# **Python venv command:**

To create environment

* **Python3 -m venv** “name of the virtual environment”

To activate environment :

* **Python3 venv/bin/activate**

To deactivate environment:

* **Deactivate**

To put all requirements in one file and installing it :

* **Pip freeze > requiremnts.txt**
* **Pip installs -r requirements.txt**

# **Git commands:**

**📌 1. Initial Setup (first time only)**

**Everyone (lead + collaborators):**

* git config --global user.name "Your Name"
* git config --global user.email "your@email.com"
* git config --list

✅ This ensures GitHub knows who made each commit.

**📌 2. Project Initialization**

**Project Lead (you):**

**# Create repo locally**

* mkdir project-name && cd project-name
* git init
* echo "# Project Name" >> README.md
* git add README.md
* git commit -m "Initial commit"

**# Create repo on GitHub (via web or CLI), then link:**

* git remote add origin https://github.com/your-username/project-name.git
* git branch -M main
* git push -u origin main

**Collaborators:**

* git clone https://github.com/your-username/project-name.git
* cd project-name

**📌 3. Daily Workflow for Collaborators**

**(A) Always update main before working:**

* git checkout main
* git pull origin main

**(B) Create a feature branch:**

* git checkout -b feature/feature-name

**(C) Work on files, then save progress:**

* git add .
* git commit -m "Added new feature"

**(D) Push branch to GitHub:**

* git push origin feature/feature-name

**(E) Create Pull Request (PR) on GitHub:**

* Go to repo → “Pull Requests” → “New Pull Request”.
* Assign to **you (the lead)** for review.

**📌 4. Project Lead Responsibilities**

**(A) Review PR on GitHub**

* If ✅ approve → merge into main.
* If ❌ changes requested → collaborator revises branch.

**(B) To merge locally (if you prefer CLI):**

* git checkout main
* git pull origin main
* git merge feature/feature-name
* git push origin main

**📌 5. Revising Code After Review (Collaborators)**

If you reject a PR, the collaborator:

**# Go back to their branch**

* git checkout feature/feature-name

**# Make fixes**

* git add .
* git commit -m "Fix review comments"

# Push again (same branch)

git push origin feature/feature-name

🔁 GitHub will **update the same pull request** automatically with their fixes.

**📌 6. Getting a Teammate’s Branch**

If a collaborator wants to test or help on another branch:

* git fetch origin
* git checkout -b feature/teammate-feature origin/feature/teammate-feature