Version Control, IDEs, and Documentation

Using Git and Markdown in VSCode

Kristine Lacek
US CDC – Influenza Division

Disclaimer

The findings and conclusions in this presentation are those of the authors and do not necessarily represent the official position of the Centers for Disease Control and Prevention.
Use of trade names and commercial sources is for identification only and does not imply endorsement by the U.S. Department of Health and Human Services.
References to non-CDC sites on the Internet do not constitute or imply endorsement of these organizations or their programs by CDC or the U.S. Department of Health and Human Services. CDC is not responsible for the content of pages found at these sites.

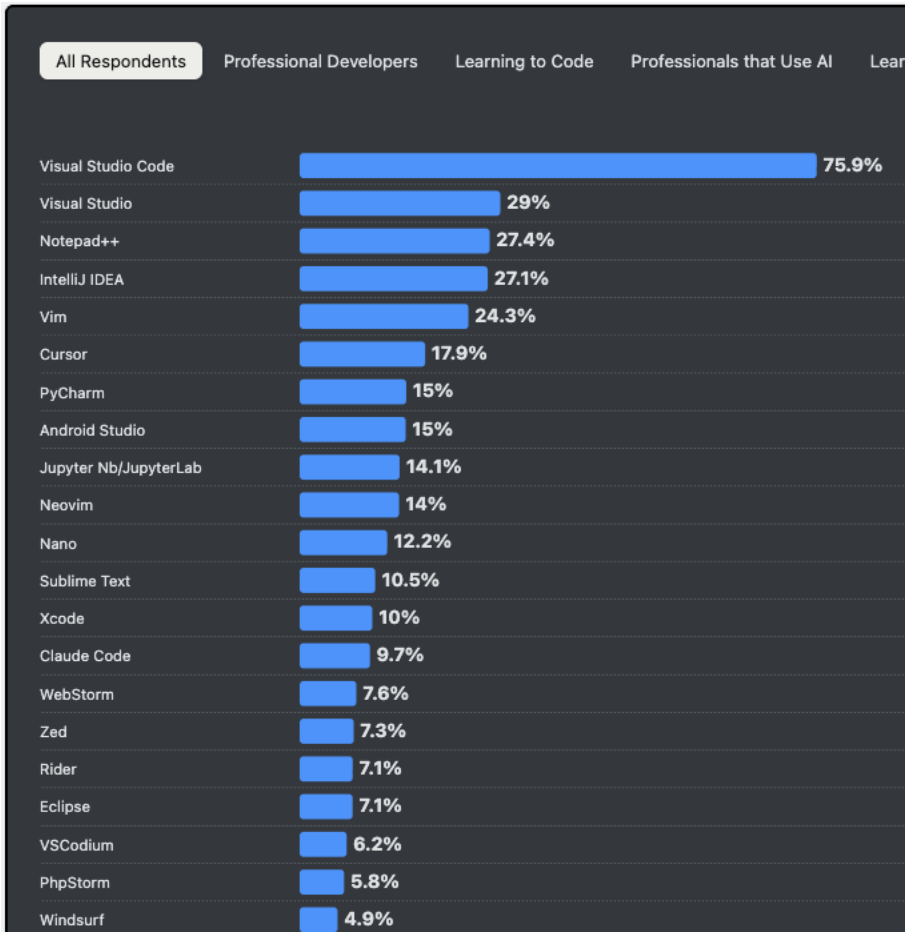
Module Objectives

- Introduce VSCode
- Introduce version control and git
- Introduce Markdown for note taking
- Distribute Markdown cheat sheet
- Windows users: install WSL
- Install VSCode

VSCode

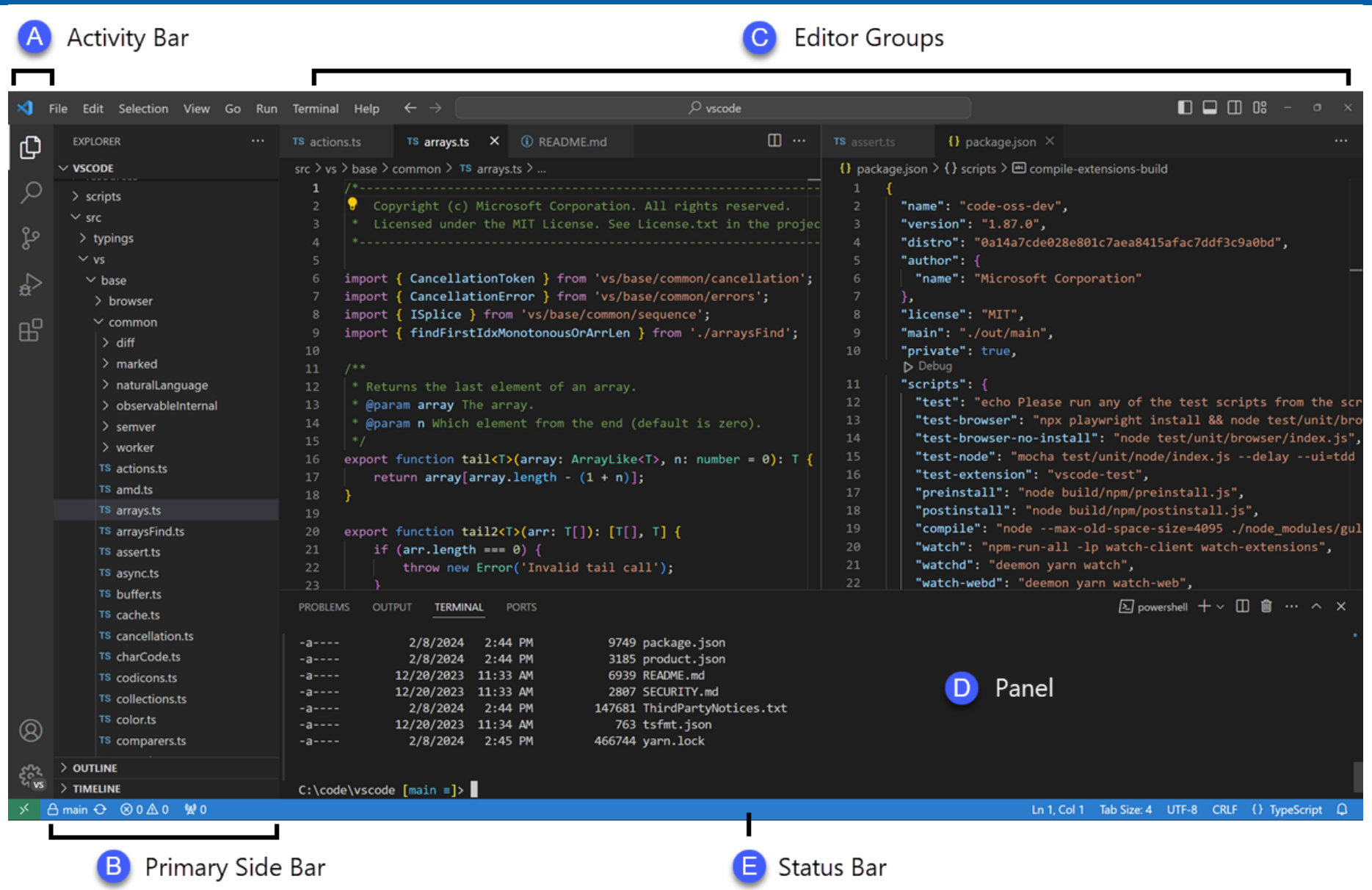
- Professional programmers often use an “Integrated Development Environment” (**IDE**)
  - Write code, browse your files, and even run your program all in the same window

- **Visual Studio Code (VSCode):** free application from Microsoft
  - Runs on Windows, Mac, Linux—Windows users will run on WSL
  - Supports all popular coding languages
  - Additional plugins for expanded functionality
  - As you write code, VS Code offers suggestions, catches errors, and highlights your code in color to make it easier to read and understand



https://survey.stackoverflow.co/2025/technology#1-dev-id-es

VSCode


A Activity Bar
C Editor Groups
File Edit Selection View Go Run Terminal Help
vscode
EXPLORER
VSCODE
scripts
src
typings
vs
base
browser
common
diff
marked
naturalLanguage
observableInternal
semver
worker
actions.ts
amd.ts
arrays.ts
arraysFind.ts
assert.ts
async.ts
buffer.ts
cache.ts
cancellation.ts
charCode.ts
codicons.ts
collections.ts
color.ts
comparers.ts
OUTLINE
TIMELINE
actions.ts
arrays.ts
README.md
src > vs > base > common > arrays.ts > ...
Copyright (c) Microsoft Corporation. All rights reserved.
Licensed under the MIT License. See License.txt in the projec
import { CancellationToken } from 'vs/base/common/cancellation';
import { CancellationError } from 'vs/base/common/errors';
import { ISplice } from 'vs/base/common/sequence';
import { findFirstIdxMonotonousOrArrLen } from './arraysFind';
/**
* Returns the last element of an array.
* @param array The array.
* @param n Which element from the end (default is zero).
*/
export function tail<T>(array: ArrayLike<T>, n: number = 0): T {
return array[array.length - (1 + n)];
}
export function tail2<T>(arr: T[]): [T[], T] {
if (arr.length === 0) {
throw new Error('Invalid tail call');
assert.ts
package.json
package.json > { } scripts > compile-extensions-build
{
"name": "code-oss-dev",
"version": "1.87.0",
"distro": "0a14a7cde028e801c7aea8415afac7ddf3c9a0bd",
"author": {
"name": "Microsoft Corporation"
},
"license": "MIT",
"main": "./out/main",
"private": true,
Debug
"scripts": {
"test": "echo Please run any of the test scripts from the scr
"test-browser": "npx playwright install && node test/unit/bro
"test-browser-no-install": "node test/unit/browser/index.js",
"test-node": "mocha test/unit/node/index.js --delay --ui=tdd
"test-extension": "vscode-test",
"preinstall": "node build/npm/preinstall.js",
"postinstall": "node build/npm/postinstall.js",
"compile": "node --max-old-space-size=4095 ./node_modules/gul
"watch": "npm-run-all -lp watch-client watch-extensions",
"watchd": "deemon yarn watch",
"watch-webd": "deemon yarn watch-web",
PROBLEMS OUTPUT TERMINAL PORTS
powershell
-a---- 2/8/2024 2:44 PM 9749 package.json
-a---- 2/8/2024 2:44 PM 3185 product.json
-a---- 12/20/2023 11:33 AM 6939 README.md
-a---- 12/20/2023 11:33 AM 2807 SECURITY.md
-a---- 2/8/2024 2:44 PM 147681 ThirdPartyNotices.txt
-a---- 12/20/2023 11:34 AM 763 tsfmt.json
-a---- 2/8/2024 2:45 PM 466744 yarn.lock
C:\code\vscode [main ≡]>
D Panel
main
Ln 1, Col 1 Tab Size: 4 UTF-8 CRLF TypeScript
B Primary Side Bar
E Status Bar

Git

- Git is a version control software running on your local machine
    - Tracks changes to files over time
- Designed for collaboration
    - Multiple people can work on the same codebase
    - Features like issues and milestones allow collaborators to share to-dos and track problems
    - Think of it like a shared document (Microsoft office, Google docs)
- Keeps a history of your work: see what changed, when, and why
- Common in bioinformatics
    - Pipelines, scripts, documentation, and workflows

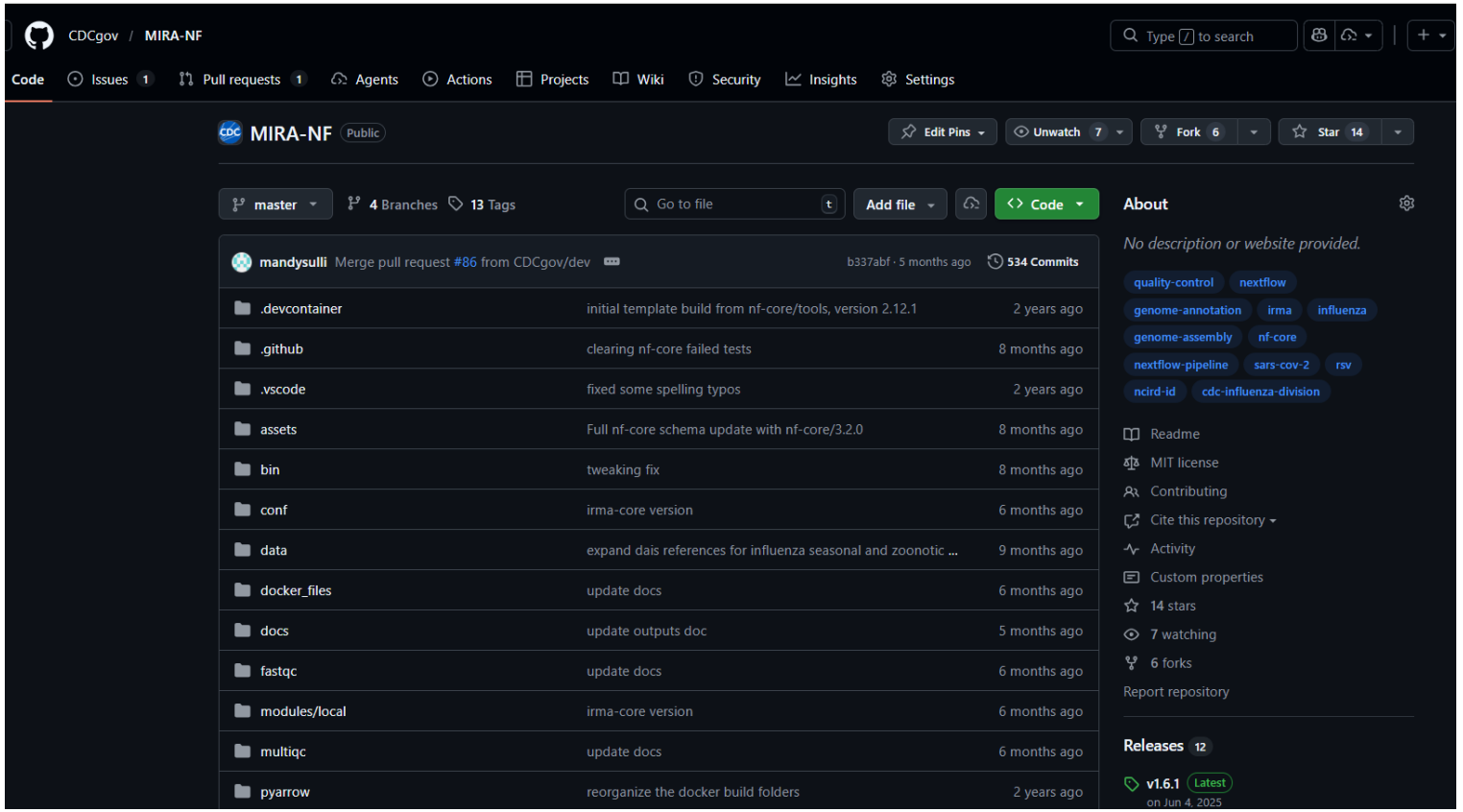
- Gitlab and Github most common *cloud repositories*

Git

Core Git Concepts

- Repository (repo)
  - A project tracked by Git
- Commit
  - A saved snapshot of changes with a message
- Branch
  - A parallel version of the code for development or testing

- Remote repository
  - Shared copy on platforms like GitHub or GitLab

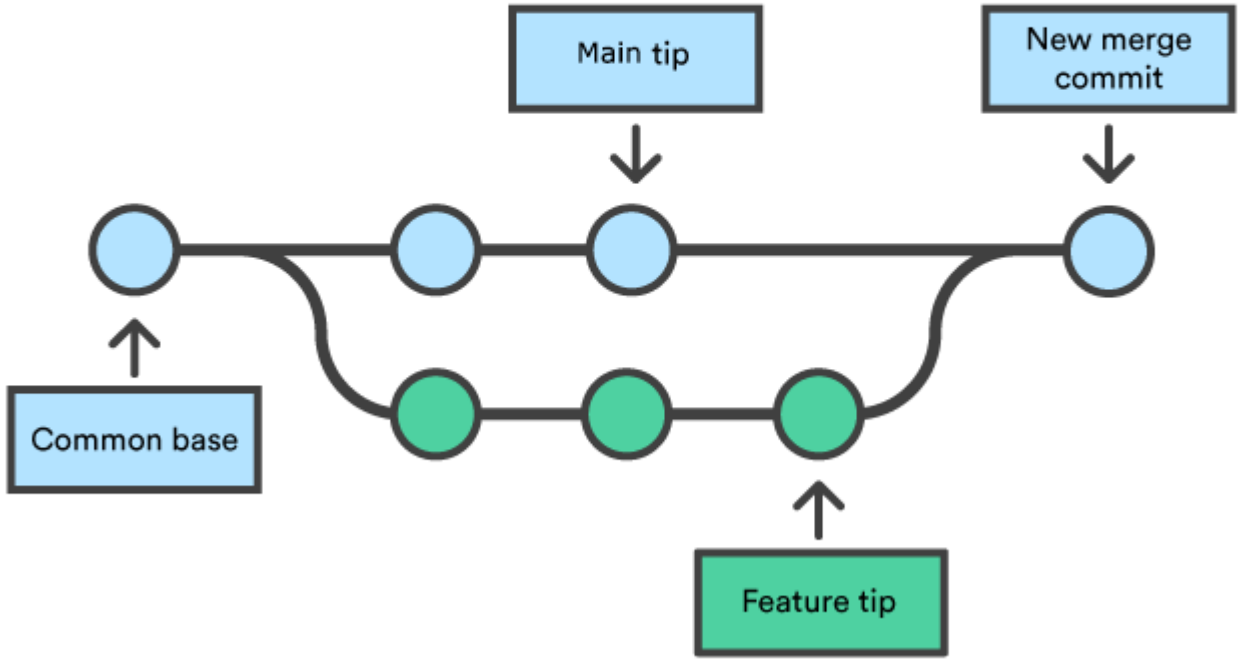



```
qgx6@rosalind02:MIRA-NF$ git status
On branch dev
Your branch is up to date with 'origin/dev'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        .MIRA_nextflow.sh

nothing added to commit but untracked files present (use "git add" to track)
```

Git

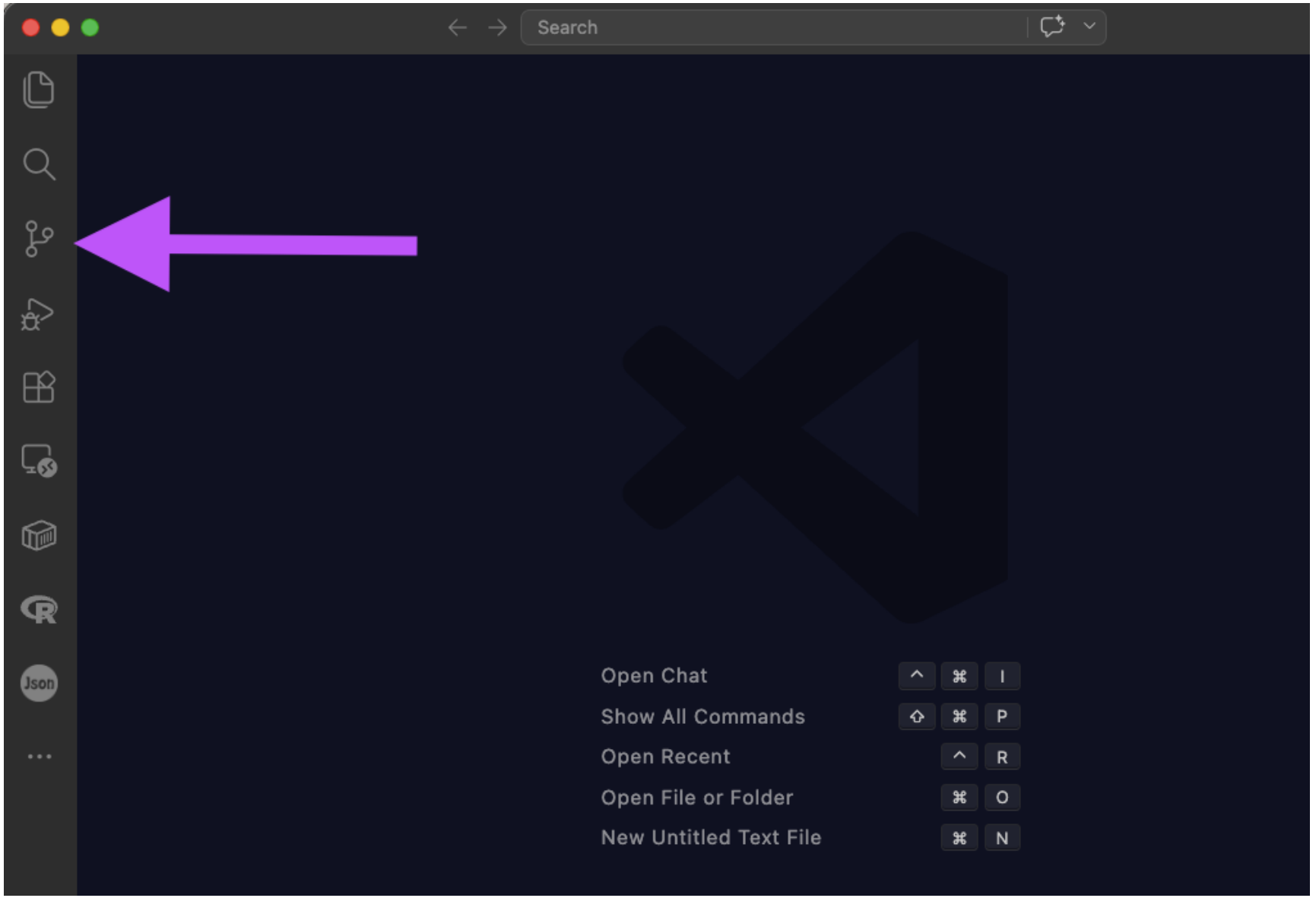
Common Git Commands

- Create or get a repository
  - git init — start a repo
  - git clone <repo> — copy an existing repo
- Track and save changes
  - git status — see changes
  - git add <file> — stage changes
  - git commit -m "message" — save snapshot
- Sync with others
  - git pull — get updates
  - git push — share your changes
- Merge or Pull request
  - Combine your changes with the “main” branch



Git in VSCode

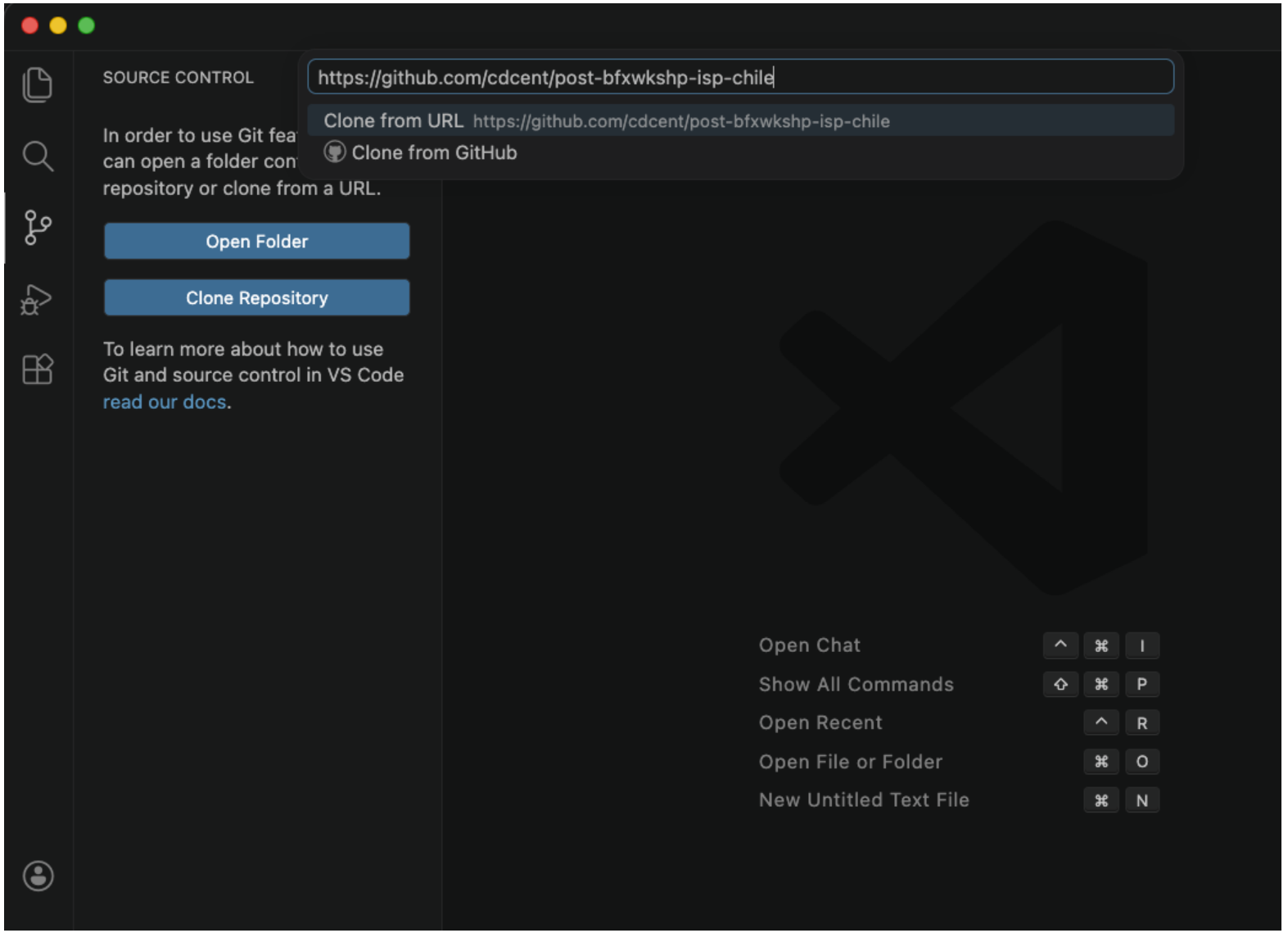
Source control tab



Git in VSCode

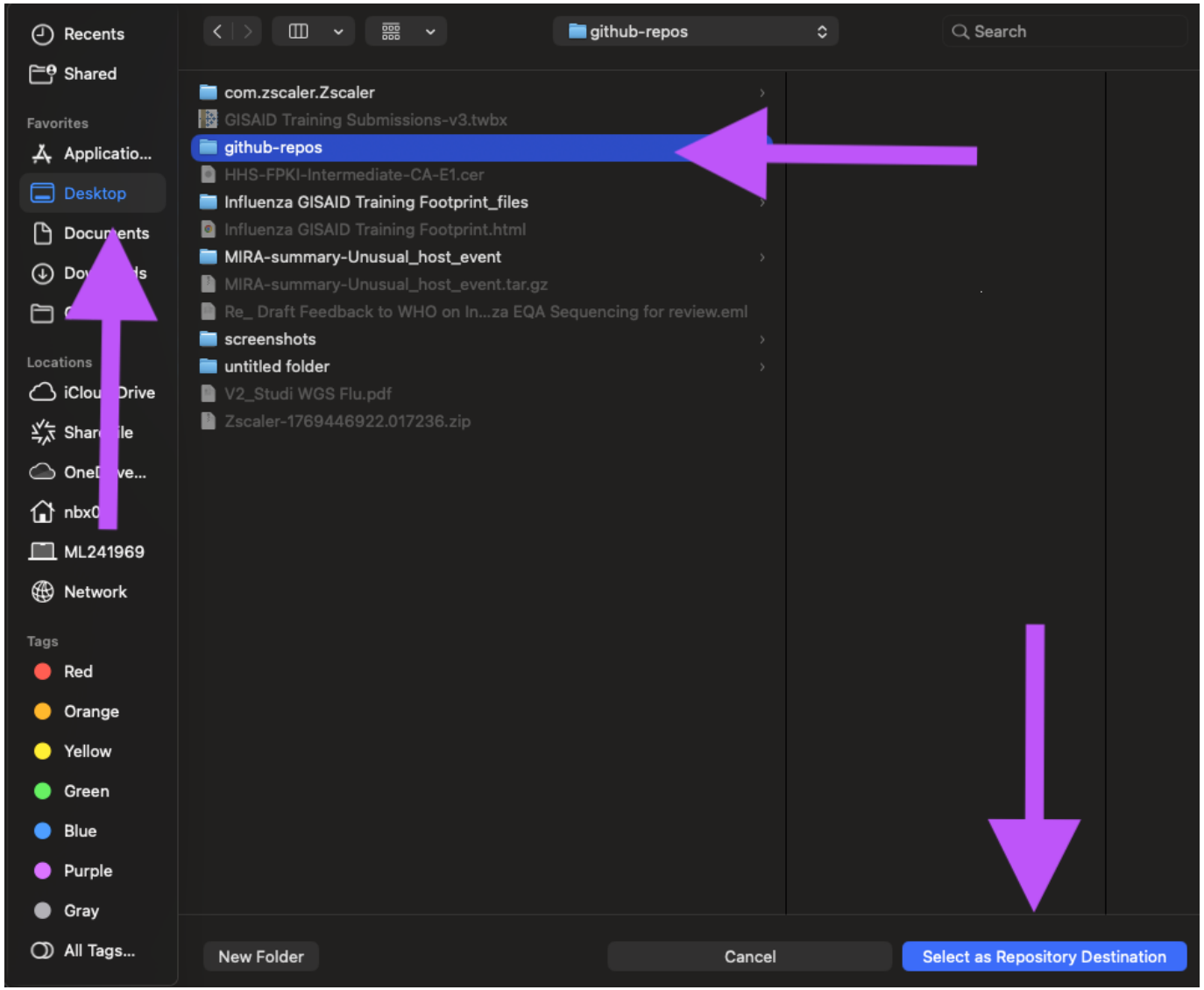
Clone repository with URL

https://github.com/nbx0/bangkok-bfx-wksp



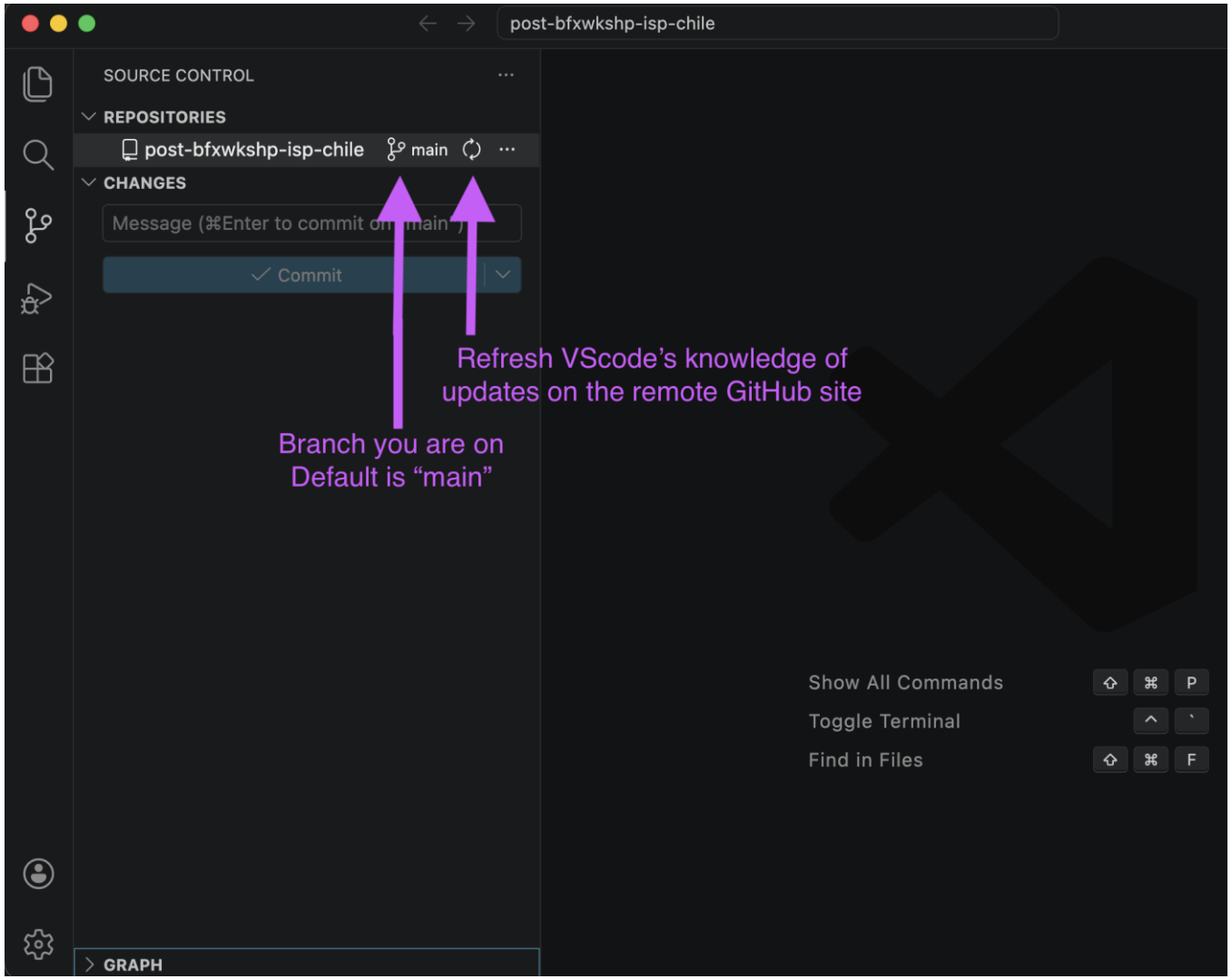
Git in VSCode

Choose a local folder to clone into and click **Select as Repository Destination**. Below, I have created a folder on my Desktop called github-repos to store all my cloned repositories.



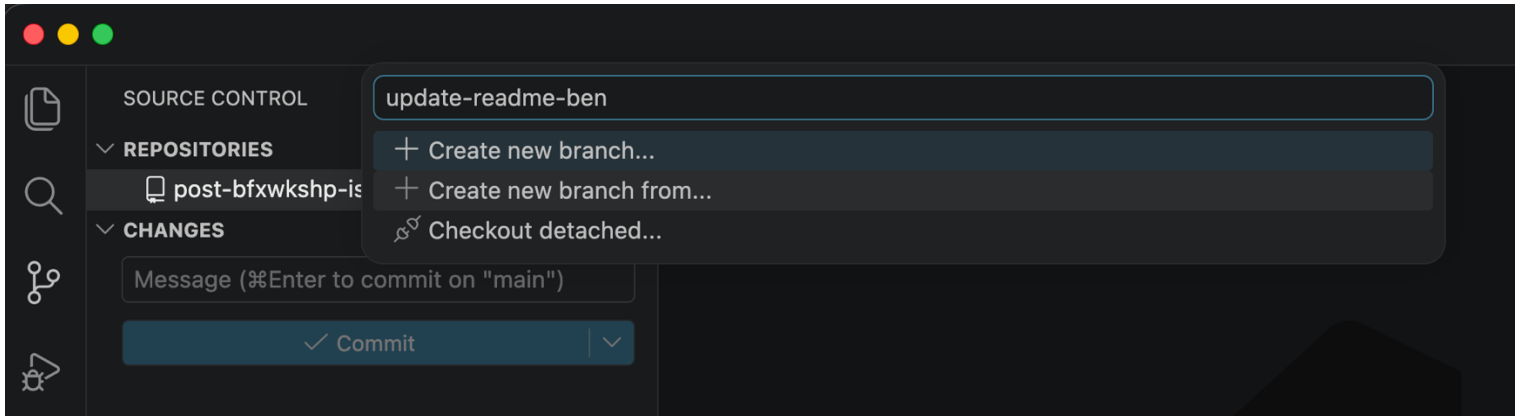
Git in VSCode

In the **Source Control** tab, click the “branch” icon beside “main” at the top of the **REPOSITORIES** panel.

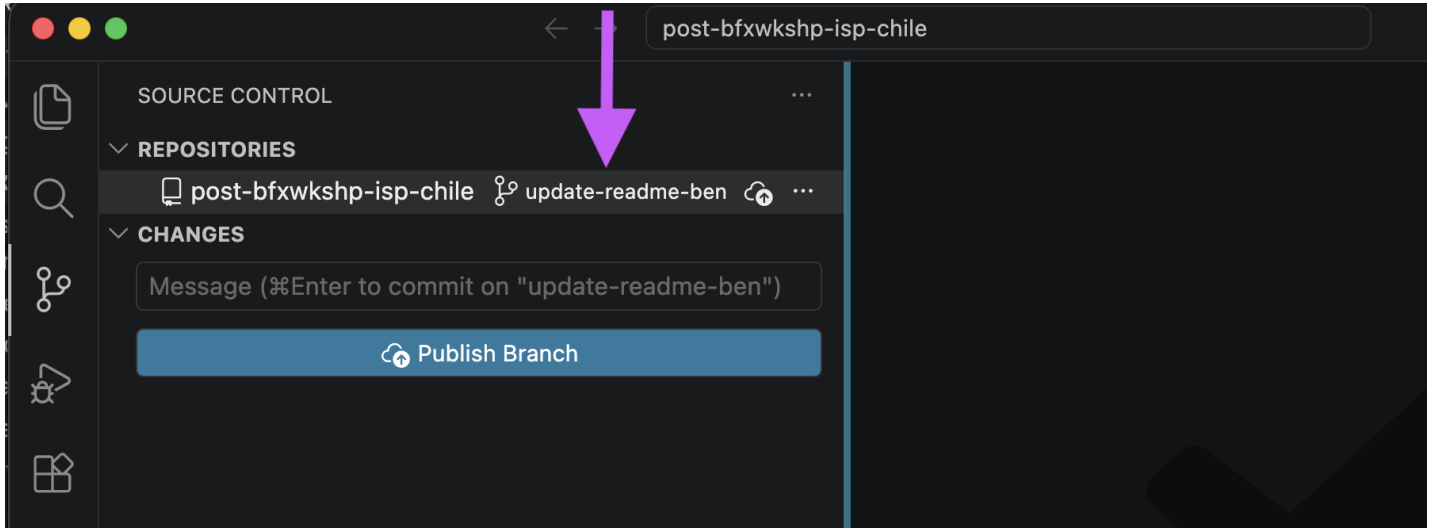


Git in VSCode

Your cursor will move to the input box at the top of the window. Type in a name for your new branch, hit Enter



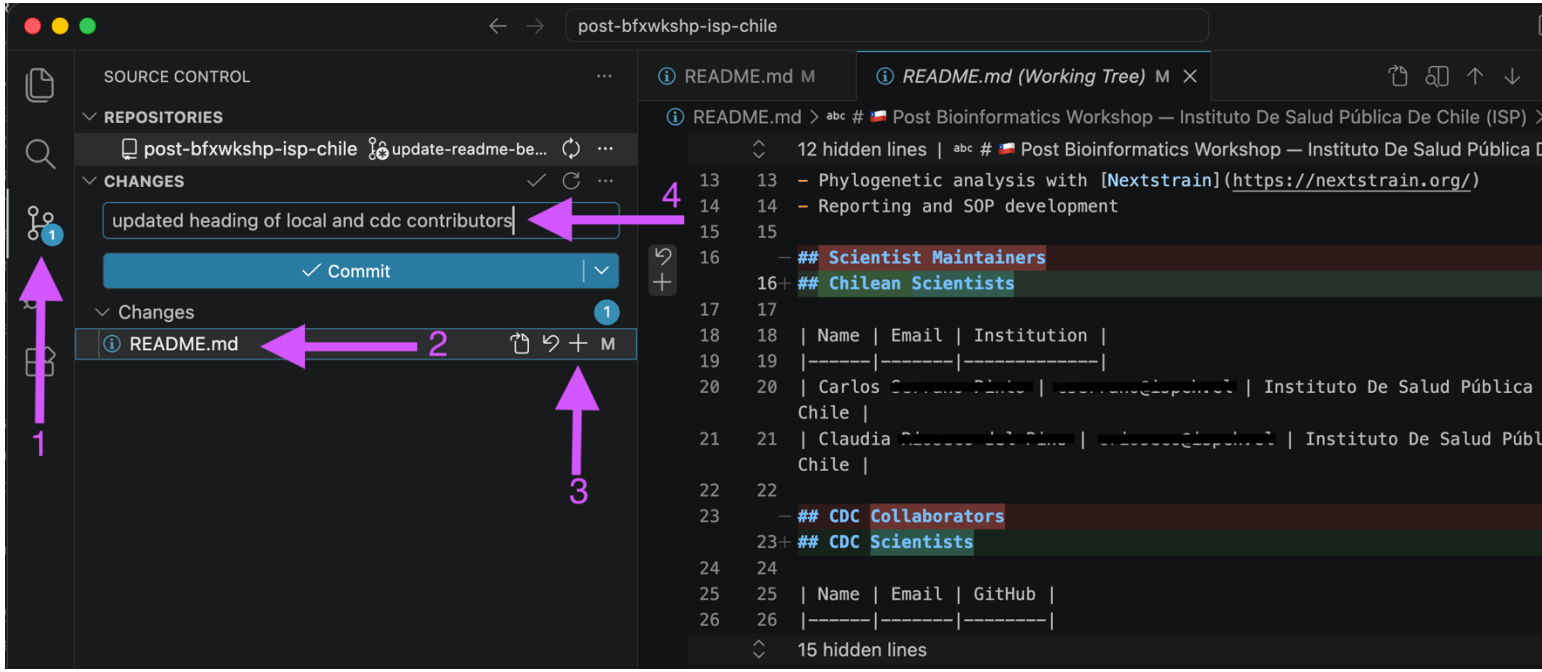
VS Code will create the branch and switch to it automatically. “Publish” the branch to GitHub by clicking the **Publish Branch** button that appears in the status bar at the bottom of the window.



Git in VSCode

After editing and saving a file in the repo via VSCode, Stage Commit, and Push

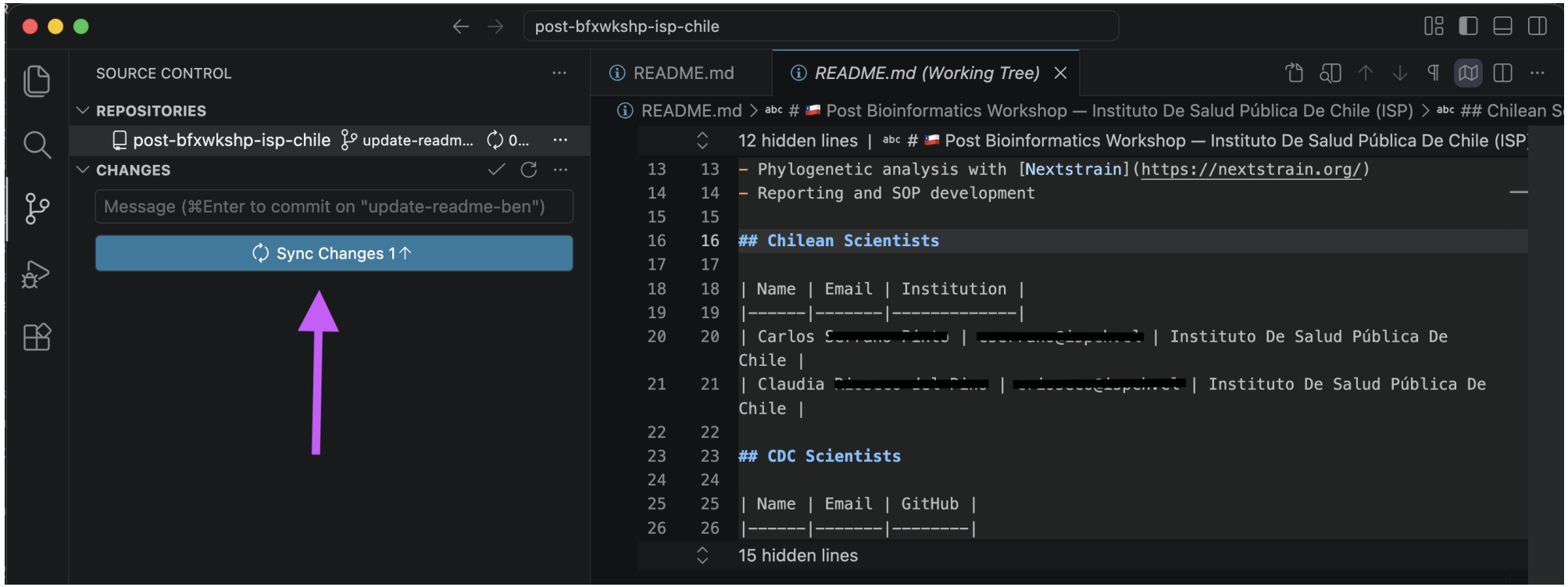
1. Click the **Source Control** tab in the left sidebar.
2. You should see README.md listed under the **Changes** section. Clicking on this file will display a “diff” view showing your changes. Red highlights indicate removed text, and green highlights indicate added text.
3. Hover over README.md and click the **+** (Stage Changes) icon to move it to the **Staged Changes** section.
4. Type a commit message in the **Message** text box at the top of the Source Control panel, e.g., updated heading of local and cdc contributors.
5. • 5. Click the **Commit** button (checkmark icon) to commit your staged changes.



Git in VSCode

After editing and saving a file in the repo via VSCode, Stage Commit, and Push

The blue button will now say **Sync Changes**. Click it to push your commit to GitHub. You may be prompted to sign in to GitHub if you haven't already

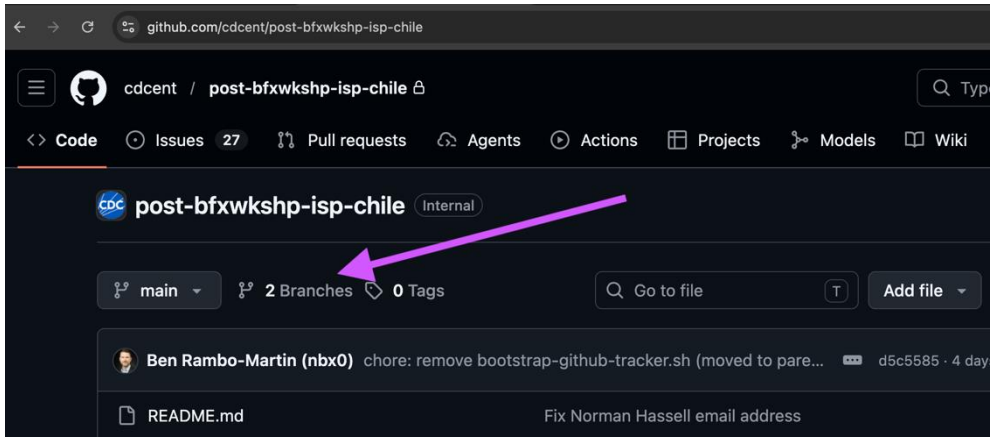


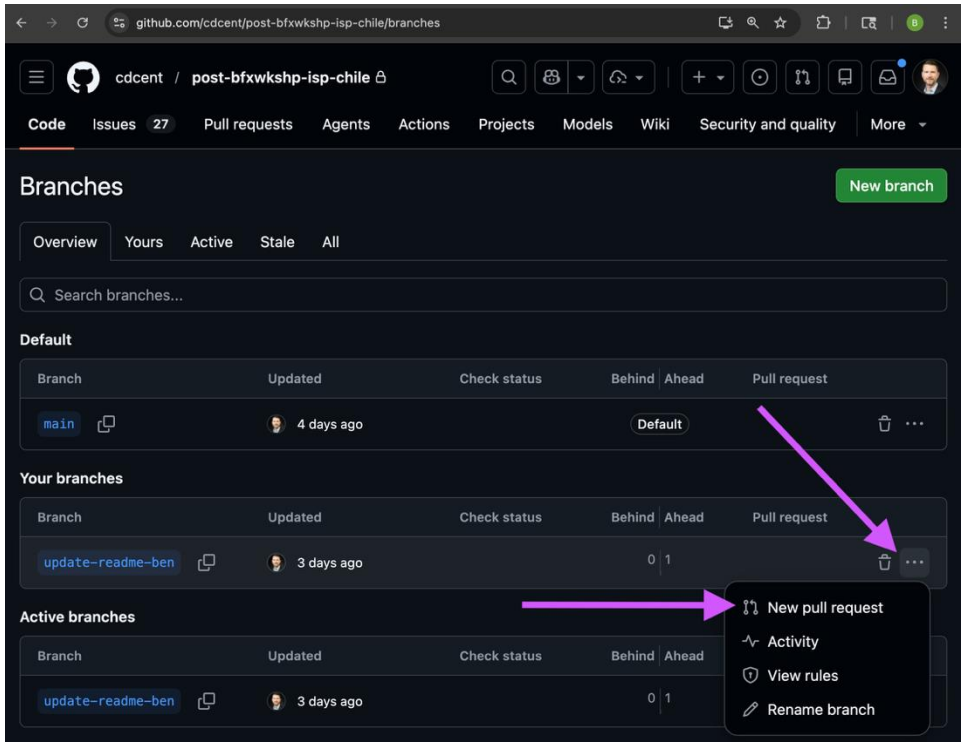
Git in VSCode

Create a Pull Request and Merge (GitHub Website, not VS Code)

1. Open your web browser and navigate to your repository on github.com.
2. Click on the **Branches** tab, then find your branch in the list and click the three dots next to it and click **New pull request**.
3. Review the changes, add a title and description if needed, then click **Create pull request**

4. For these workshop repos, CDC administrators will complete the merge

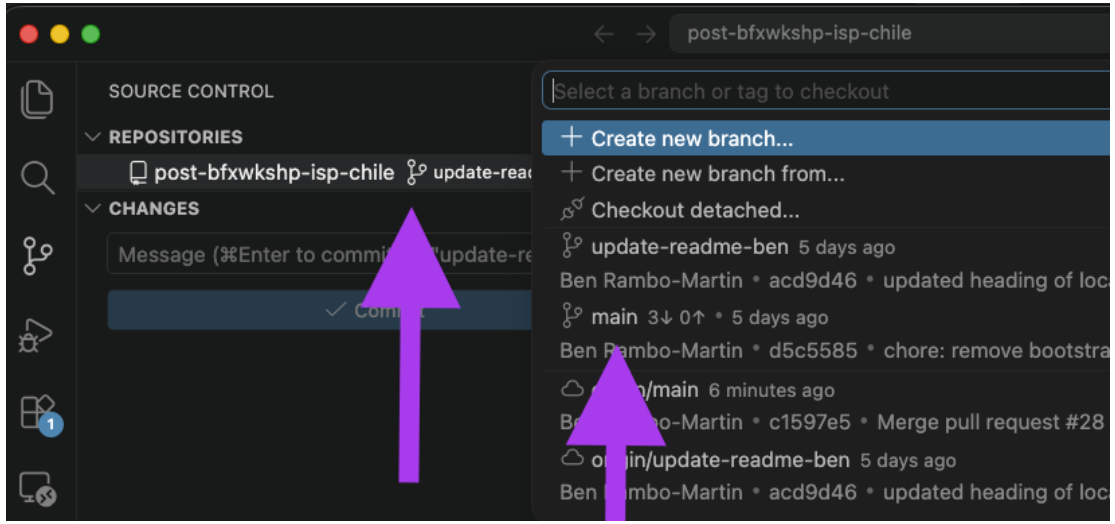


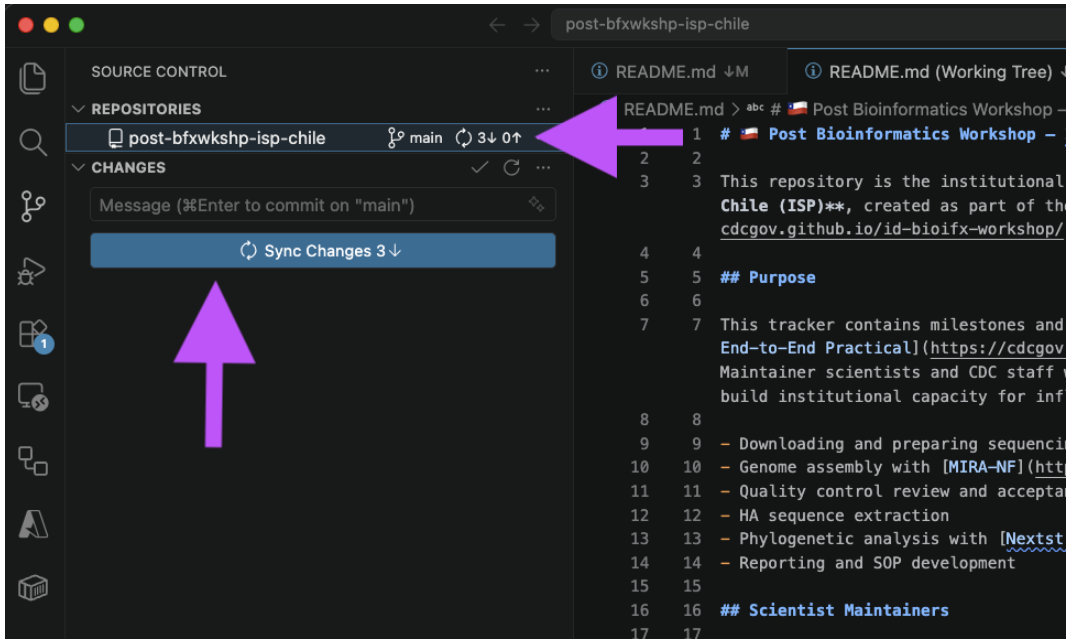


Git in VSCode

Finally: pull changes locally

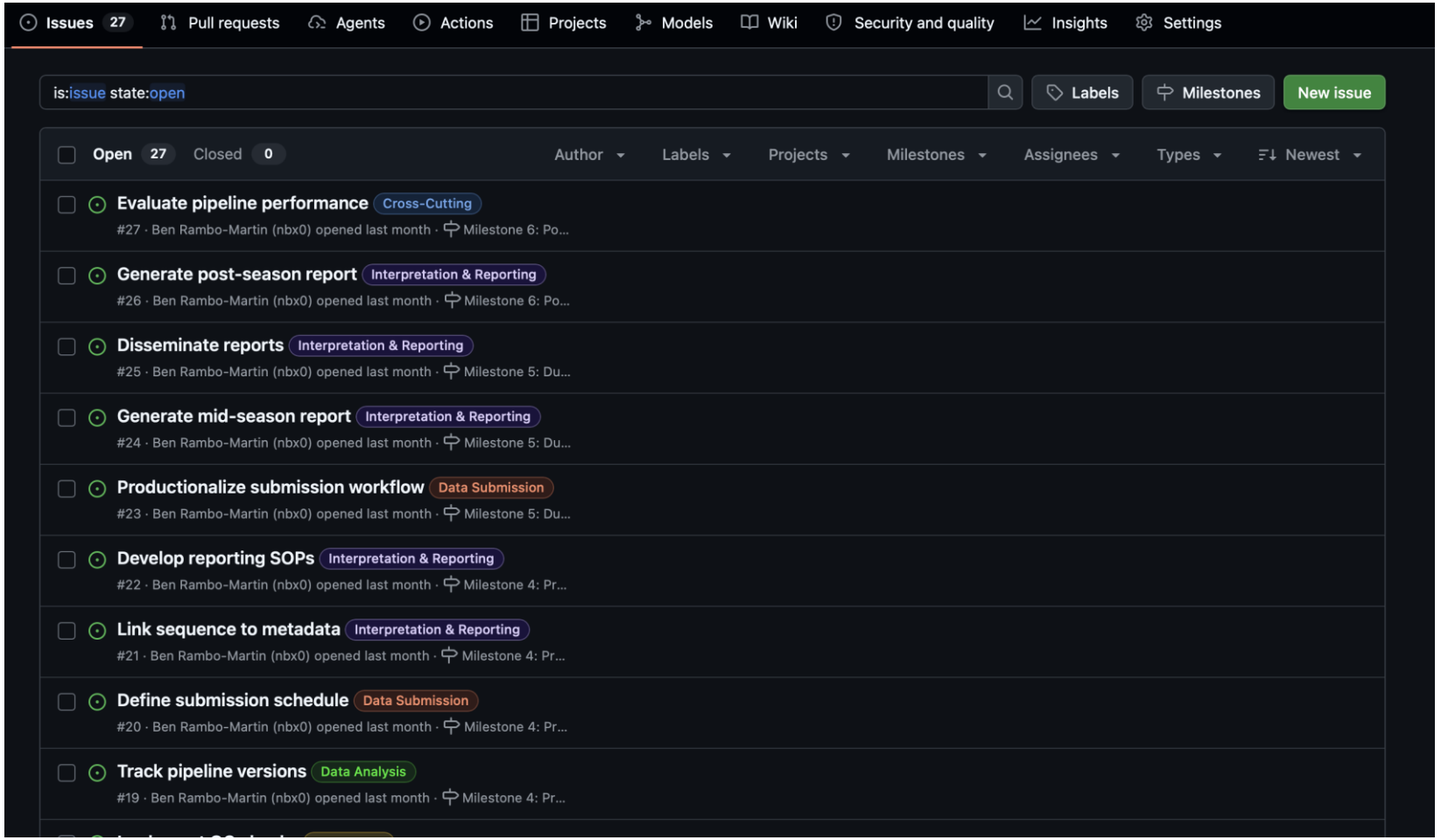
1. Back in VS Code, click the **Source Control** tab. Click the “branch” icon at the top of the **REPOSITORIES** panel to switch branches. Select main to switch back to the main branch
2. Click **Sync Changes** to pull the latest changes from GitHub, which will include the merged changes from your branch.





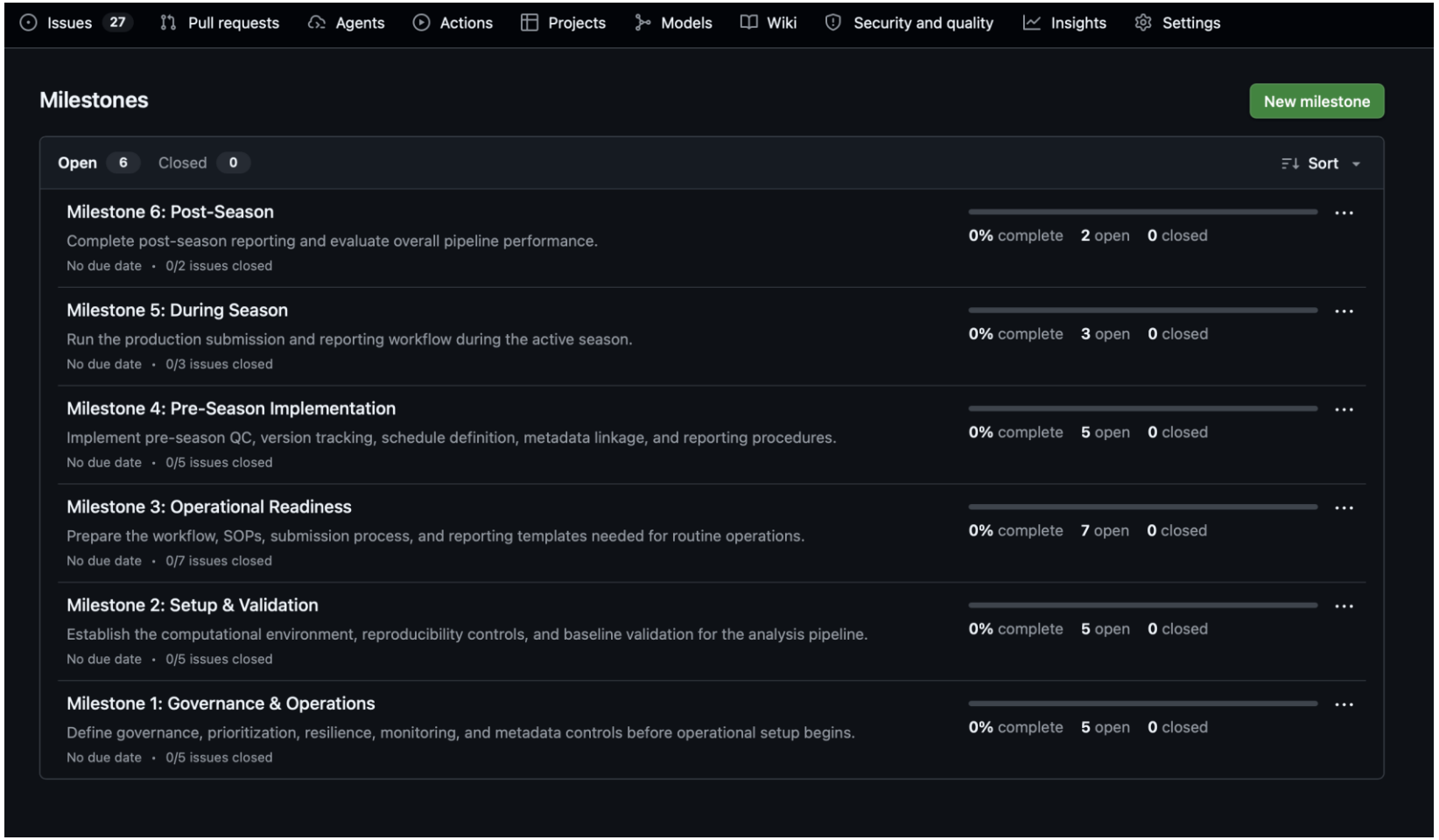
Git issues and milestones

- Git issues (on web) are a to-do list
  - Search
  - Assign
  - Label
  - Leave comments, conversation
  - @ other users
  - “Close” when complete



Git issues and milestones

- Milestones are a collection of issues
  - Bin related issues
  - Track overall progress



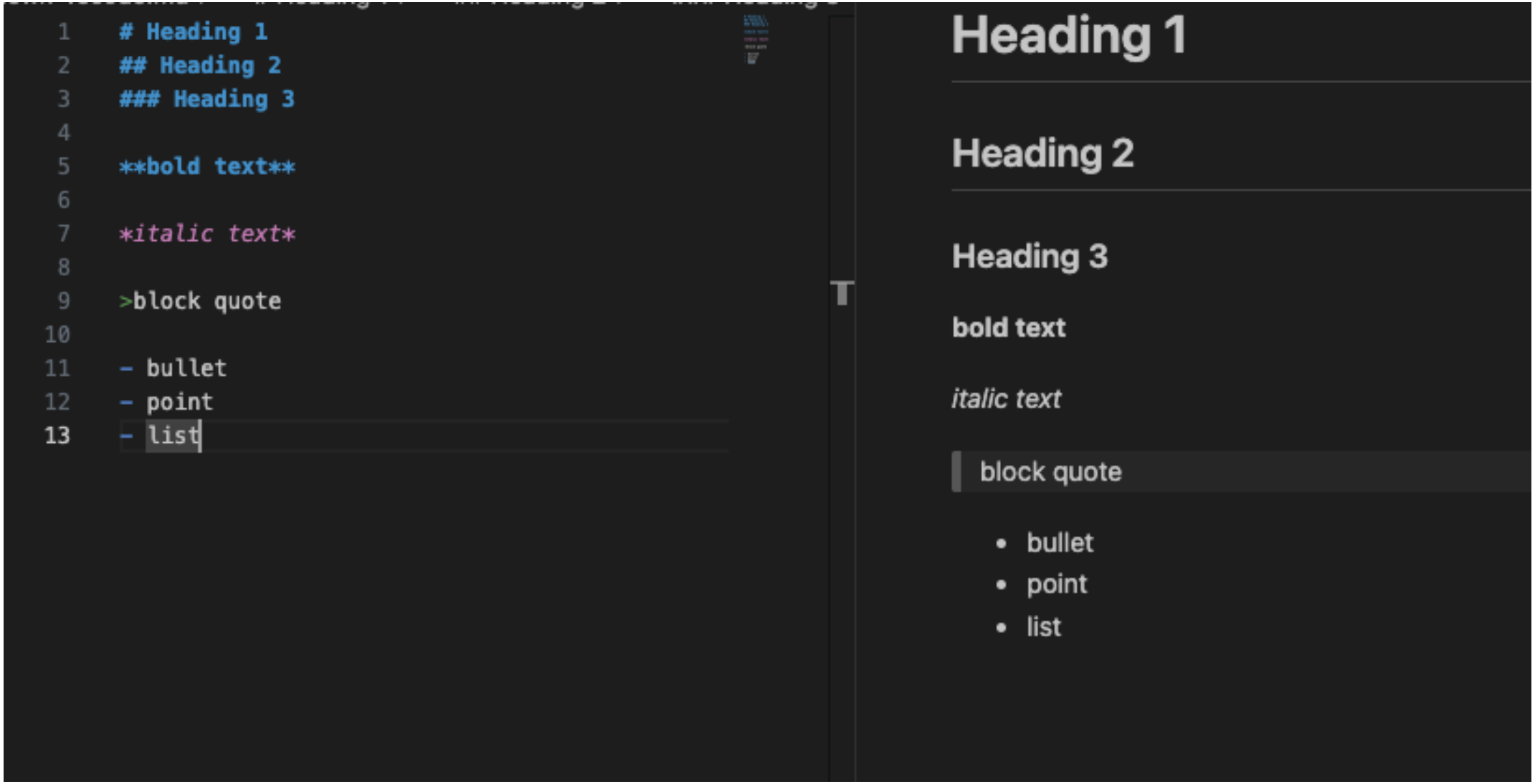
Markdown

- A simple way to format text using plain, readable symbols like asterisks and hashtags
- No special software needed — it can be written in any basic text editor
- Very easy to learn, with a minimal set of rules
- Used widely across the web, including GitHub, blogs, and note-taking apps
- Automatically converts into clean, formatted web pages or documents when **rendered**
- Markdown files end with the extension .md

Markdown

- `# Heading 1, ## Heading 2, ### Heading 3`
  - create headings with pound signs
  - 1 is biggest
- `**bold text**`
  - wrap in double asterisks for **bold**
- `*italicized text*`
  - wrap in single asterisks for *italic*
- `> blockquote`
  - add a > to create an indented quote block
- `- bullet point list`

Markdown


1 # Heading 1
2 ## Heading 2
3 ### Heading 3
4
5 **bold text**
6
7 *italic text*
8
9 >block quote
10
11 - bullet
12 - point
13 - list
Heading 1
Heading 2
Heading 3
bold text
italic text
block quote
bullet
point
list

Markdown

- `1. 2. 3.` : start lines with numbers for an ordered list
- `` `inline code` `` : wrap in backticks for inline code
- ```` ``` ```` on its own line : opens and closes a fenced code block
- `[link text](url)` : Create a clickable link
- `![alt text](image.jpg)` : embed an image (same as a link, but with !)
- `---` : three dashed lines make a horizontal dividing line

Markdown

markdown-vscode.md

```
1. numeric
2. list
3. now

`one line of code`

```
multiple
lines
of
code
```

[Go to google](www.google.com)

![CDC](CDClogo.png)

```

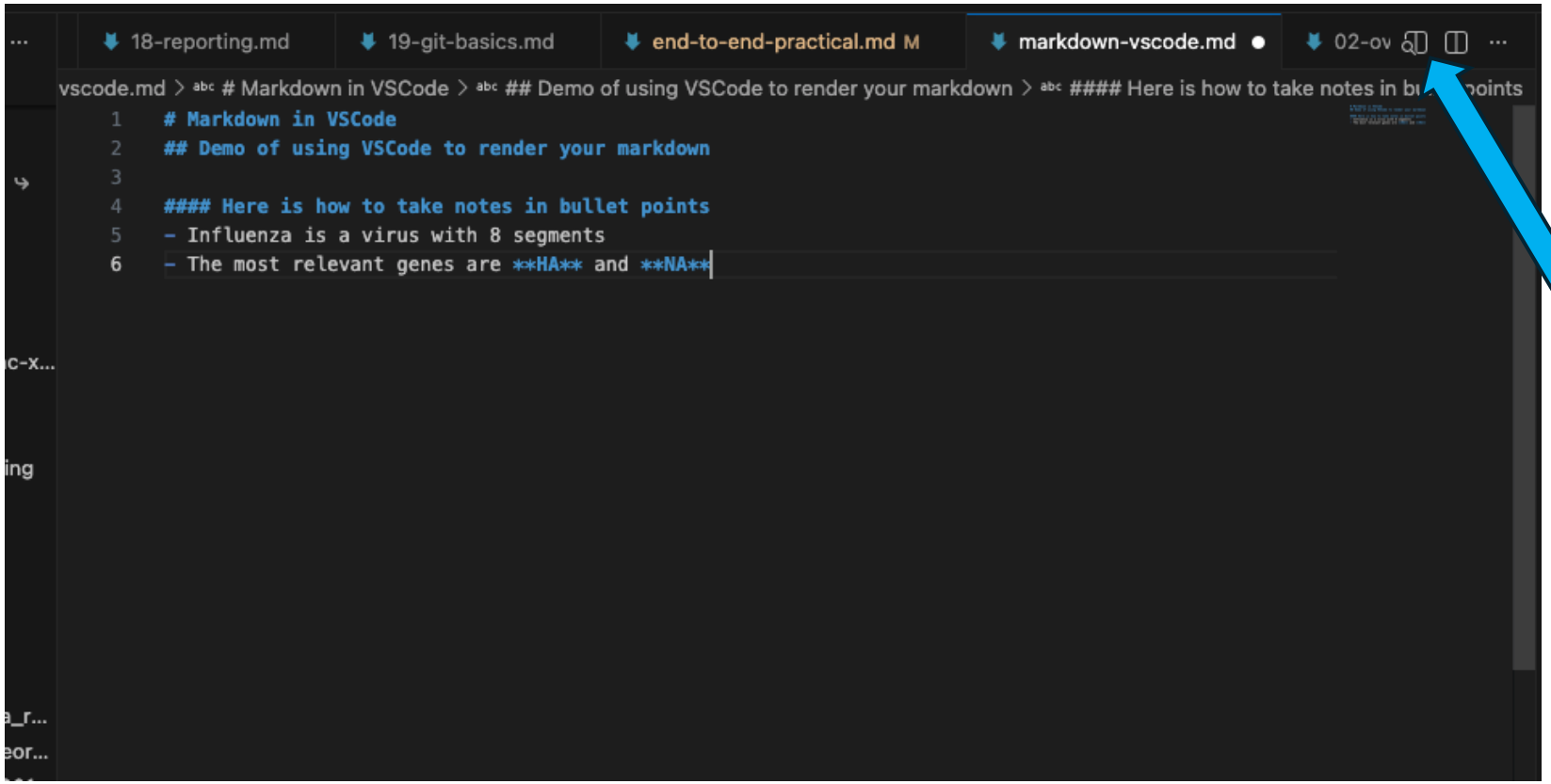
1. numeric
2. list
3. now

`one line of code`

```
multiple
lines
of
code
```

Go to google

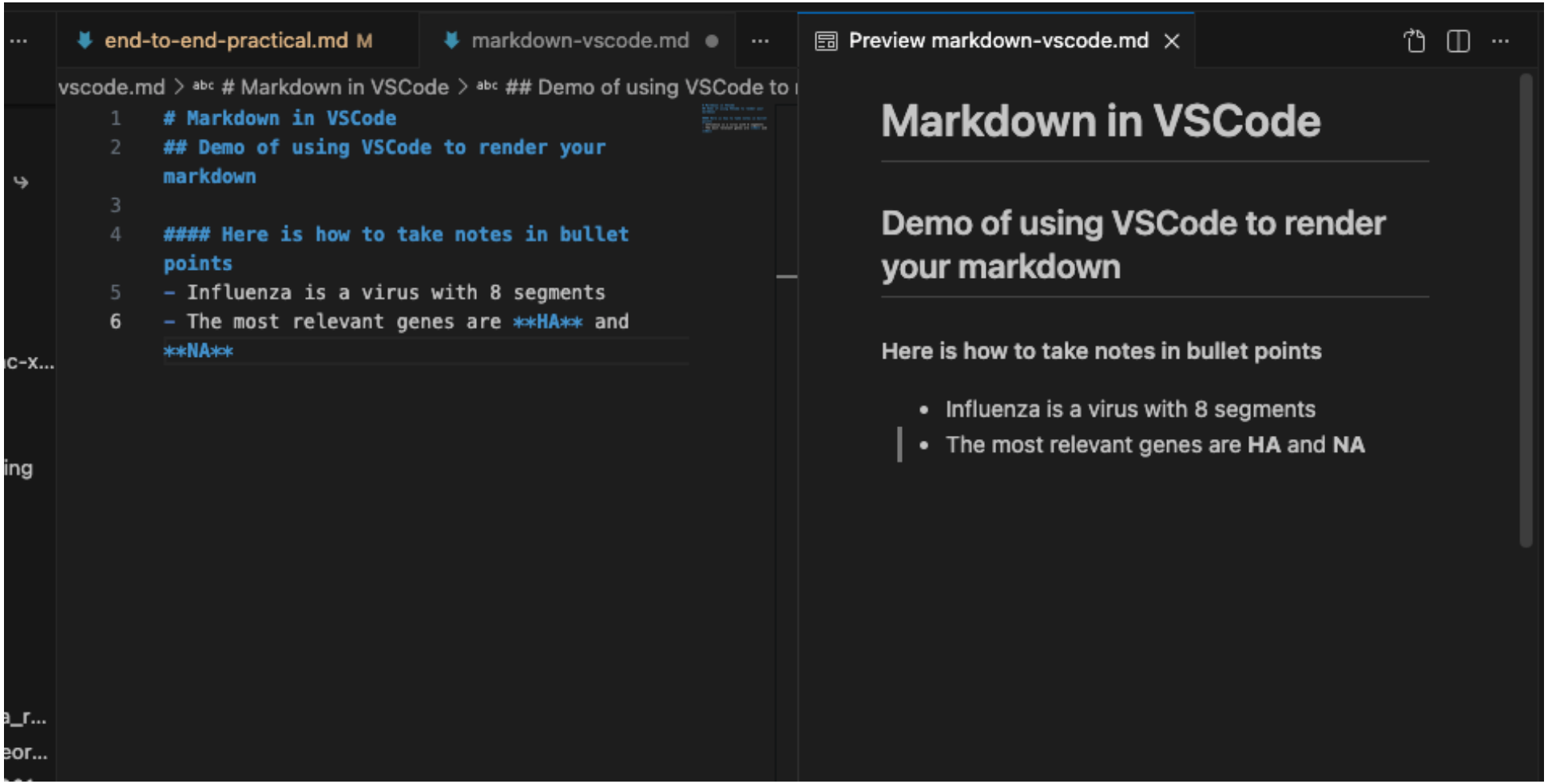
Markdown in VSCode



Open a .md file

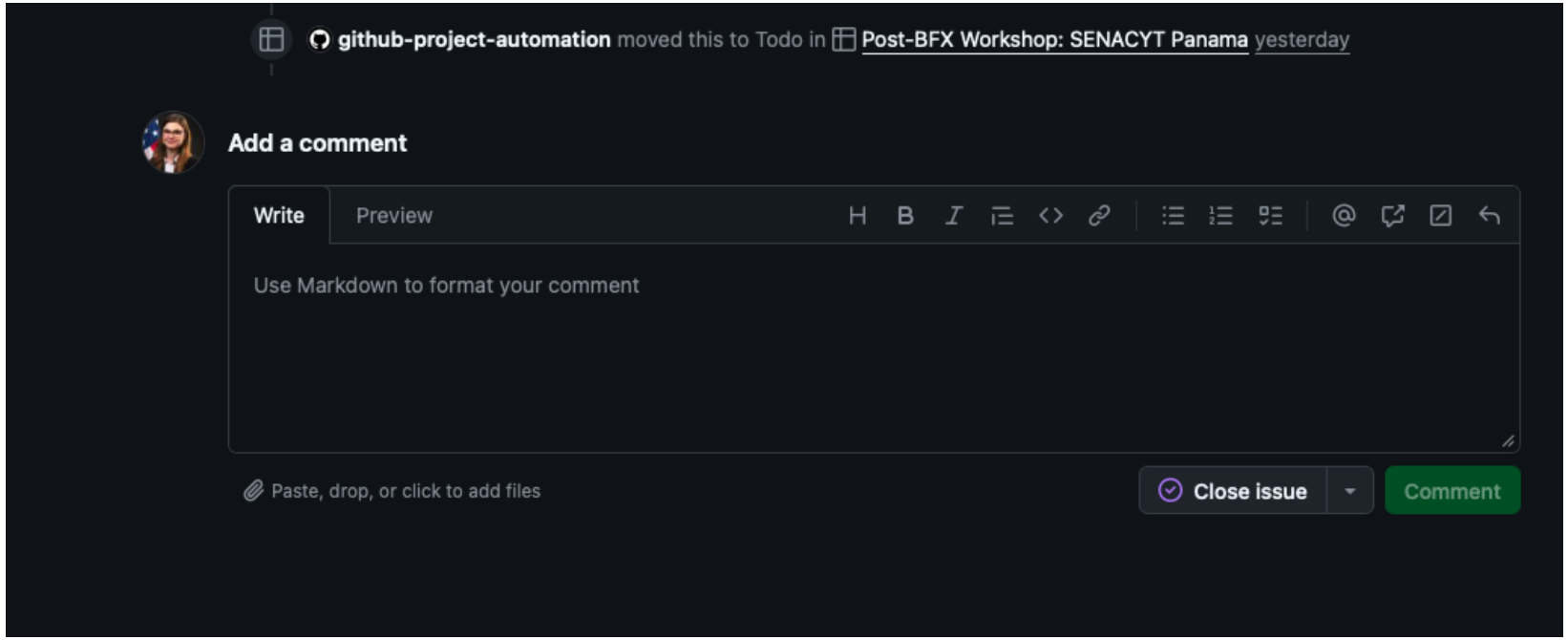
Click the magnifying glass to render a preview

Markdown in VSCode



Git renders Markdown

- Readme files
- Git issues
- Comments on gitlab
- Documentation
- Even our website for this course was written in Markdown!



Markdown in VSCode: note-taking

- This week, take notes in VSCode with Markdown
- It is worthwhile for future reporting
- Cleaner way to write notes about code and commands
- You can commit notes to git where they can be backed up daily
- Collaborate/combine notes with others from your lab

- In the practical, open a new file called Day1_notes.md

Practical: install VSCode

1. If you are on a Windows machine without WSL (Ubuntu), have a trainer help you through the install
2. Go to code.visualstudio.com and install VSCode
3. Clone your workshop repo
4. Follow instructions on course website to branch, start a note-taking markdown file
5. Merge !