Quickstart Guide: Using Python Virtual Environments (venv)

Version 1.2 | September 2025 | Prepared by Tiffany Smith

[Maintained by IT Security Team]

Introduction

This Quickstart guide was prepared by the IT team to help employees use **Python** safely and consistently. While **Python** is a powerful tool, installing **packages** without a **virtual environment** (shortened to "**venv**") can cause conflicts that affect other projects or even company systems.

By following this guide, you'll learn how to create a safe "workspace" for each project. This reduces errors, improves collaboration, and ensures company data remains secure.

Purpose

This guide shows employees how to safely run **Python** scripts by using **virtual environments** (**venv**). A **virtual environment** keeps your project's tools and libraries separate so that one project doesn't accidentally affect or break another.

In concept, a **venv** functions much like a toolbox for a particular project. Instead of keeping all tools for all uses in one place, the **venv** ensures each project is paired with its specific, applicable toolbox.

When to Use This Guide

- Use a **virtual environment (venv)** whenever you:
    a. Need to run a **Python** script for work.
    b. Install new **Python packages** for a project.
    c. Work on multiple projects that might use different versions of the same **package**.

Setting Up Your Python VENV

Step 1: Create a Virtual Environment

Go to your project folder in the terminal, then run:

```
python3 -m venv venv
source venv/bin/activate   # Mac/Linux
venv\Scripts\activate      # Windows
```

This creates a folder named **venv** that holds your project's "toolbox."

Step 2: Turn On the Environment

Before running scripts, activate the environment:

Mac/Linux:

```
bash
source venv/bin/activate
```

Windows (PowerShell)

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

If successful, you'll see (**venv**) appear at the start of your command line.

Step 3: Install Packages

While the environment is active, install **packages** like this:

```
bash
pip install pandas
```

Everything stays inside the project's toolbox — not on your computer as a whole.

Step 4: Turn Off the Environment

When you're done, type:

```
bash
deactivate
```

This safely closes the toolbox.

Best Practices

- Specify "python3" when calling **Python** to run a script and avoid using "python" on its own. This calls the correct version of **Python** which ensures project stability.
- Setup only one environment per project to avoid software conflicts.
- Never email or upload your **venv** folder. **Python** has a built-in way to share **venv's**!
    - Instead, share your requirements file with:

```
bash
pip freeze > requirements.txt
```

    - Colleagues can then re-create your **venv** setup using:

```
bash
pip install -r requirements.txt
```

Troubleshooting (Common Fixes)

- **Command not found: python3:**
    - Python isn't installed or isn't on your PATH. Contact IT.

```
python3 app.py
zsh: command not found: python3
```

- **No module named venv:**
    - Ask IT to install the Python **venv** package.

```
python3 -m venv myenv
ModuleNotFoundError: No module named 'venv'
```

- **Still not working?**
    - Check if (**venv**) is at the start of your command line.
        - For Linux/Mac: source **venv**/bin/activate
        - Windows (Command Prompt): **venv**\Scripts\activate.bat
        - Windows (PowerShell): **venv**\Scripts\Activate.ps1
        - Try activating again with the correct command for your OS

Glossary (Plain Language)

Python: A programming language designed to be easy to read and use.

Virtual environment (venv): A private copy of Python + tools for one project.

Package: A ready-made set of Python code that adds features (like Excel add-ins).

bash: A common terminal (command-line tool) used on Mac and Linux.

PowerShell: The Windows version of a terminal (command-line tool).

requirements.txt: A simple text file listing all the packages your project uses.

Revision History

V1.0 (August 2025) - Initial version

V1.1 (September 2025) - Revised for clarity, error screenshots added

V1.2 (September 2025) - Removed screenshots and replaced with code blocks