

PythonAnywhere Git Reference Guide

This guide covers the essential Git commands for managing your project lifecycle on PythonAnywhere.

1. Initial Setup (One-time only)

If you haven't connected your project yet, use these to bring your code over for the first time.

- **Clone the repository:**

```
git clone https://github.com/your-username/your-repo-name.git
```

- **Set up your identity:**

```
git config --global user.email "you@example.com"
```

```
git config --global user.name "Your Name"
```

2. Standard Update Workflow

Use these commands when you have made changes in VS Code and want to update the live site.

Action	Command
Download latest changes	git pull origin main
Fetch new tags/releases	git fetch --tags
Switch to a specific version	git checkout v6.0
Switch back to the main line	git checkout main

3. Version Tracking & Inspection

Use these to verify which version of the code is currently "live."

- **Check current status:** (Tells you if you are in a "detached HEAD" state or on a branch).`git status`

- **List all available tags:**
`git tag`
 - **Verify current tag version:**
`git describe --tags`
 - **View recent history:** (Shows the last 5 commits for a quick check).`git log --oneline -n 5`
-

4. Troubleshooting Commands

If things get messy on the server, these help reset the environment.

- **Discard local changes:** (Useful if you accidentally edited a file on the server and is failing).`git reset --hard HEAD``git pull`
 - **Force pull from GitHub:**
`git fetch --all`
`git reset --hard origin/main`
-

5. The "PythonAnywhere Final Step"

Remember: Git only updates the files on the disk. To update the actual website, you must:

1. Log into the **PythonAnywhere Dashboard**.
 2. Go to the **Web Tab**.
 3. Click the **Reload** button.
-

Pro-Tip for your Workflow:

Since you are using **Tags (v6.0)**, your standard update command should always be:

1. `git fetch --tags`
2. `git checkout v[NewVersionNumber]`
3. **Reload Web App.**