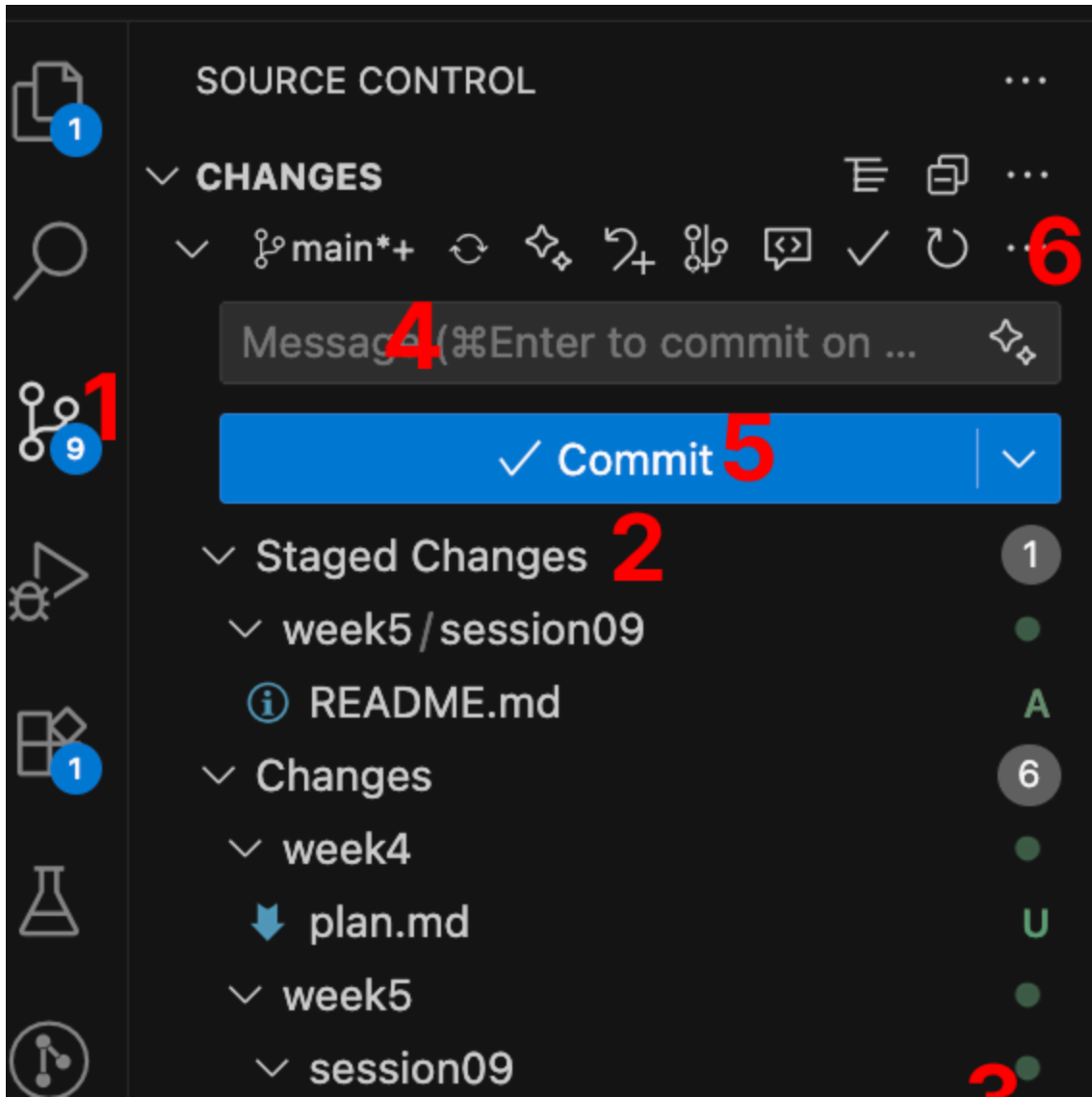


Git & GitHub: Advanced Topics

Quick Review: Git Basics (from week 1)

- `git status`
- `git add`
 - `git add <filename>` adds a specific file
 - `git add .` adds all changed files
- `git commit`
 - `git commit -m "your message"` commits changes with a message
- `git push`
 - `git push origin main` pushes changes to the main branch on the origin remote

Git Basics from the VS Code GUI



Git Branches

- A branch is a separate line of development with a history of commits
- The default branch is usually called `main` (or sometimes `master`)
- You can create new branches to work on features or fixes without affecting the main codebase
- Common commands:
 - `git branch <branch-name>` : Create a new branch
 - `git checkout <branch-name>` : Switch to a different branch
 - `git checkout -b <branch-name>` : Create and switch to a new branch
 - `git merge <branch-name>` : Merge changes from another branch into the current branch
- To merge a second branch into the current working directory, there must be:
 - no uncommitted changes in the working directory

Working with Multiple Remotes

```
@presnick → /workspaces/course-labs-and-other-resources-presnick (main) $ git remote -v
origin  https://github.com/umsi211F2025/course-labs-and-other-resources-presnick (fetch)
origin  https://github.com/umsi211F2025/course-labs-and-other-resources-presnick (push)
upstream      https://github.com/umsi211F2025/umsi211f2025-classroom-00a9d4-course-labs-and-other-resources-umsi211-f25-course-resources.git (fetch)
upstream      https://github.com/umsi211F2025/umsi211f2025-classroom-00a9d4-course-labs-and-other-resources-umsi211-f25-course-resources.git (push)
```

- A remote is a version of your repository hosted somewhere else (e.g., GitHub)
- You can have multiple remotes for a single local repository
 - in this case, `origin` is my personal GitHub repo and `upstream` is the class GitHub repo

Pulling from Different Remotes

- `git pull <remote> <branch>` fetches and merges changes from the specified remote
 - e.g., `git pull upstream main`
 - fetches the `main` branch of the `upstream` remote, creating a local copy of it as a branch called `upstream/main`
 - then merges that branch into your current working directory
- This is a less mysterious way to get update from the class repo than our `./sync-upstream.sh` script

Example: Pulling from Upstream Remote



The screenshot shows a code editor interface with several tabs at the top: 'Preview README.md', '\$ sync-updates.sh', 'html_intro.md M', 'Preview html_intro.md', '<> poll_static.html M', '<> poll_static_with_css.html M', and 'MERGE_MSG X'. The 'MERGE_MSG' tab is active, displaying a merge message template with line numbers 1 through 7. Below the editor, a terminal window shows the execution of git commands to configure pull behavior and perform a pull.

```
.git > MERGE_MSG
1 Merge branch 'main' of https://github.com/umsi211f2025/umsi211f2025-classroom-00a9d4-course-labs-and-other-resources-umsi211-f25-course-resources
2 # Please enter a commit message to explain why this merge is necessary,
3 # especially if it merges an updated upstream into a topic branch.
4 #
5 # Lines starting with '#' will be ignored, and an empty message aborts
6 # the commit.
7

@presnick → /workspaces/course-labs-and-other-resources-presnick (main) $ git config pull.rebase false
@presnick → /workspaces/course-labs-and-other-resources-presnick (main) $ git pull upstream main
From https://github.com/umsi211f2025/umsi211f2025-classroom-00a9d4-course-labs-and-other-resources-umsi211-f25-course-resources
* branch      main      -> FETCH_HEAD
hint: Waiting for your editor to close the file... [ ]
```

Important Notes

- One time you need to run `git config pull.rebase false` to set the default pull behavior to "merge" changes with any changes you have.
 - After that, it should know. You might have to do this again if your codespace resets.

What Can Go Wrong with Git Pull?

- If you have uncommitted changes to a file that would be overwritten, `git pull` may fail with an error message.
 - I recommend committing your changes before pulling.
 - There is another option called stashing that can help with this, but it's another thing to learn and I don't recommend it for beginners.
- If you have committed changes to a file that was also changed in the upstream repo, git will try to merge the changes within the file, but if there are conflicting changes, you will get a merge conflict.

Resolving Merge Conflicts

This YouTube video explains it really well: <https://www.youtube.com/watch?v=DloR0BOGNU0>

Be sure you watch it and understand it. You can always ask your copilot to help you understand any vocabulary that's tripping you up.