

Git Cheatsheet

AI-Assisted Programming for PhD Researchers

One page of the Git and uv commands you'll actually use during the labs. Keep this open in a second window; you don't need to memorize any of it.

Identity

Set this once per machine, before your first commit.

Command	What it does
<code>git config --global user.name "Your Name"</code>	Sets the name attached to your commits
<code>git config --global user.email "you@example.com"</code>	Sets the email attached to your commits

Everyday loop

The cycle you repeat constantly: check status, stage, commit.

Command	What it does
<code>git init</code>	Turns the current folder into a new Git repository
<code>git clone URL</code>	Copies a remote repository to your machine
<code>git status</code>	Shows what's changed, staged, and untracked
<code>git add <file></code>	Stages a file's changes for the next commit
<code>git add -p</code>	Stages changes interactively, hunk by hunk
<code>git commit -m "message"</code>	Records the staged changes as a new commit

Looking around

Inspect history and changes without modifying anything.

Command	What it does
<code>git log --oneline --graph</code>	Compact, one-line-per-commit history with branch graph
<code>git diff</code>	Shows unstaged changes in the working directory
<code>git diff --staged</code>	Shows changes that are staged for the next commit
<code>git show HEAD</code>	Shows the full commit at <code>HEAD</code> (message + diff)

Undoing

Safe first, destructive last: these go roughly in order of risk.

Command	What it does
<code>git restore <file></code>	Discards unstaged changes to a file, back to the last commit
<code>git restore --staged <file></code>	Unstages a file, keeping the edits in your working directory
<code>git reset --soft HEAD~1</code>	Undoes the last commit but keeps its changes staged

Branches

Command	What it does
<code>git switch -c <name></code>	Creates a new branch and switches to it
<code>git switch <name></code>	Switches to an existing branch
<code>git merge <name></code>	Merges <code><name></code> into the current branch
<code>git branch -d <name></code>	Deletes a branch (only if it's already merged)

Tags & checkpoints

Tags mark a specific commit, useful for “this is what I submit” or “this is the state before the lab.”

Command	What it does
<code>git tag -a <tag> -m "message"</code>	Creates an annotated tag on the current commit

Command	What it does
<code>git fetch --tags</code>	Downloads tags from the remote that you don't have locally
<code>git checkout tags/<tag> -b <branch></code>	Checks out a tag into a new branch (tags themselves are detached)

Worktrees

A worktree lets you have two branches checked out **at the same time**, in separate folders, handy for running an AI agent on one branch while you keep working on another.

Command	What it does
<code>git worktree add ../wt-branch <branch></code>	Checks out <branch> into a new sibling folder ../wt-branch
<code>git worktree list</code>	Lists all worktrees attached to this repository
<code>git worktree remove <path></code>	Removes a worktree (the branch itself is untouched)

.gitignore patterns

List files and folders Git should never track: build artifacts, virtual environments, secrets. One pattern per line.

Pattern	Matches
<code>__pycache__/</code>	Any folder named <code>__pycache__</code> , anywhere in the repo
<code>*.pyc</code>	Any file ending in <code>.pyc</code>
<code>.venv/</code>	A local virtual environment folder
<code>.env</code>	A local environment/secrets file
<code>data/raw/*.csv</code>	CSVs directly under <code>data/raw/</code> (but not subfolders)

Python / uv

[uv](#) manages Python and project dependencies for every lab. See the [Setup](#) page for installation.

Command	What it does
<code>uv --version</code>	Confirms uv is installed and prints its version

Command	What it does
<code>uv init --lib</code>	Creates a new Python library project (with <code>pyproject.toml</code>)
<code>uv add --dev <package></code>	Adds <code><package></code> as a development dependency
<code>uv run <command></code>	Runs <code><command></code> inside the project's managed environment
<code>uv run pytest</code>	Runs the test suite inside the project's managed environment

Bibliography