NAWI National Alliance for Water Innovation

GitHub 101 for WaterTAP Academy

Principles and Best Practices

WaterTAP Academy 2025

Outline

GitHub Basics

- What is GitHub and Git?
- Terminology & common commands

Git Basics:

- Downloading the latest material
- Tracking and sharing file changes
- Working on a local branch

What is GitHub and Git

GitHub - Collaborative Development

- Cloud-based code hosting service
- Helps manage git repositories
- Provides a convenient review process
- Users can report bugs by opening issues

Git - Version Control

- Version control for source code
- No need to create multiple copies locally
- Easy to work on independent projects without tampering source code

GitHub/Git Terminology

Term	Meaning
Repository or repo	Project folder
Issue	Bug report or feature request
Pull Request or PR	Request to merge your contribution to the project
Branch	Name given to a set of changes to the repo
Main*	By convention, the name of the default branch
Fork	Copy of repo on your GitHub account
Merge	Means to bring changes from one branch into another
Clone	Get a local copy of a repository

** - Also called master but main is the recommended name by GitHub*

Common Commands

Command	Function
`git clone <link>`	Clone a repo locally
`git status`	Displays the status of the current branch you are in
`git branch`	Displays all the current branches on your local machine
`git remote -v`	Displays where your remotes are pointed to on GitHub
`git merge <branch name>`	Merge current local branch with another branch
`git checkout <branch name>`	Move from current branch to other NOTE: Git complains unless all changes have been committed in the current branch
`git pull <remote name> <branch name>`	Pulls from a particular branch on a remote.
`git push <remote name> <branch name you want to push>:<new name>`	Push a particular local working branch to a remote.
`git config --global core.editor "nano"`	Sets Nano as your default text editor for Git commands that require user input; "vim" can be used in place of "nano" but is less intuitive.

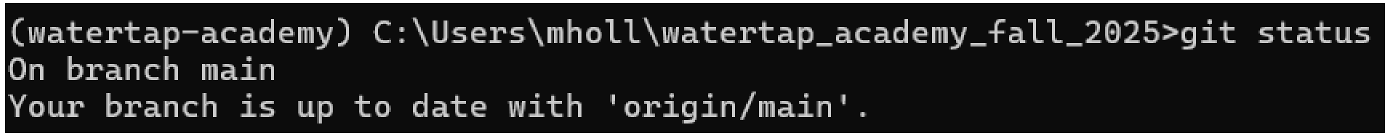
Downloading the Latest Material

To ensure your version of the WaterTAP Academy repo is up to date, go through the following steps:

- Verify that you are on the main branch:
  - git status → will display “On branch main”
- If you are not on the main branch, switch to it and check the status:
  - git checkout main → git status

NOTE: You may need to commit changes to your current local branch before switching

- If your branch is up to date, you should see:
  - “Your branch is up to date with origin/main”



```
(watertap-academy) C:\Users\mholl\watertap_academy_fall_2025>git status
On branch main
Your branch is up to date with 'origin/main'.
```

- If your branch is not up to date:
  - “This branch is behind origin/main by X commits and can be fast-forwarded”
  - git pull origin main → will update your branch to be in-line with the public repository

Tracking and Sharing File Changes

Step 1: Create a local working branch

- `git branch <branch name>`

Step 2: Checkout to your local working branch

- `git checkout <branch name>`

Pro-tip: Combine steps 1 & 2:
`git checkout –b <branch name>`

Step 3: Check the status (always recommended)

- `git status` → will tell you that you are now on branch <branch name>

Step 4: Make changes to files

- `git add <filepath>` → Git will track and save all the changes to this file

Pro-tip: `git status` will display all the modified files

Step 5: Commit the changes and summarize with a message

- `git commit –m <message>` → Commit messages will be publicly displayed on the PR

Step 6: Push the changes to the main branch

- `git push myfork-academy <branch name>` → creates a link for a public PR

Working on a Local Branch

- Assume you have a new file → new.py that you want to commit
- On your local working branch:
  - git status → should show “new.py” in untracked files
- Add the file to be tracked by Git:
  - git add new.py *(Note: you should be in that folder)*
- **Carefully, check and add each file and do not add all files at once.**
- Do a check on what was added
  - git status
- Commit the file
  - git commit –m “your commit message”
- Check the status again:
  - git status → should tell you “nothing to commit, working directory clean”
- If you made multiple changes, you will see multiple files listed under untracked files

NOTE: Never execute git add --all. Always add one file and know which file is being added.

Helpful GitHub Resources

- Simple tutorial from GitHub: https://guides.github.com/activities/hello-world/
- GitHub workflow explained: https://guides.github.com/introduction/flow/
- Forking projects: https://guides.github.com/activities/forking/
- GitHub Docs: https://docs.github.com/en
- Git handbook: https://guides.github.com/introduction/git-handbook/
- GitHub's official YouTube channel for video guides:
  - https://www.youtube.com/githubguides
- Highly recommended tutorials:
  - https://www.atlassian.com/git/tutorials

Thank you

This material is based upon work supported by the National Alliance for Water Innovation (NAWI), funded by the U.S. Department of Energy, Office of Energy Efficiency and Renewable Energy (EERE), Industrial Technologies Office (ITO), under Funding Opportunity Announcement DE-FOA-0001905.

- **National Energy Technology Laboratory:** Adam Atia, Elmira Shamlou, Hunter Barber, Marcus Holly, Chenyu Wang, Alejandro Garciadiego, Maojian Wang, Carolina Tristan
- **Lawrence Berkeley National Laboratory:** Dan Gunter, Keith Beattie, Ludovico Bianchi, Xiangyu Bi, Oluwamayowa Amusat, Michael Pesce
- **National Renewable Energy Laboratory:** Kurby Sitterley, Kinshuk Panda, Zach Binger, Mukta Hardikar, Ben Knueven
- **Oak Ridge National Laboratory:** Johnson Dhanasekaran, Fahim Abdullah, Kris Villez, Srikanth Allu
- **SLAC National Accelerator Laboratory**: Alex Dudchenko

Disclaimer: *This presentation was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government, nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors, or their employees, make any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government, any agency thereof, or any of their contractors or subcontractors. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government, any agency thereof, or any of their contractors. Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC., a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA-0003525.*

Contact Us!

NAWI

For general inquiries, collaborations, etc.:

- **watertap-contact@lbl.gov**

For user support on WaterTAP:

- **watertap-support@lbl.gov**

For detailed coding questions, open an issue or discussion on GitHub:

- **https://github.com/watertap-org/watertap_academy_fall_2025**

WaterTAP Documentation:
https://watertap.readthedocs.io/en/stable/