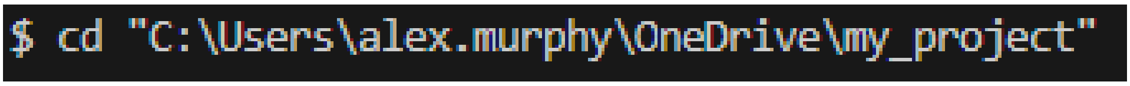
1. Open your command prompt or terminal.
2. Navigate to the directory where you want to create your virtual environment using **cd**.
    - e.g.: cd ~/Desktop
3. Type python -m venv <*python-environment-name*> and press enter (change the name of the environment to whatever you choose). This will create a new virtual environment in your current directory.
4. To activate the virtual environment, type source <*python-environment-name*>/Scripts/activate on Windows and press enter.
5. Once activated, you can install packages using pip install <*package-name*> and run your Python code as usual.
6. To deactivate the virtual environment, simply type deactivate and press enter.

Example in VS Code

1. Open the Terminal from the menu

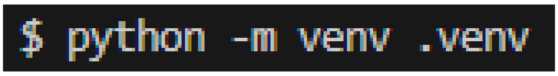


2. Navigate to directory where you want your python environment to be



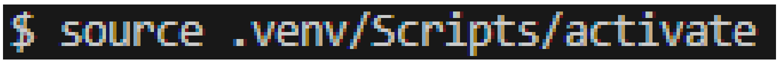
```
$ cd "C:\Users\alex.murphy\OneDrive\my_project"
```

3. Create the Virtual Environment in the directory



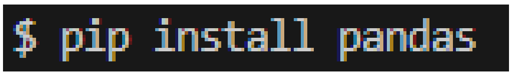
```
$ python -m venv .venv
```

4. Activate the Virtual Environment



```
$ source .venv/Scripts/activate
```

5. Now you can install any package you like (via pip) and it will only be installed on the virtual environment that you have set up.



```
$ pip install pandas
```

Sharing and Using Shared Virtual Environments

Sharing Virtual Environments

When you have built a project in a virtual environment, you can share it with others by following these steps:

1. Create a requirements.txt file that lists all the packages and dependencies required for your project. You can generate this file by running **pip freeze > requirements.txt** in your virtual environment.
2. Share the **requirements.txt** file with the person you want to share the project with.
3. Ask the person to create a new virtual environment and activate it.
4. Once the virtual environment is activated, they can install the required packages by running **pip install -r requirements.txt**.
5. They can now run your Python code as usual.

If you want to share the entire virtual environment, including the Python interpreter and installed packages, you can create a copy of the virtual environment directory (via SharePoint for instance) and share it with the person you want to share the project with. They can then activate the virtual environment and run your Python code as usual.

Remember that virtual environments are an essential tool for Python developers to manage dependencies and ensure that their code works as expected across different environments.

Using a Shared Virtual Environment

1. Ask the person who shared the virtual environment with you to provide you with the requirements.txt file.
2. Create a new directory where you want to store the virtual environment.
3. Create and activate the Virtual Environment as stated previously
4. Once activated, type **pip install -r /path/to/requirements.txt** and press enter. This will install all the required packages and dependencies for the project.
5. You can now run the Python code as usual.