Step-by-Step Guide: Creating a Python Virtual Environment (.venv)

This guide walks you through creating a Python virtual environment named **.venv** using the built-in **venv** module. You will also create a sample project folder, and learn how to activate and deactivate the virtual environment on **Windows**, **macOS**, and **Linux**.

1. Prerequisites

Before you begin, make sure you have the following:

- Python 3 installed on your system.
- Access to a terminal (Command Prompt or PowerShell on Windows, Terminal on macOS/Linux).
- Basic familiarity with running commands in a terminal.

2. Create a Sample Project Folder

1. **Choose a location** on your system where you want to keep your project (for example, your Desktop or Documents folder).
2. Create a new folder named **my_project**. This folder will contain your project files and the virtual environment.

Example commands to create the folder using a terminal:

Windows (Command Prompt or PowerShell):

```
cd %USERPROFILE%\Desktop
mkdir my_project
cd my_project
```

macOS / Linux (Terminal):

```
cd ~/Desktop
mkdir my_project
cd my_project
```

3. Create the Virtual Environment (.venv)

In your **my_project** folder, you will create a virtual environment named **.venv** using Python's **venv** module. Make sure you are inside the **my_project** directory in your terminal before running the following commands.

Check your Python version (optional but recommended):

```
python --version
```

If that does not work, try:

```
python3 --version
```

Create the virtual environment named .venv:

Windows:

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

macOS / Linux:

```
python3 -m venv .venv
```

After running this command, a new folder named **.venv** will be created inside **my_project**.

4. Activate the Virtual Environment

Once the virtual environment is created, you must **activate** it before installing packages or running Python for this project. Activation adjusts your environment so that the **python** and **pip** commands refer to the virtual environment instead of the system-wide Python.

4.1 Windows: Activate .venv

1 Make sure you are in the **my_project** folder.
2 Run the activation script depending on the shell you are using.

Command Prompt (cmd.exe):

```
.venv\Scripts\activate.bat
```

PowerShell:

```
.venv\Scripts\Activate.ps1
```

After activation, your prompt should change to show the name of the environment, for example: **(.venv) C:\Users\YourName\Desktop\my_project>**

4.2 macOS / Linux: Activate .venv

1 Make sure you are in the **my_project** folder.
2 Run the following command to activate the virtual environment:

Bash, Zsh and most common shells:

```
source .venv/bin/activate
```

After activation, your prompt should change to show the name of the environment, for example: **(.venv) yourname@machine:~/Desktop/my_project$**

5. Using the Virtual Environment

With the virtual environment activated, any Python packages you install using **pip** will be installed inside the **.venv** folder, isolated from your system Python.

Example: upgrade pip and install a package

```
python -m pip install --upgrade pip
python -m pip install requests
```

6. Deactivate the Virtual Environment

When you are finished working in your project for now, you should **deactivate** the virtual environment. This returns your terminal session to using the system-wide Python.

While the environment is active (you see **(.venv)** in your prompt), run:

```
deactivate
```

This command is the same on **Windows**, **macOS**, and **Linux**. After deactivation, the **(.venv)** prefix will disappear from your prompt.

7. Quick Reference Summary

Here is a quick summary of the key commands for creating, activating, and deactivating a **.venv** virtual environment inside a **my_project** folder.

Create project folder (common examples):

```
mkdir my_project
cd my_project
```

Create virtual environment:
Windows:

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

macOS / Linux:

```
python3 -m venv .venv
```

Activate virtual environment:
Windows (Command Prompt):

```
.venv\Scripts\activate.bat
```

Windows (PowerShell):

```
.venv\Scripts\Activate.ps1
```

macOS / Linux:

```
source .venv/bin/activate
```

Deactivate virtual environment (all platforms):

```
deactivate
```

You now have a complete, isolated Python environment for your project using a .venv folder.