

Web Programming

Git collaboration

Leander Jehl | University of Stavanger

Git model

changes

```
..<title>Document</title>
..<title>Example</title>
```



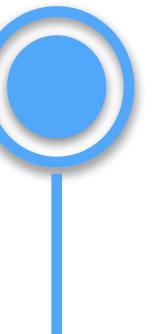
staged



committed



pushed

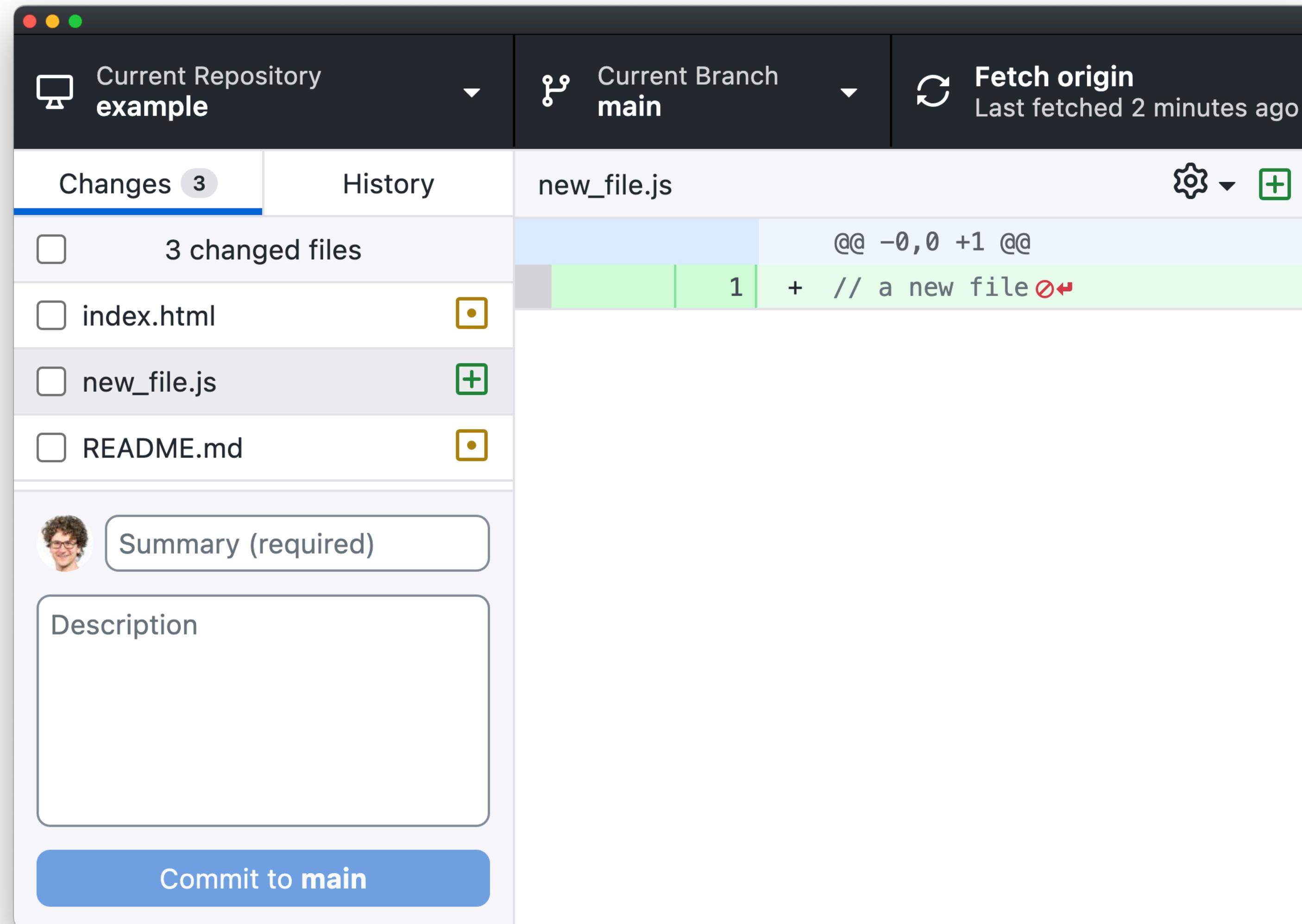


Git detects changes

changes

```
.. <title>Document</title>
.. <title>Example</title>
```

new files

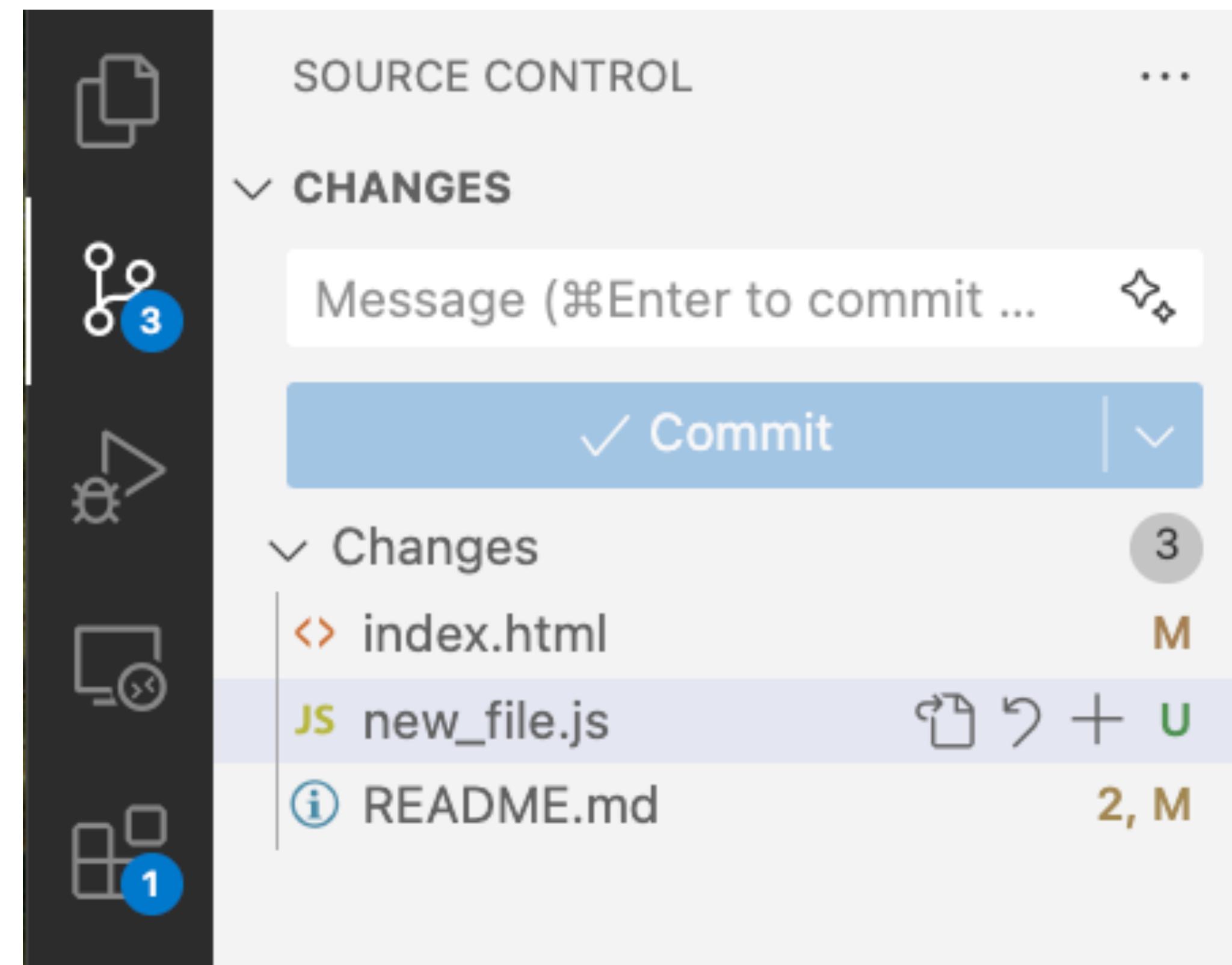


Git detects changes

changes

```
..<title>Document</title>
..<title>Example</title>
```

new files



Git detects changes

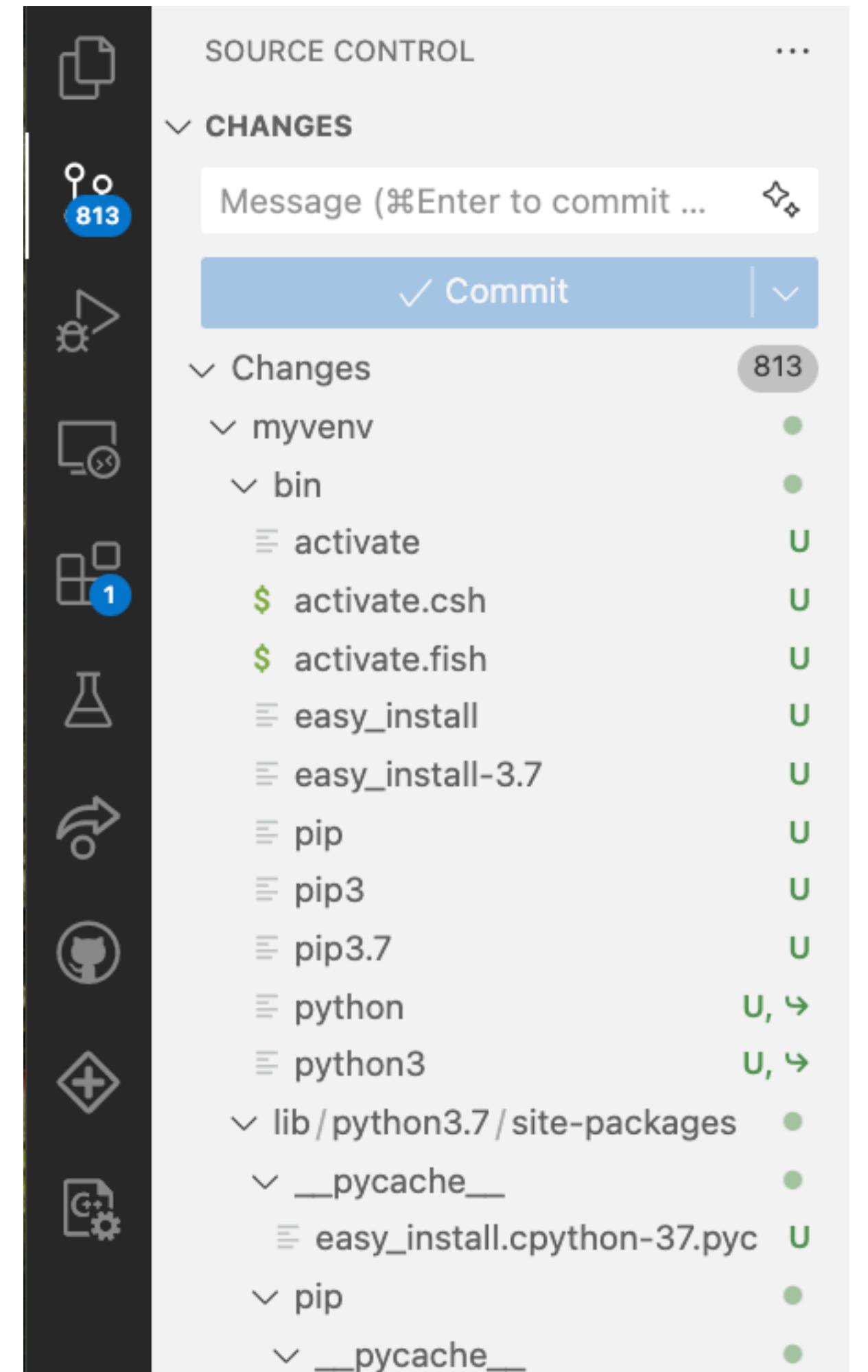
changes

```
...<title>Document</title>
...<title>Example</title>
```

new files



- Some changes should not be committed, e.g. installations in a virtual environment or python cached files.



Git detects changes

changes

```
..<title>Document</title>
..<title>Example</title>
```

new files



- Create file with name `.gitignore`
- Add one line `foldertoignor/`

```
◆ .gitignore U ×
◆ .gitignore
1 myvenv/
2
```

Staged changes should go into one commit

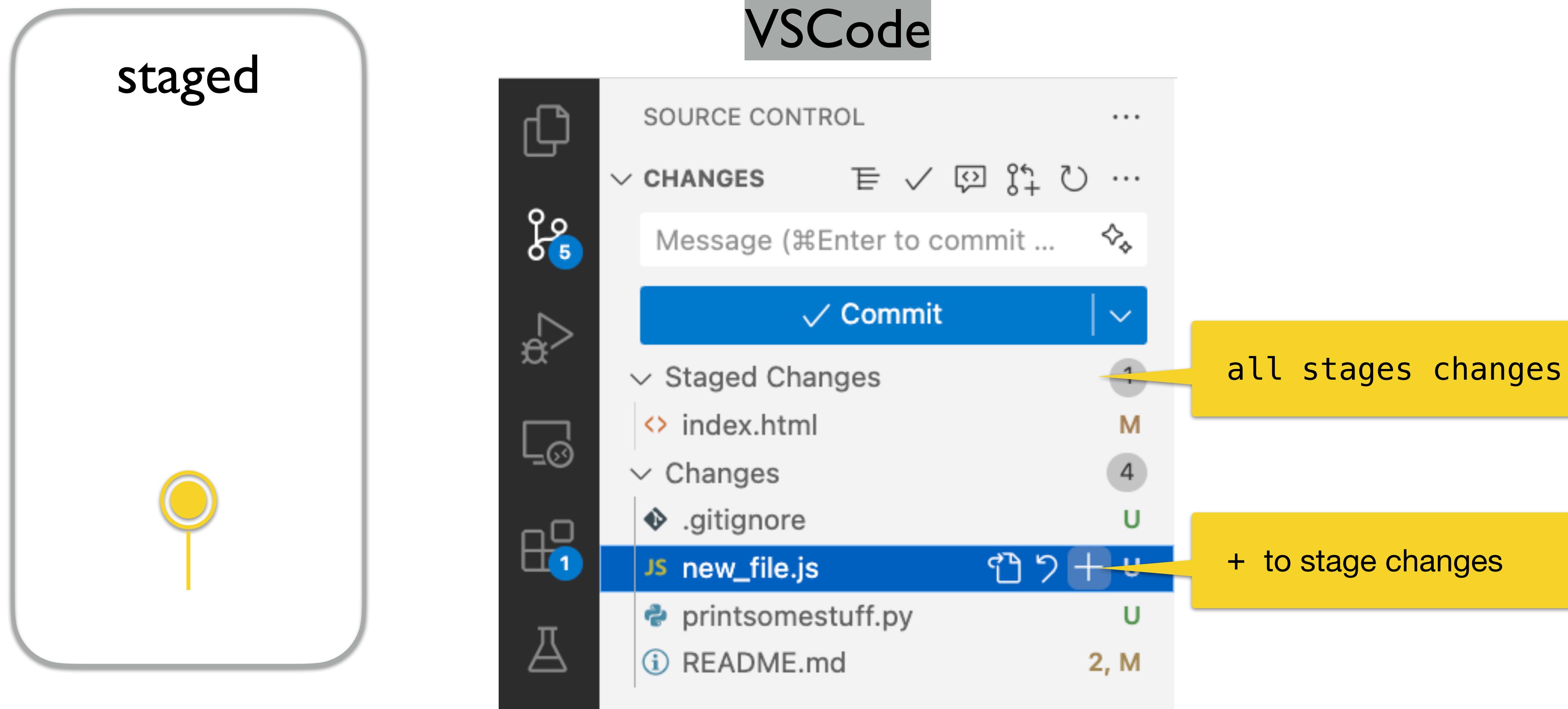
staged



- Select some files from the detected changes.
- Think, which files/changes are part of this feature

```
git add filename
```

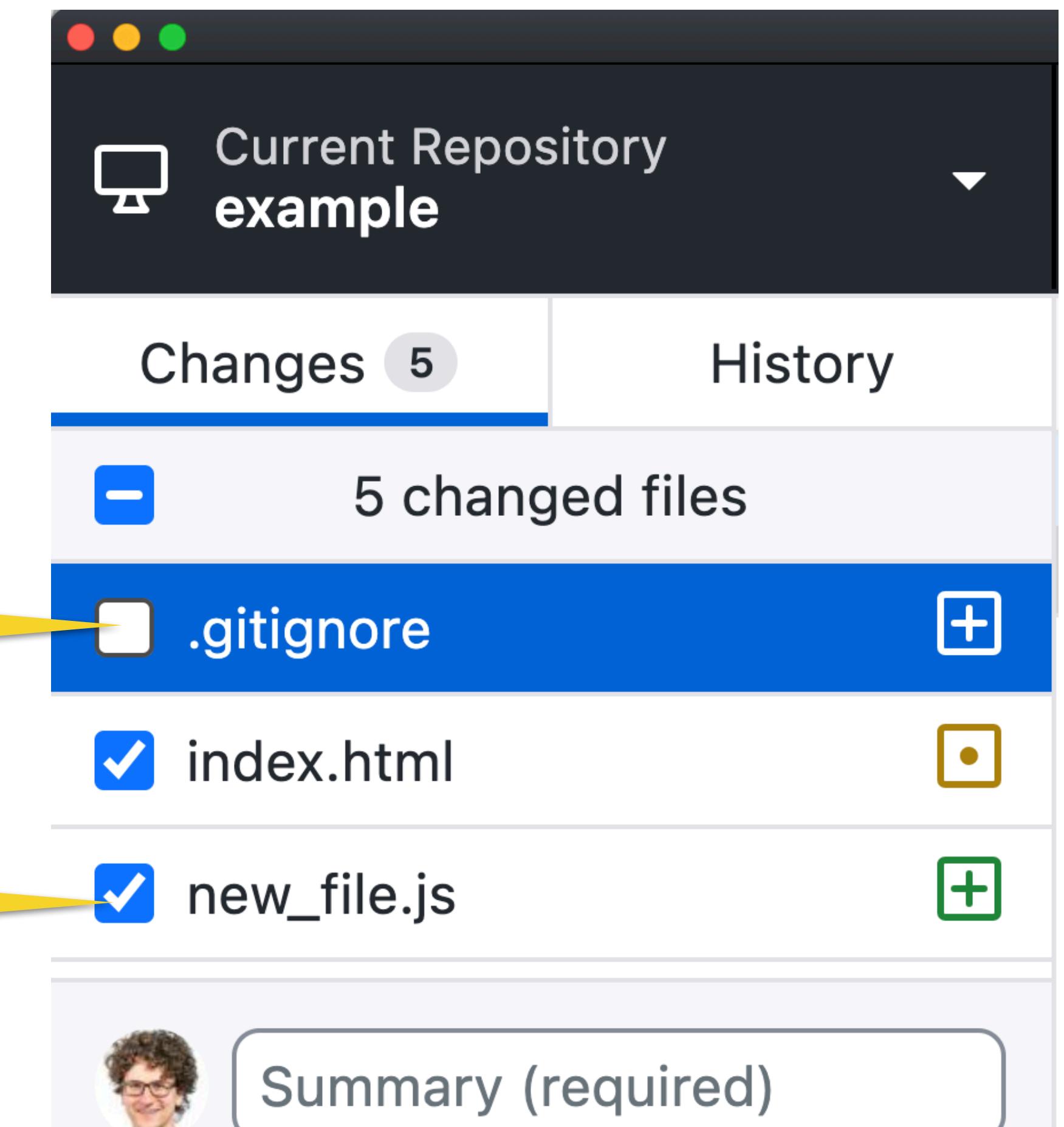
Staged changes should go into one commit



Staged changes should go into one commit



Github Desctop



Commit staged changes

committed



- Create one new version including the changes
- Includes only staged changes
- Add a commit message describing the change

```
git commit -m "commit message"
```

Commit staged changes

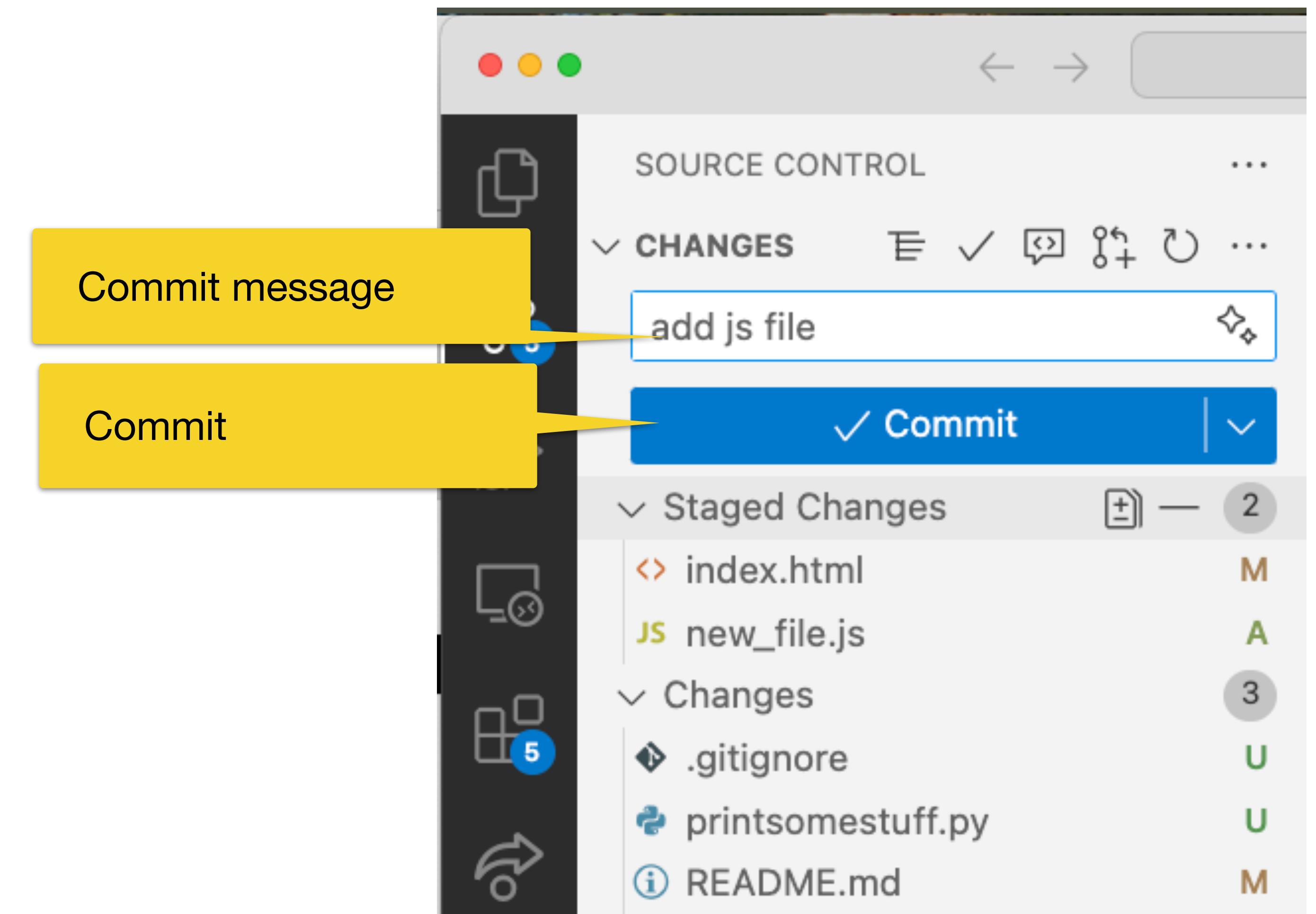
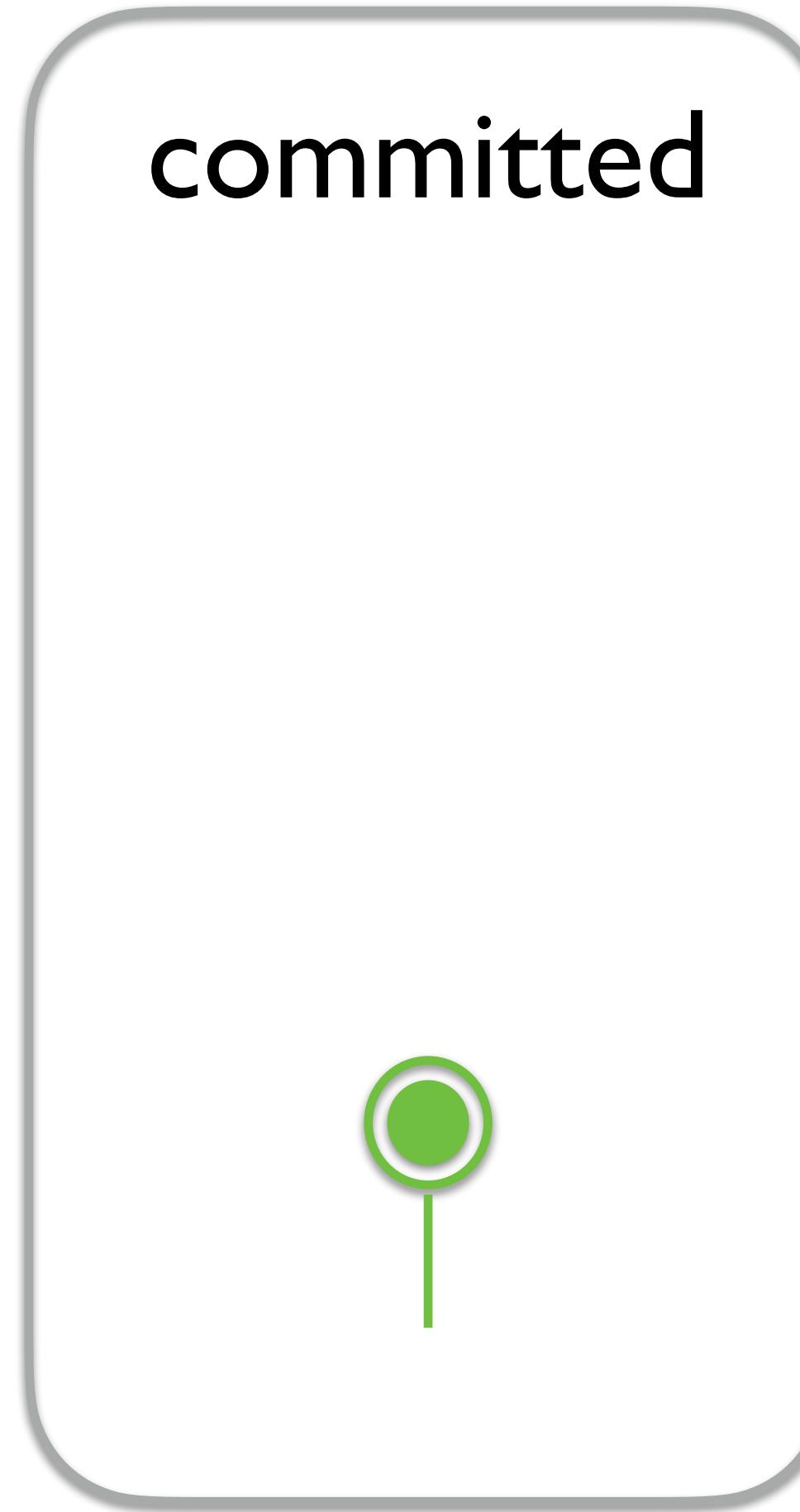


Commit message
must be nonempty

Commit

A screenshot of a Git commit interface. The top bar shows "Current Repository example". Below it, a "Changes" tab is selected, showing "5 changed files": ".gitignore" (unstaged), "index.html" (staged), and "new_file.js" (staged). A yellow callout points to the "index.html" entry. A blue input field contains the text "add js file". A "Description" text area is below. At the bottom, a blue button says "Commit to main".

Commit staged changes



View commits

committed



- Possible to see a list of commits

git log

- First 6 digits are id

```
commit 4f37403ca1c7c06d434f7887564cb4c14af02489 (HEAD -> main)
Author: Leander Jehl <leanderjehl@yahoo.no>
Date: Mon Apr 7 08:54:04 2025 +0200

    add js file

commit 6c2b10fb07af4054347d6e997572dbace0ec34d (origin/main, origin/HEAD)
Author: Leander Jehl <leanderjehl@yahoo.no>
Date: Fri Apr 4 11:50:29 2025 +0200

    change page title

commit 334517edc849a2dfd75965365e4cb0f5afaa80db
Author: Leander Jehl <leanderjehl@yahoo.no>
Date: Fri Apr 4 10:00:13 2025 +0200

    added html example

commit 610285d990d4257761f73e5b3b18fe8efbc8edd3
Author: Leander Jehl <leander.jehl@uis.no>
Date: Fri Apr 4 09:57:43 2025 +0200

    Initial commit
:
```

View commits

committed



- See changes after this commit:
`git diff 6c2b10`
- See changes after commit to index.html file
`git diff 6c2b10 -- index.html`



View commits

committed



See commits under History

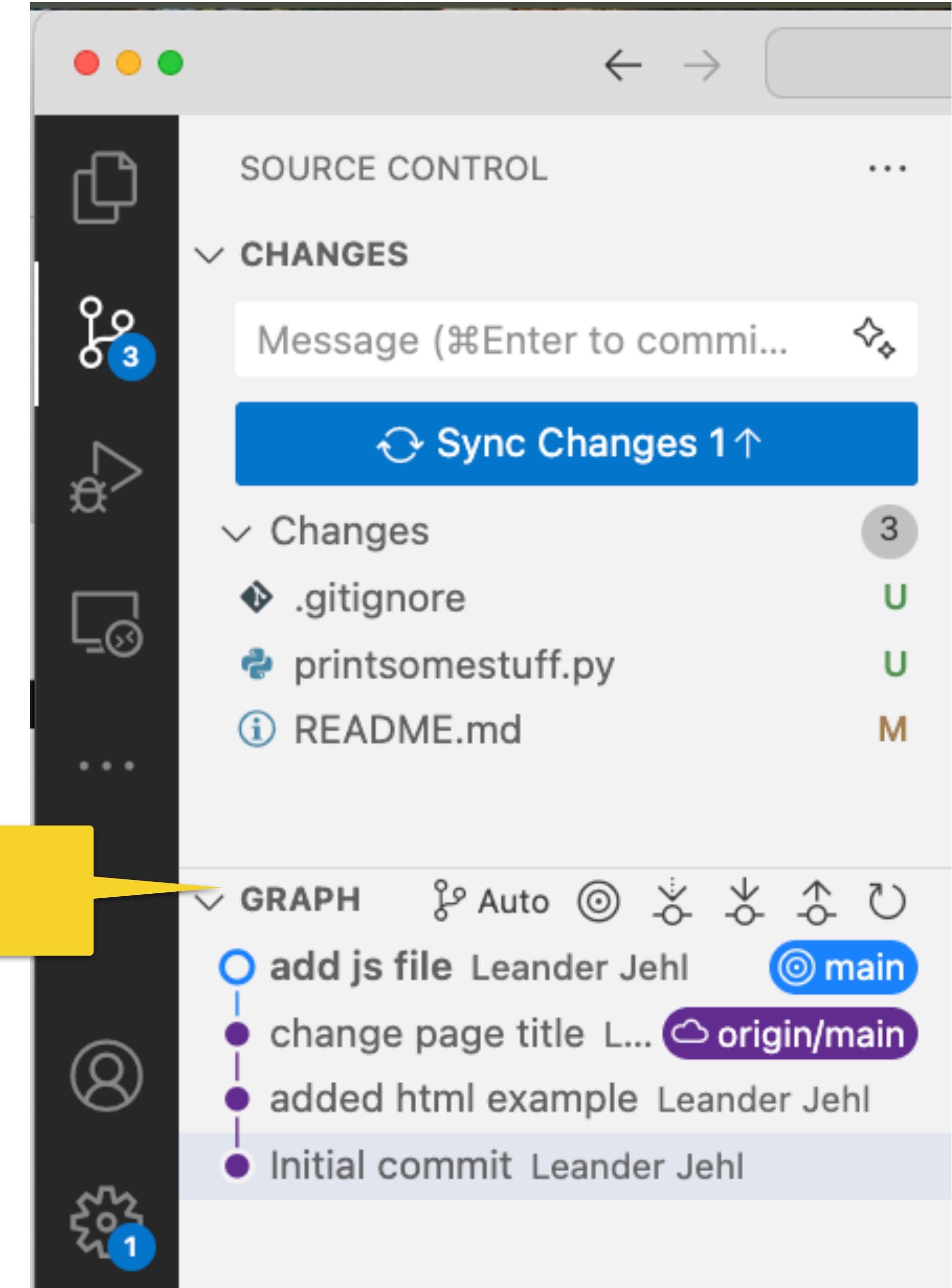
The screenshot shows a GitHub desktop interface with the following details:

- Current Repository:** example
- Current Branch:** main
- Last fetched:** Push origin (Last fetched ...)
- Commits:**
 - add js file** by Leander Jehl (4f37403) - 7 minutes ago (2 changed files: index.html)
 - change page title** by Leander Jehl (3 days ago)
 - added html example** by Leander Jehl (3 days ago)
 - Initial commit** by Leander Jehl, GitHub (3 days ago)
- File Changes:** index.html (diff view: @@ -4, +4, 7 @)
- Branches:** new_file.js (status: +)

View commits

committed

Commits



Push commits to origin

pushed



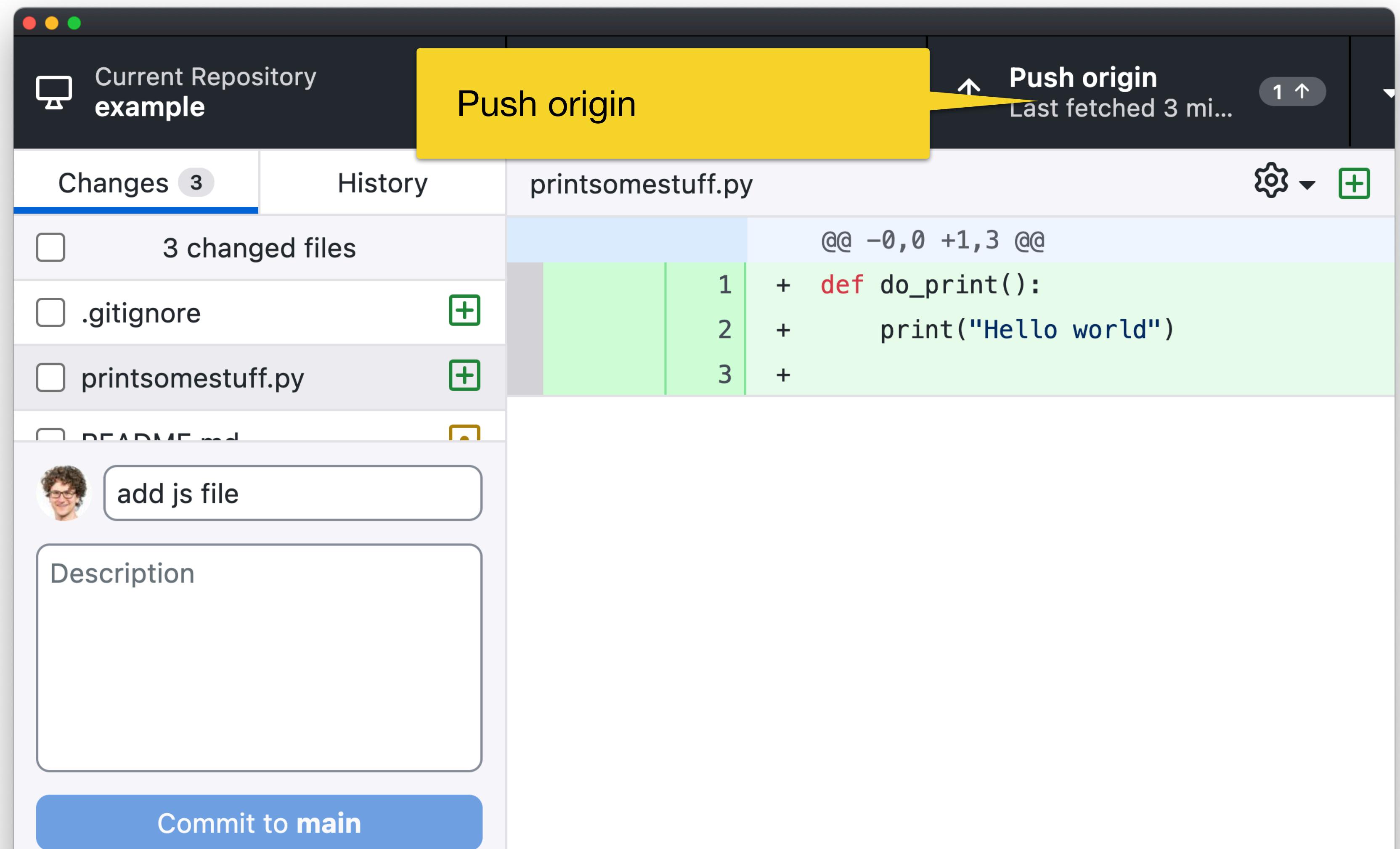
- Store new commit to GitHub

```
git push
```

- Only possible if there are no other commits on GitHub that no local yet.

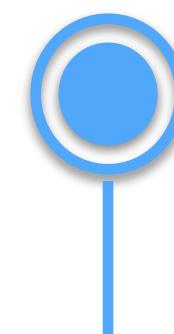
Push commits to origin

pushed



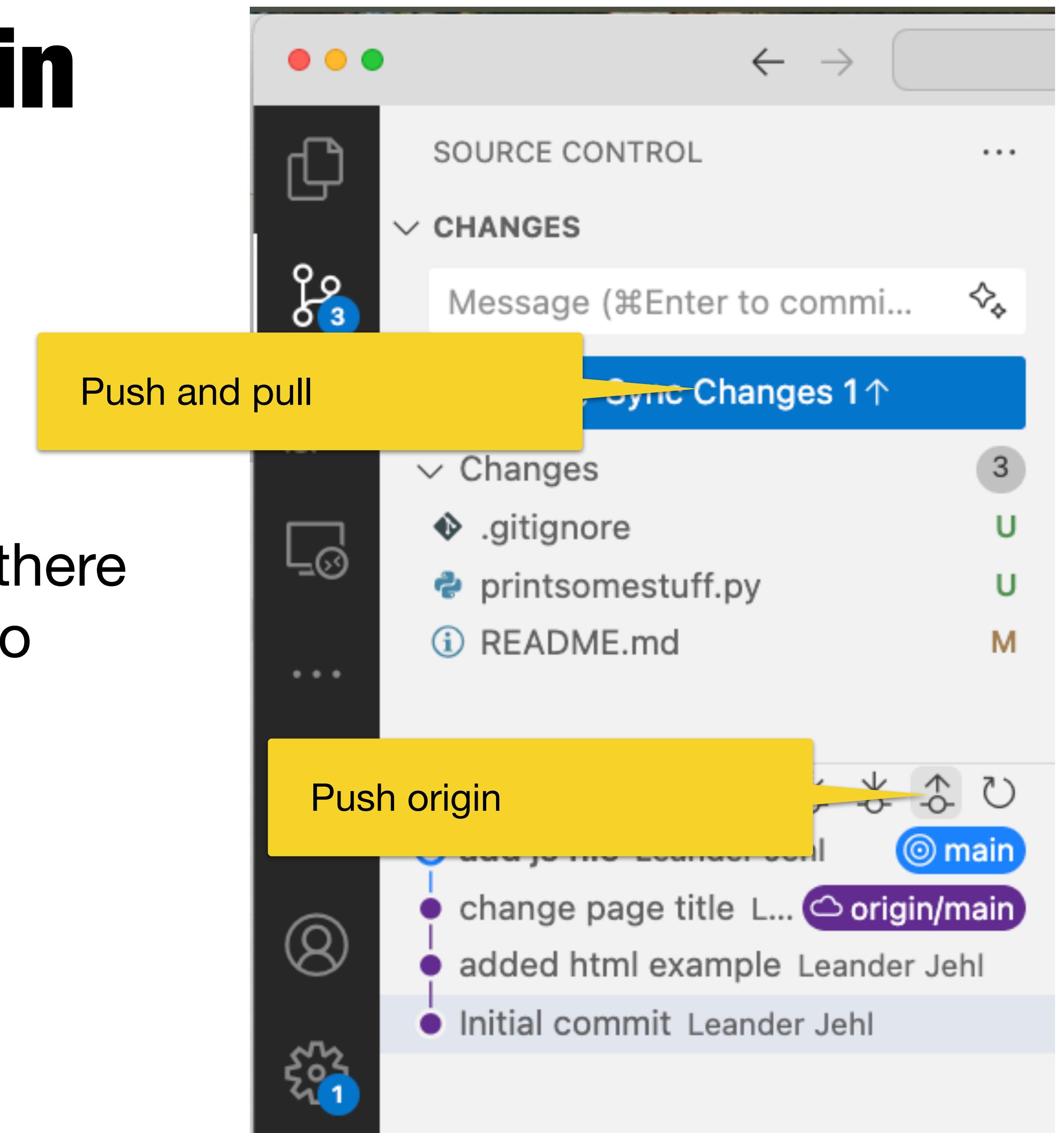
Push commits to origin

pushed



VS Code:

Sync only possible if there
are no local changes to
stages files.



Pull commits from origin

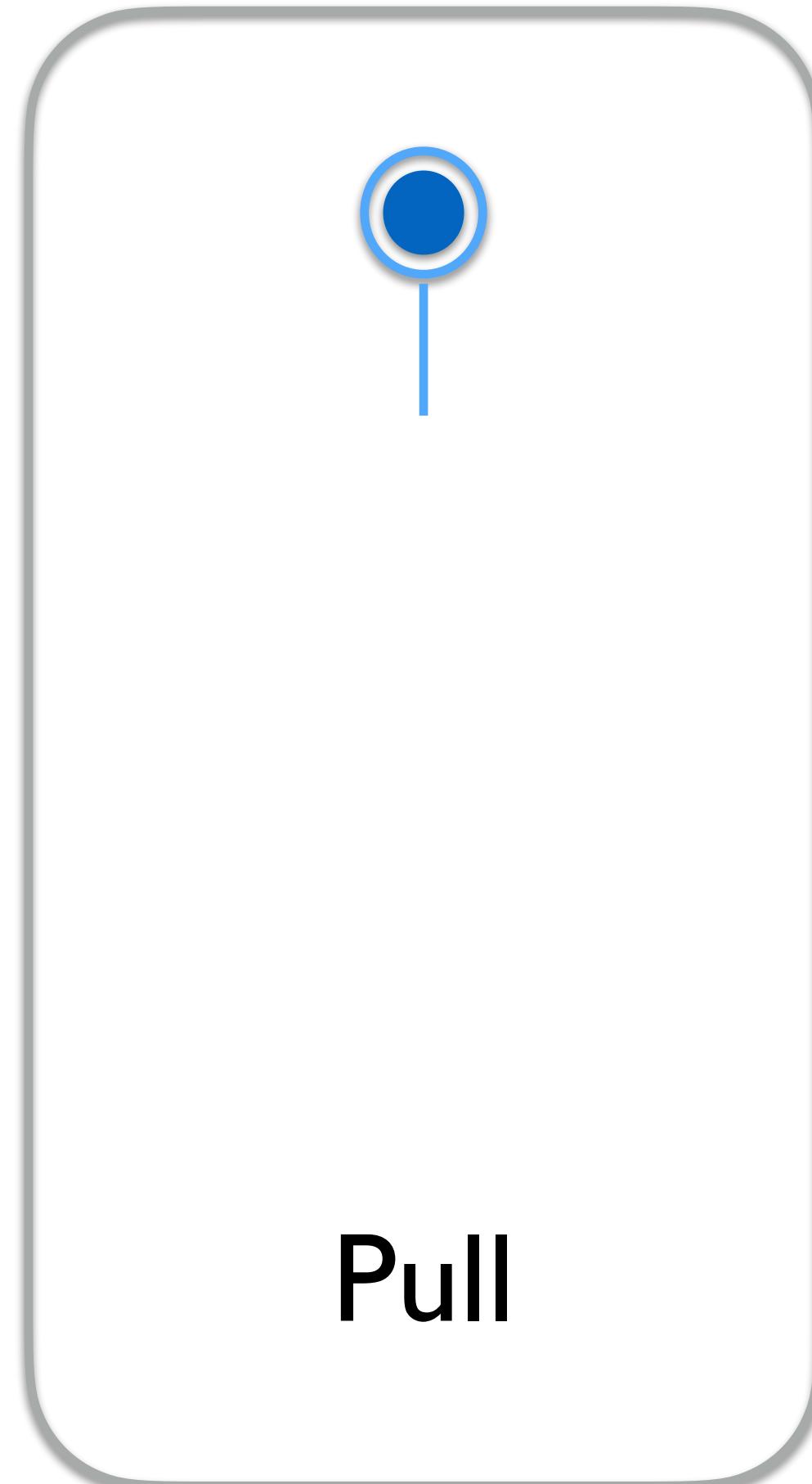


Pull

- Pull any new commits from origin

```
git pull
```

Pull commits from origin



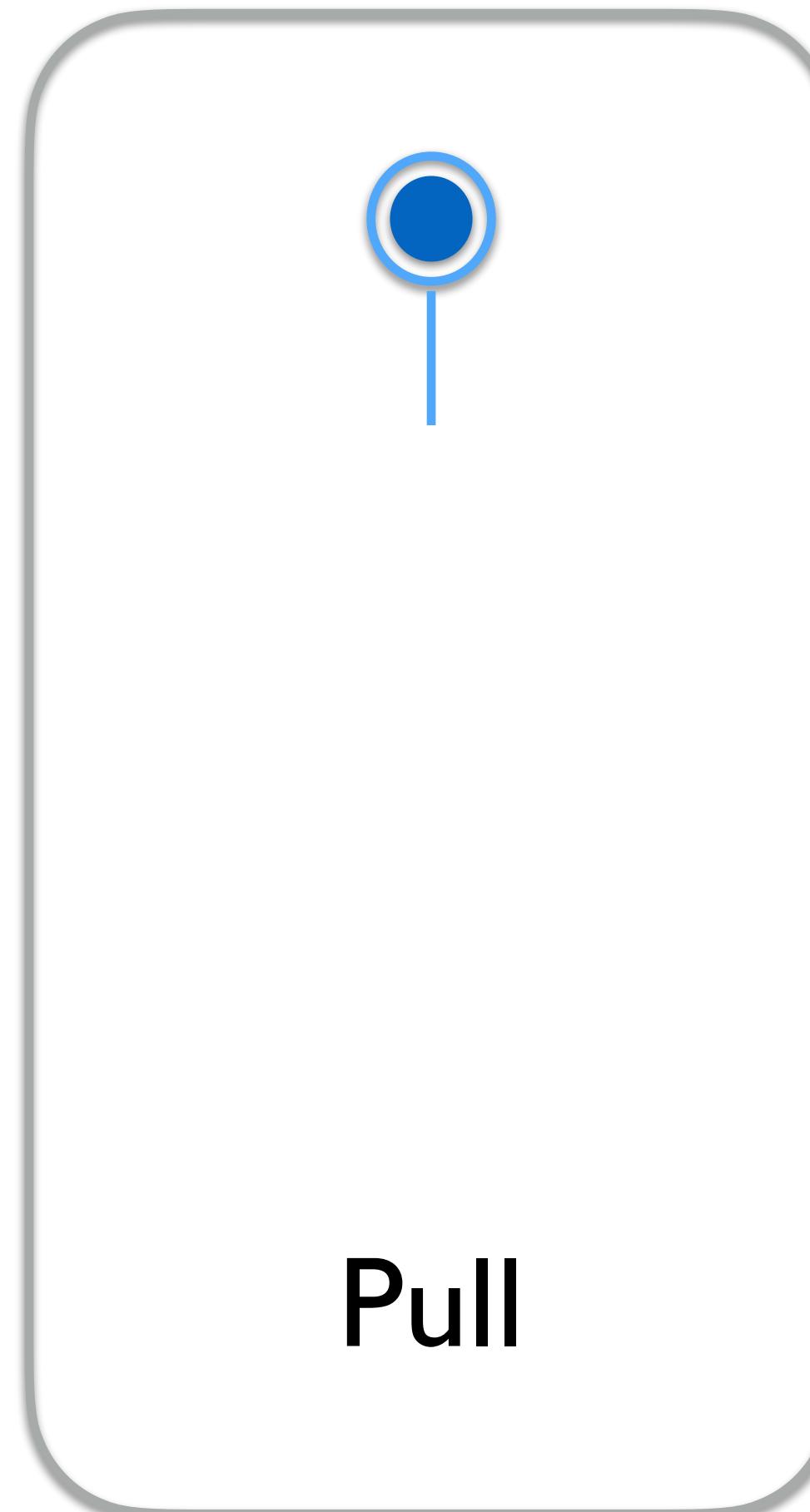
Fetch to check for changes

Fetch origin

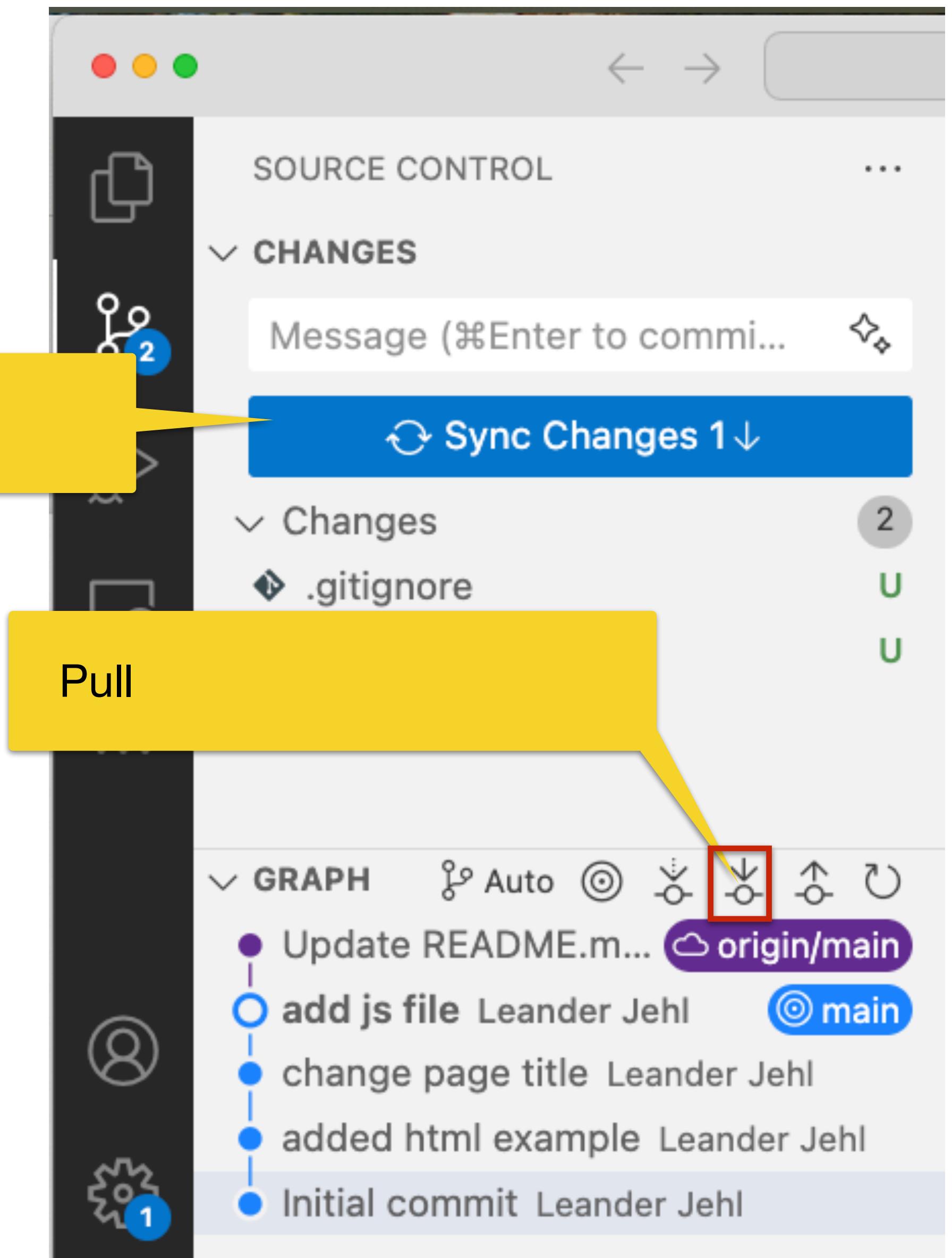
Last fetched 31 minutes ago

The screenshot shows a GitHub pull request interface. At the top, there's a yellow button labeled "Pull". To its left is a "Current Repository" section with a monitor icon and the text "example". To the right is a "Pull origin with..." section with a "Last fetched 2 mi..." message and a "1 ↓" button. Below this, there are two tabs: "Changes" (with 2 items) and "History" (selected). A message "No Branches to Compare" is displayed. The main area shows a commit titled "add js file" by Leander Jehl (commit hash 4f37403). This commit has "+2 -0" changes and involves 2 changed files: "index.html" (marked with a yellow dot icon) and "new_file.js" (marked with a green plus icon). The "index.html" file has a diff view showing additions and deletions. On the far left, a sidebar lists other commits: "add js file" (Leander Jehl, 1 hour ago), "change page title" (Leander Jehl, 3 days ago), "added html example" (Leander Jehl, 3 days ago), and "Initial commit".

Pull commits from origin



Sync fetches and pulls



Local and remote commit

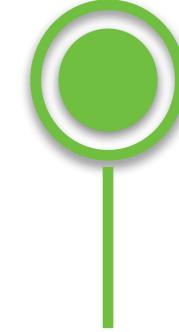
- Both updated, cannot push



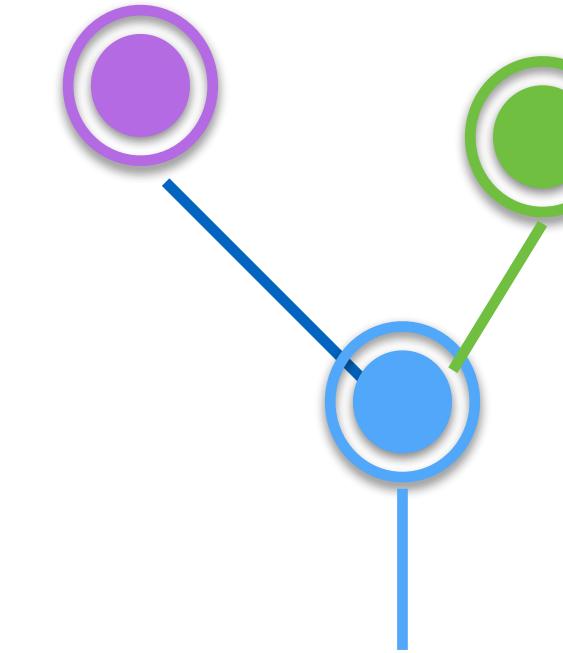
```
● ● ● example — leanderjehl@Leanders-MacBook-Pro-2 — -zsh — 83x25
.. /2025/example

[→ example git:(main) ✘ git push
To github.com:dat310-2025/example.git
 ! [rejected]      main -> main (non-fast-forward)
error: failed to push some refs to 'github.com:dat310-2025/example.git'
hint: Updates were rejected because the tip of your current branch is behind
hint: its remote counterpart. Integrate the remote changes (e.g.
hint: 'git pull ...') before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
→ example git:(main) ✘ ]
```

Local and remote commit



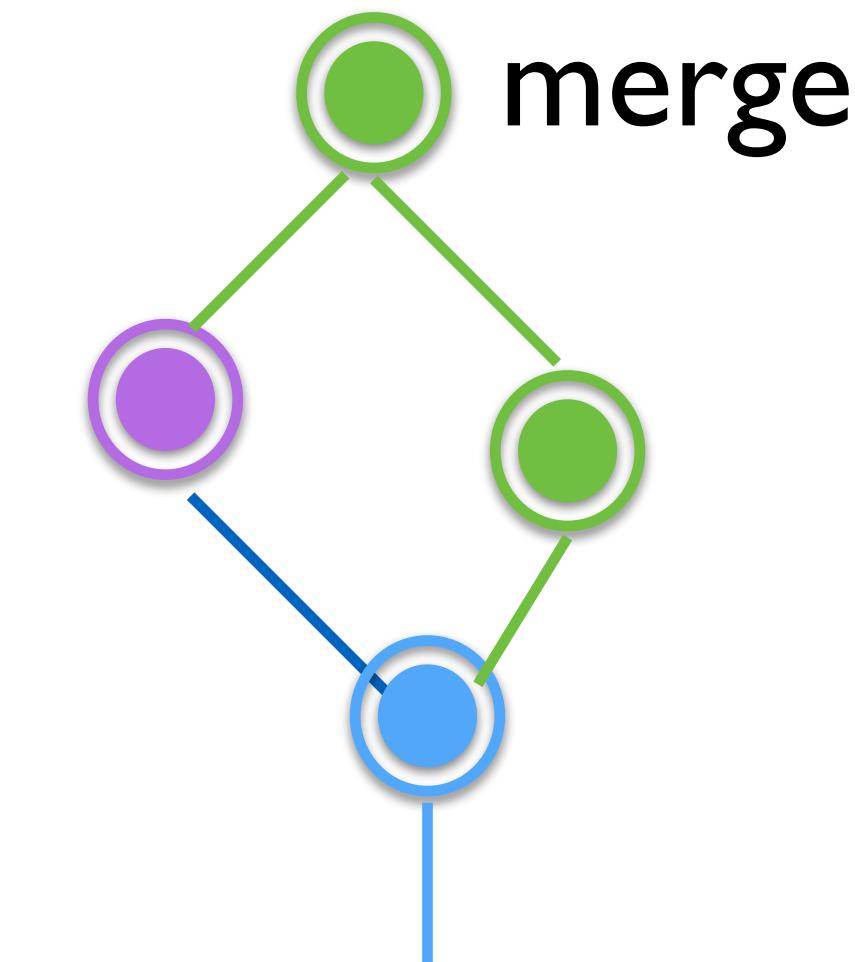
- Both updated, cannot push
- **Need to pull first**
- **Creates merge commit**



Local and remote commit



- Both updated, cannot push
- **Need to pull first**
- **Creates merge commit**
 - if changes are not conflicting



Local and remote commit



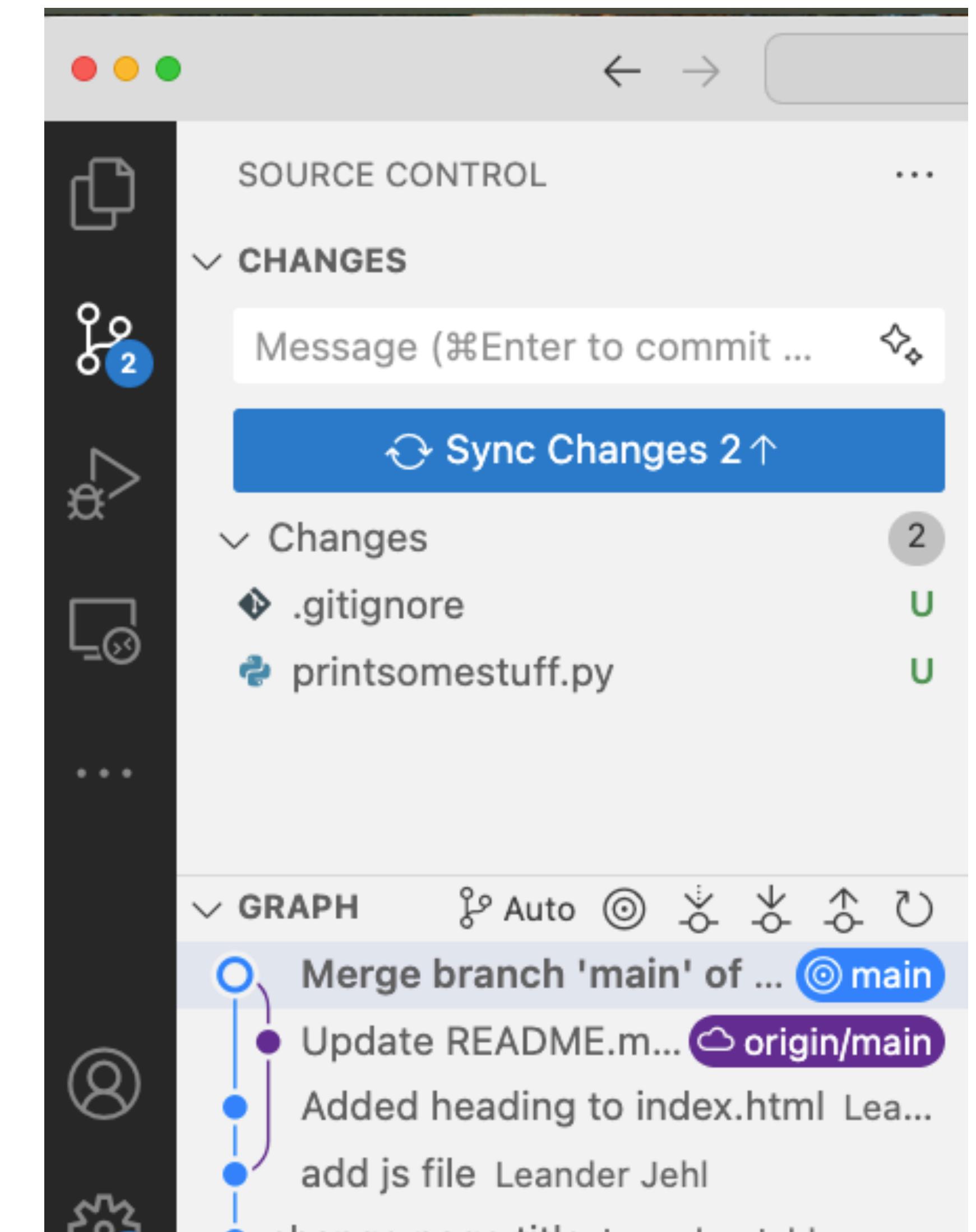
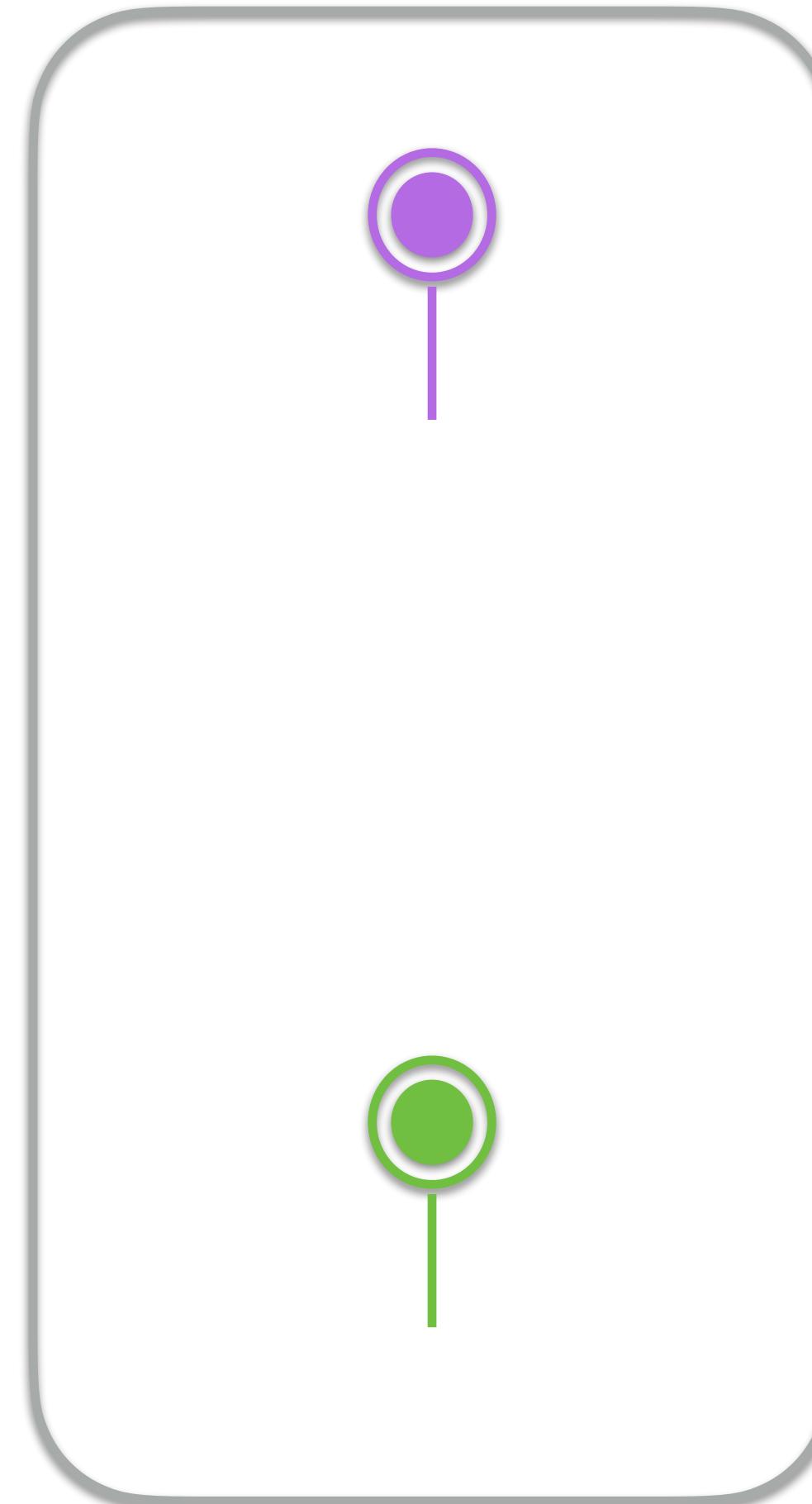
The screenshot shows a GitHub desktop application interface. The top bar displays the current repository as "example", the current branch as "main", and the push origin status. The main area has tabs for "Changes" (2) and "History". The "History" tab is active, showing a list of commits:

- Merge branch 'main' of ... by Leander Jehl • 13 minutes ago
- Added heading to index... by Leander Jehl • 9 hours ago
- Update README.md by Leander Jehl, GitHub • 9 hours ago
- add js file by Leander Jehl • 10 hours ago
- change page title by Leander Jehl • 3 days ago

The right side of the screen shows a detailed view of the most recent commit: "Added heading to index.html" by Leander Jehl (commit 371fb1a). It shows a diff with 1 changed file, "index.html", where line 7 was updated from 7 to 7. The diff highlights changes in the script section of the file.

Line	Old Value	New Value	Type
7	7	7	<script>
8	8	-	</head>
9	9	-	<body>
10	-	10	<h1>
		+	<h1>

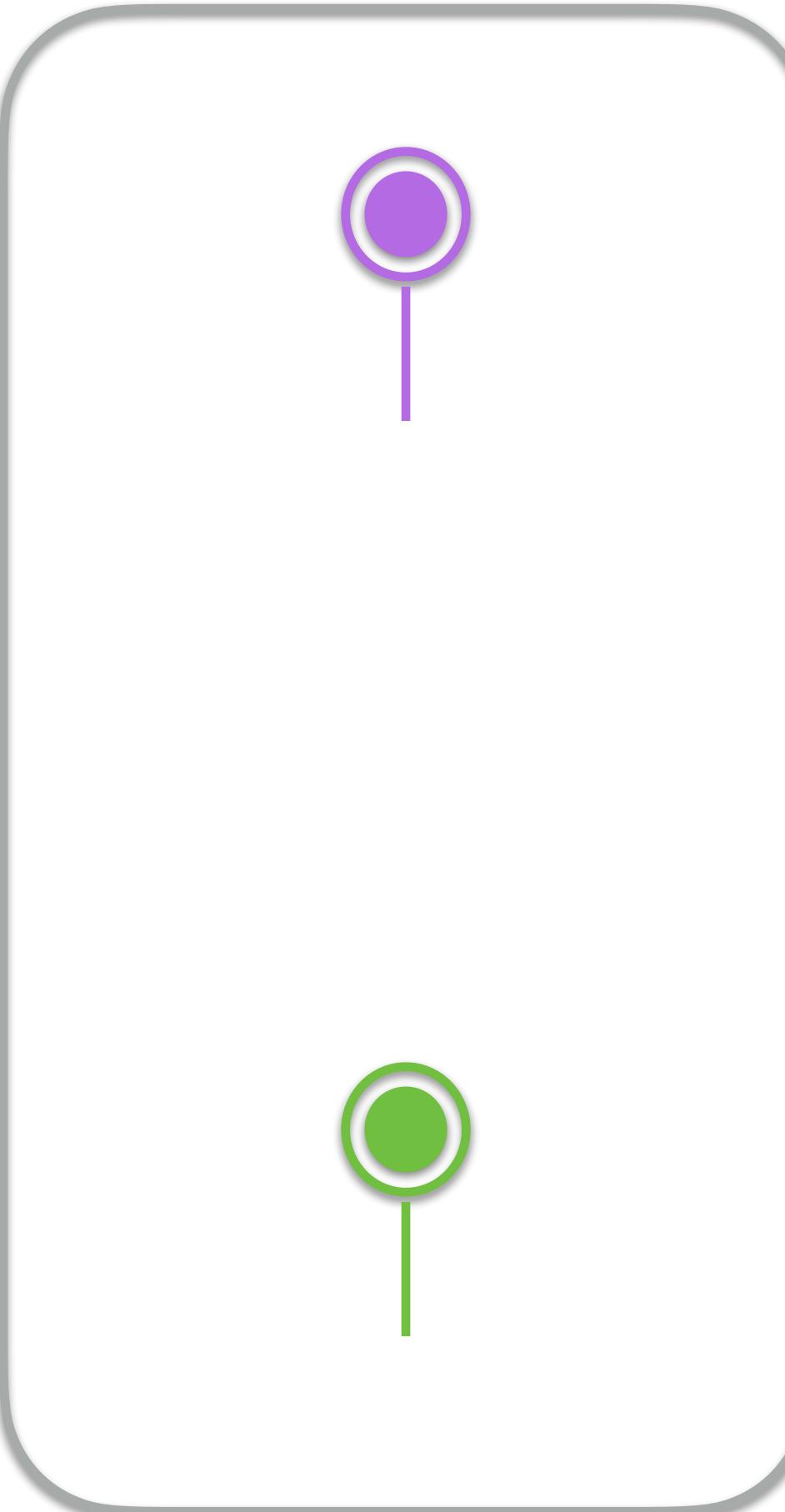
Local and remote commit



Merge

Conflict: local and remote commit

- Both updated, same content



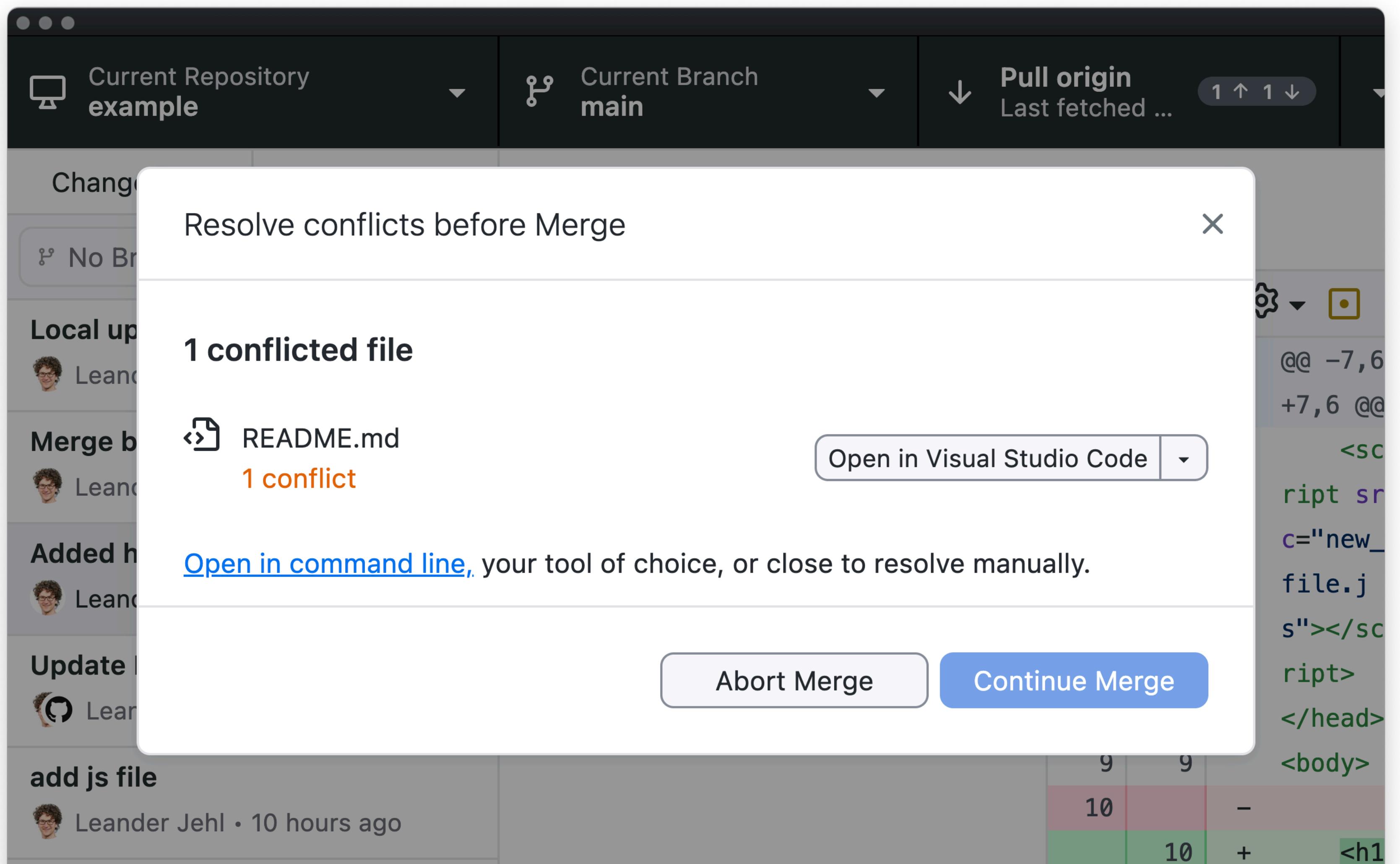
```
[* example git:(main) ✘ git pull
Auto-merging README.md
CONFLICT (content): Merge conflict in README.md
Automatic merge failed; fix conflicts and then commit the result.
[* example git:(main) ✘ git status
On branch main
Your branch and 'origin/main' have diverged,
and have 1 and 1 different commits each, respectively.
  (use "git pull" to merge the remote branch into yours)

You have unmerged paths.
  (fix conflicts and run "git commit")
    (use "git merge --abort" to abort the merge)

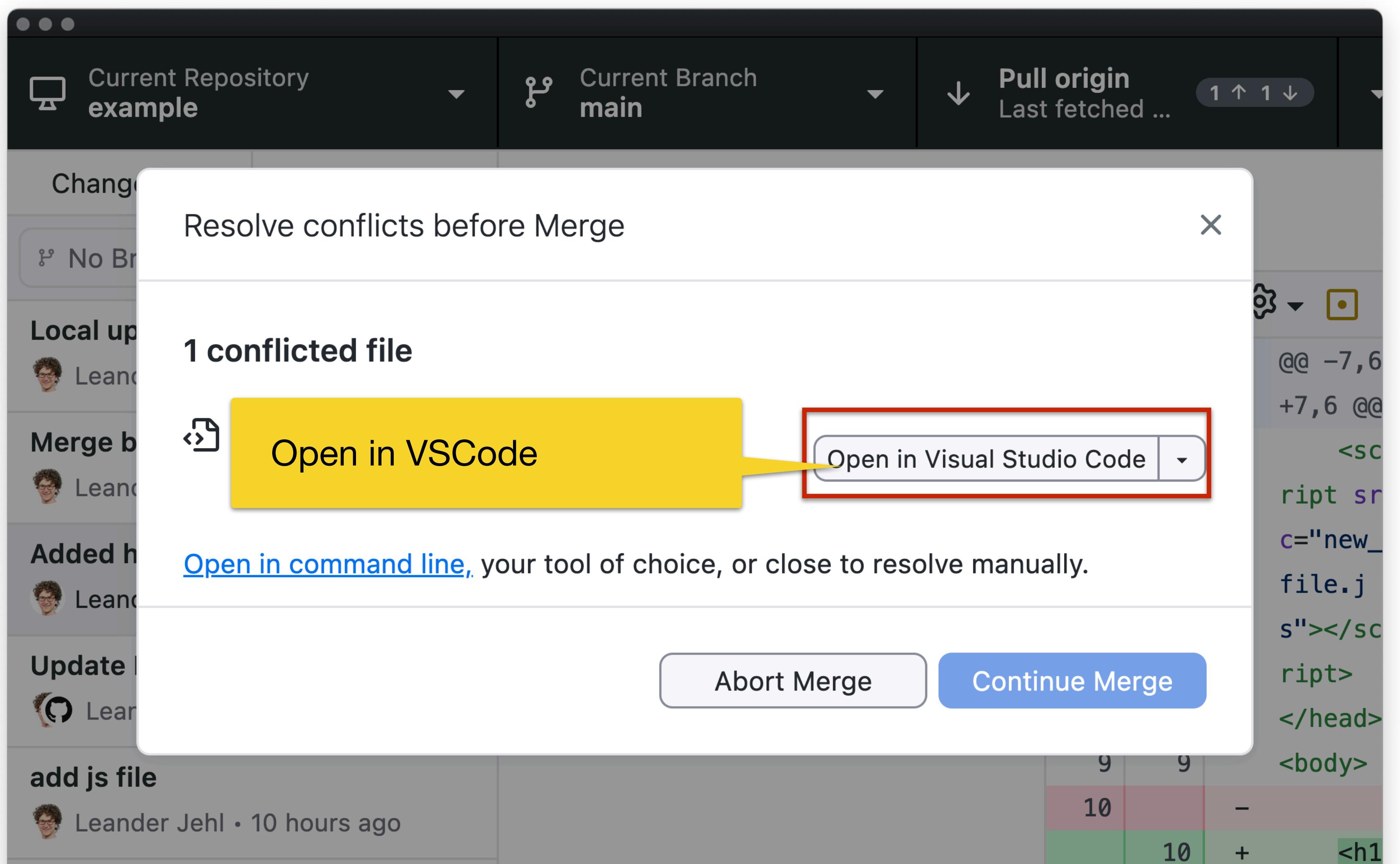
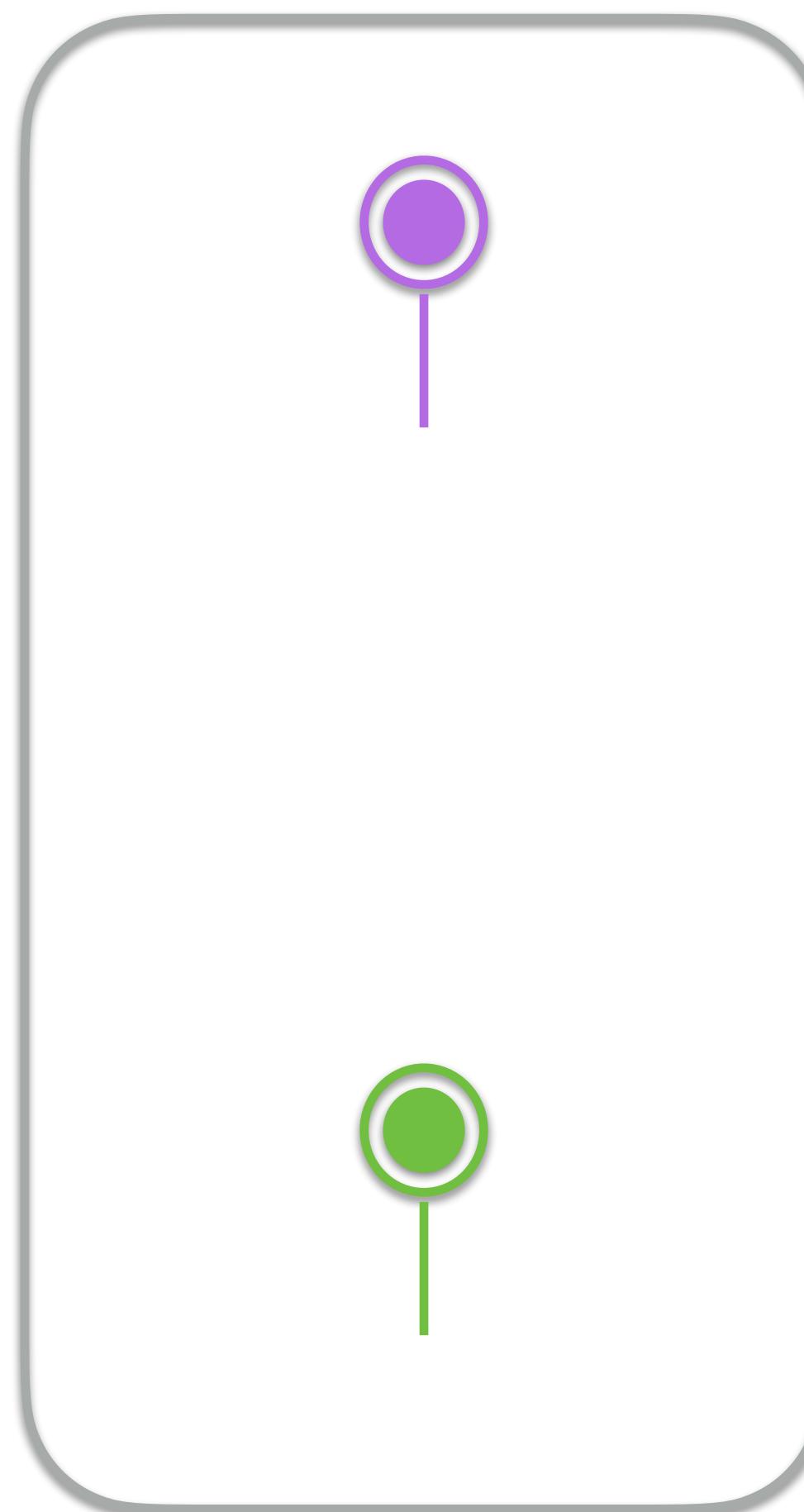
Unmerged paths:
  (use "git add <file>..." to mark resolution)
    both modified: README.md

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    .gitignore
    printsomestuff.py
```

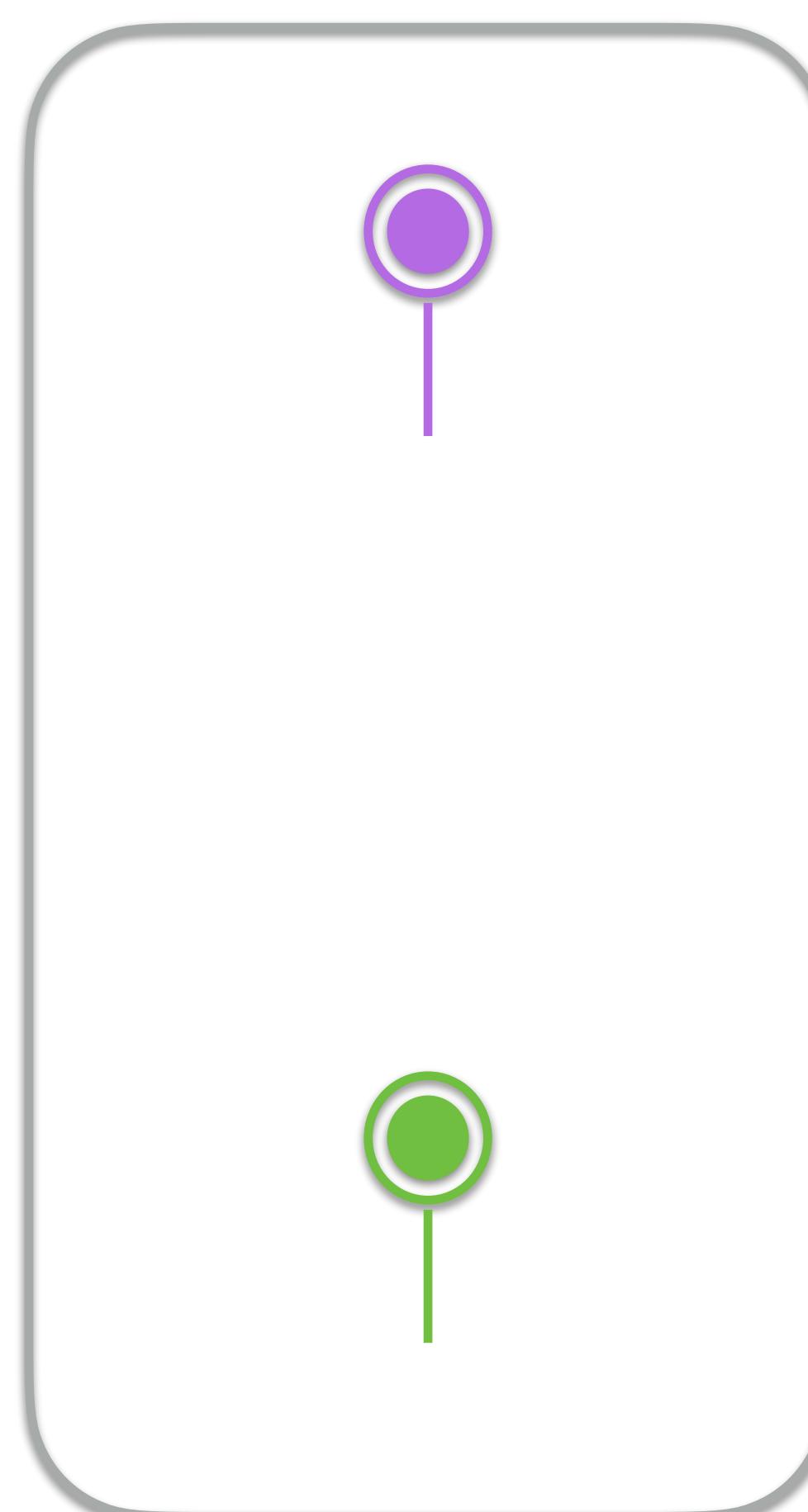
Conflict: local and remote commit



Conflict: local and remote commit



Conflict: local and remote commit



The screenshot shows a Git interface with a 'Changes' panel and a diff viewer.

Changes Panel:

- Merge branch 'main' of github.com:dat310-2025/example
- ✓ Continue
- Merge Changes
- ⓘ README.md

Diff Viewer:

```
ⓘ README.md (Working Tree) 5, ↗ ↘ ↑ ↓ ¶ M ⌂ ...  
ⓘ README.md > abc # Here is some more text that I added for an example.  
1 1 # example  
2 2 This repo is for making examples on git collaboration  
3+ <<<<< HEAD (Current Change)  
3 4  
4 5 Here is some more text that I added example. And more on the same.  
7+ >>>>>  
+ 97d333ab638e1f7308dab34dae188184bdd420f  
+ e (Incoming Change)  
5 8
```

A yellow callout box points to the 'Merge Changes' item in the Changes panel, with the text: "Files that need manual resolution".

Conflict: local and remote commit



The screenshot shows a Git merge interface with the following details:

- SOURCE CONTROL** tab is selected.
- CHANGES** section shows a merge branch 'main' from `github.com:dat310-2025/example`. A blue button labeled **✓ Continue** is visible.
- Merge Changes** section lists changes for `README.md`:
 - Local update to `readm...` (Local)
 - Update `READM...` (origin/main)
 - Merge branch 'main' of `github...`
 - Update `README.md...` Leand...
 - Added heading to `index.html...`
- GRAPH** section shows the commit history:
 - Local update to `readm...` (Local)
 - Update `READM...` (origin/main)
 - Merge branch 'main' of `github...`
 - Update `README.md...` Leand...
 - Added heading to `index.html...`
- README.md** file content:

```
1 # example
2 This repo is for making examples on git collaboration
3 <<<<< HEAD (Current Change)
4
5 Here is some more text that I added for an example. And more on the
6 same line.
7 =====
8 >>>>> 97d333ab638e1f7308dab34dae188184bdd420fe (Incoming Change)
```

Resolve in Merge Editor

Conflict: local and remote commit



- Accept Current change
 - Take the version from local
- Accept Incoming
 - Take the version from remote
- Accept both
 - Both below each other

Select one of these for each conflict.

Accept Current Change | Accept Incoming Change | Accept Both Changes | Compare Changes

<<<<< · HEAD (Current Change)

Here is some more text that I added for an example. And more on the same line.

=====

>>>>> · 97d333ab638e1f7308dab34dae188184bdd420fe (Incoming Change)

Can also edit here before, after choosing.

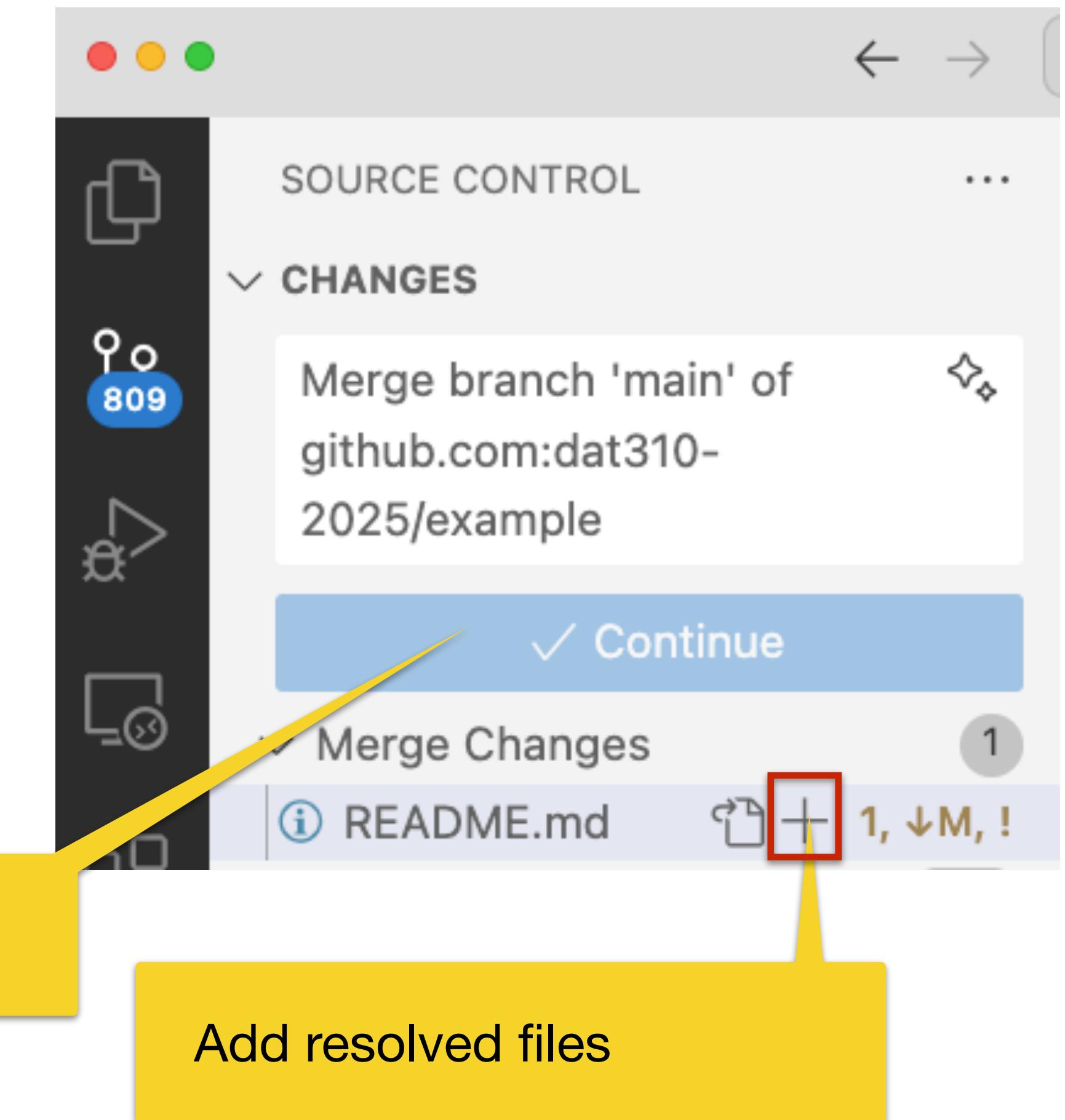
Conflict: local and remote commit



Merge Procedure

- Resolve all conflicts
 - Choose current/incoming ...
- Add resolved files
- Create merge commit

Commit



Add resolved files

Stash

Remote commit and local changes

- Unable to pull
- Stash changes
- pull
- Apply changes from stash

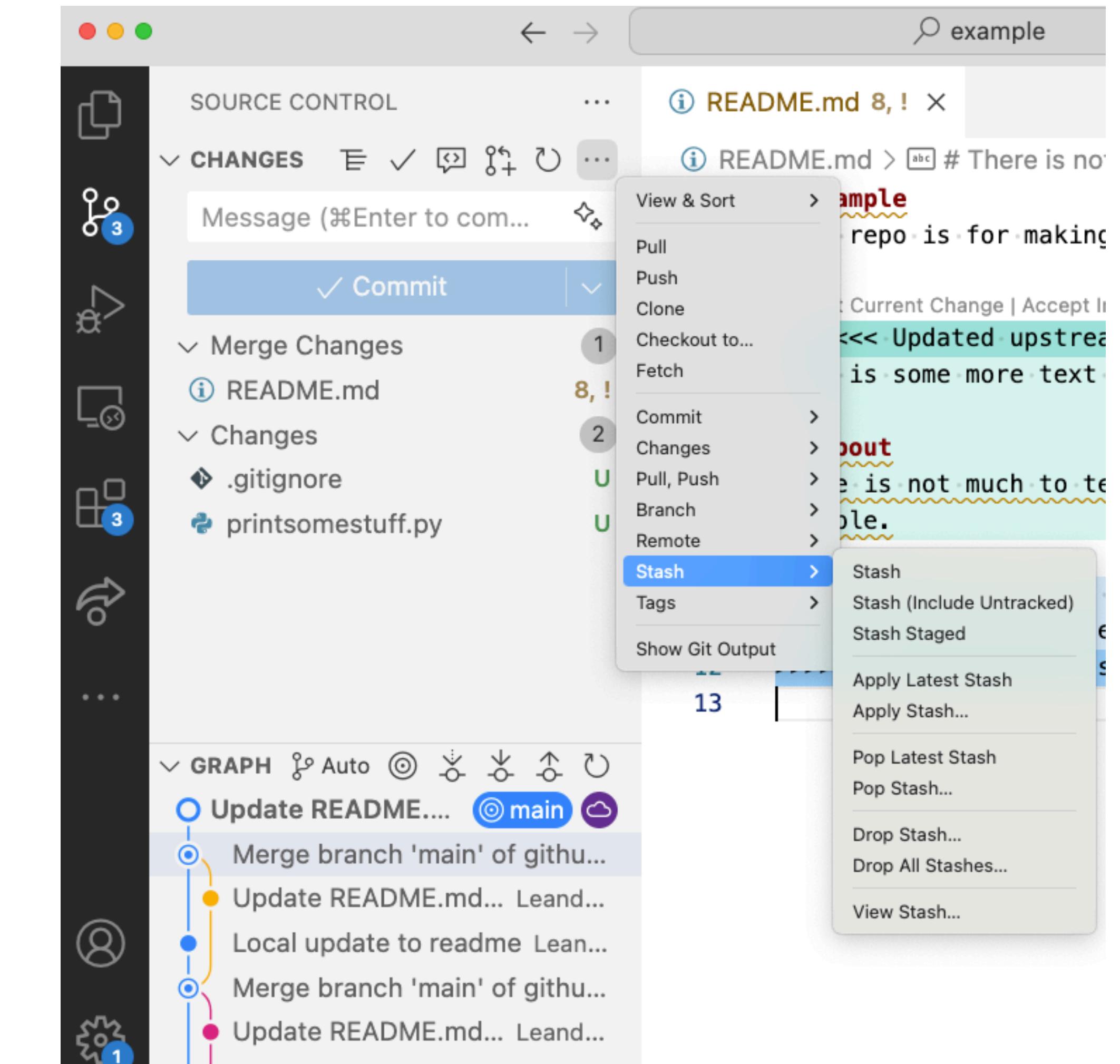
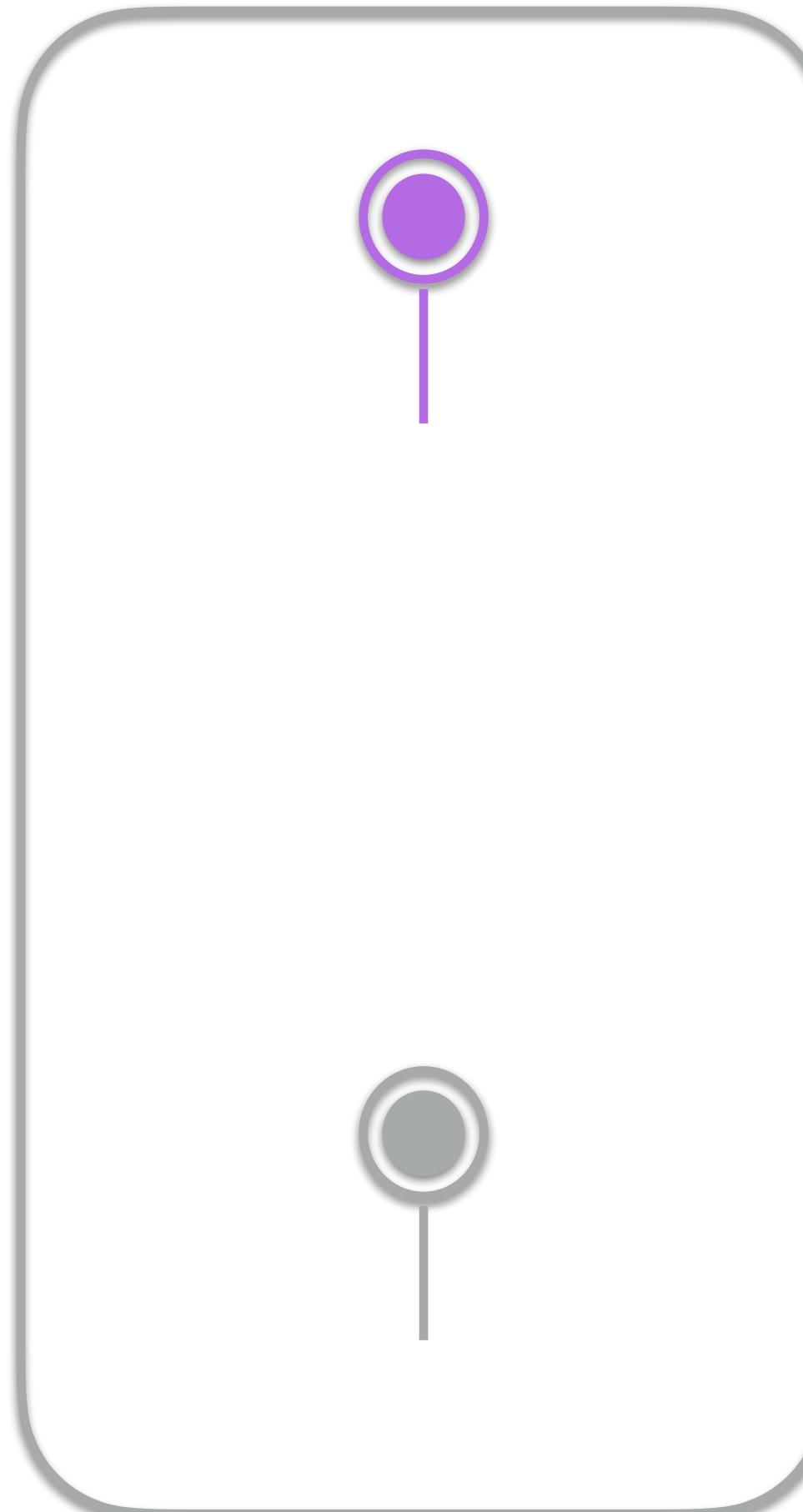
```
example — leanderjehl@Leanders-MacBook-Pro-2 — -zsh — 72x...
../2025/example

[→ example git:(main) ✘ git pull
Updating 981653c..1e5b515
error: Your local changes to the following files would be overwritten by
merge:
    README.md
Please commit your changes or stash them before you merge.
Aborting
[→ example git:(main) ✘ git stash
Saved working directory and index state WIP on main: 981653c Merge branc
h 'main' of github.com:dat310-2025/example
[→ example git:(main) ✘ git pull
Updating 981653c..1e5b515
Fast-forward
    README.md | 5 +---+
    1 file changed, 4 insertions(+), 1 deletion(-)
[→ example git:(main) ✘ git stash pop
Auto-merging README.md
CONFLICT (content): Merge conflict in README.md
On branch main
Your branch is up to date with 'origin/main'.

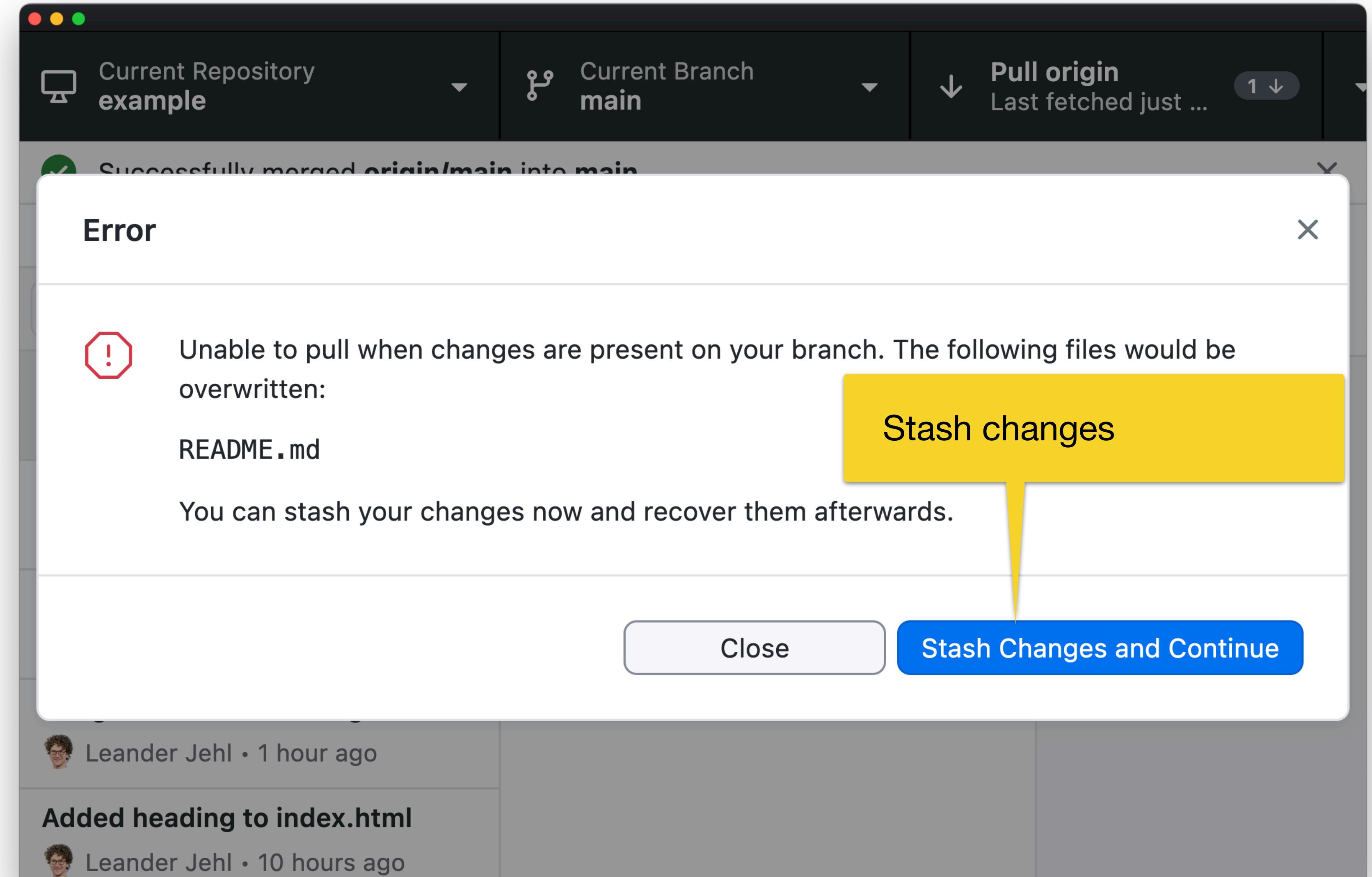
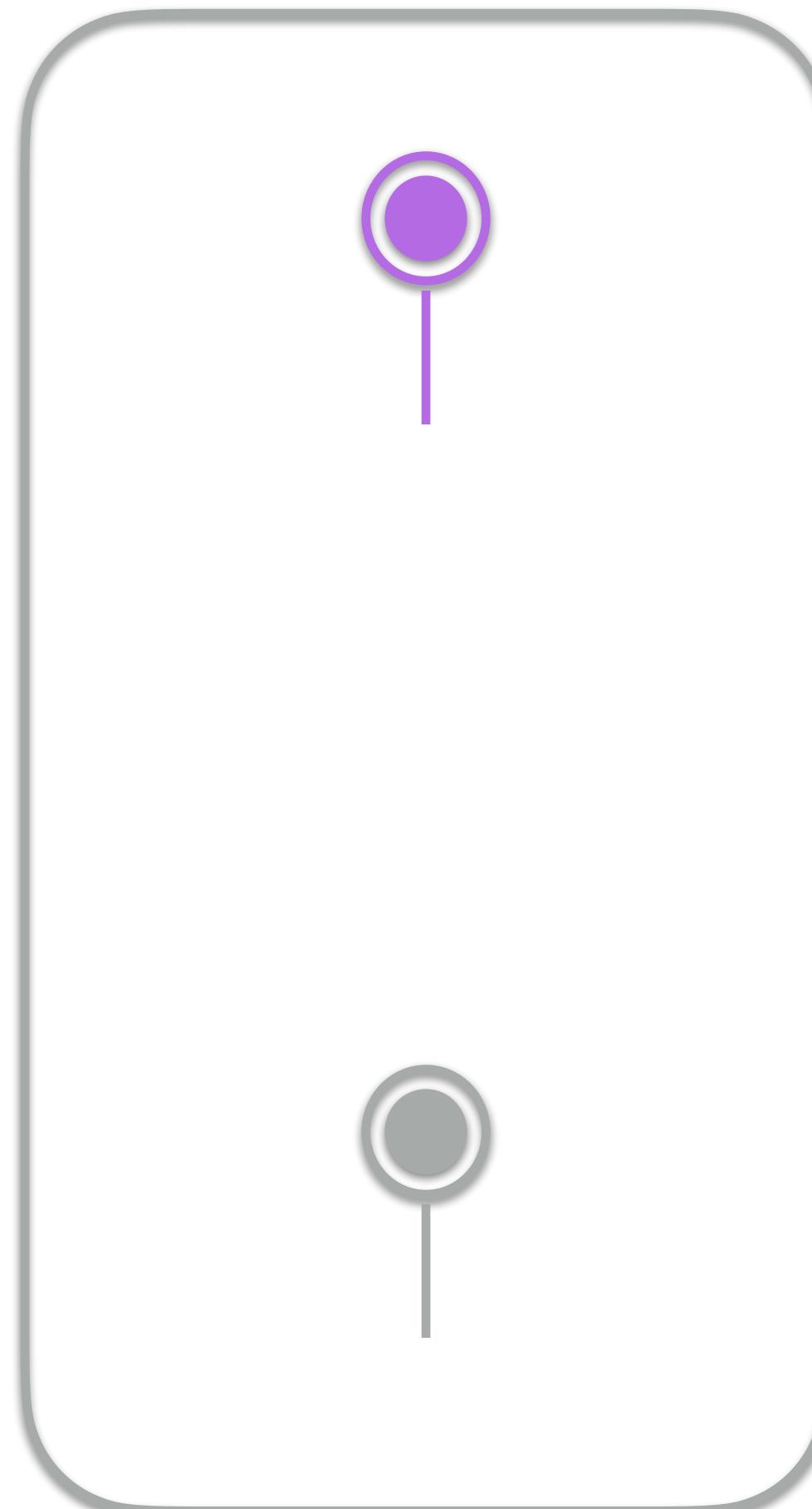
Unmerged paths:
  (use "git restore --staged <file>..." to unstage)
  (use "git add <file>..." to mark resolution)
```

Remote commit and local changes

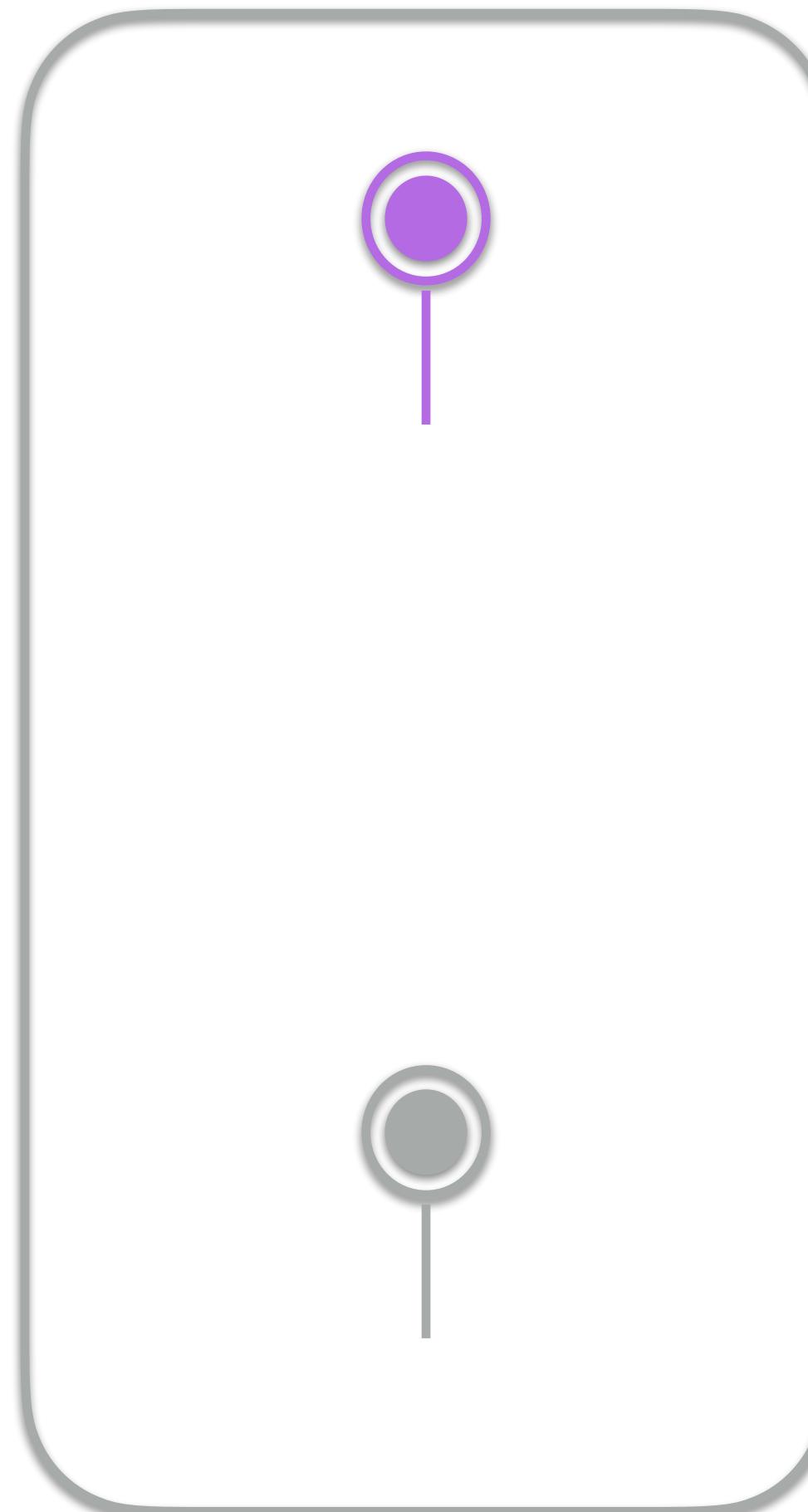
- Unable to pull
- Stash changes
- pull
- Apply changes from stash



Remote commit and local changes



Remote commit and local changes



Current Repository **example** Current Branch **main** Fetch origin Last fetched 1 minute ago

Successfully merged **origin/main** into **main**

Changes History

0 changed files

Stashed Changes

Summary (required)

Description

No local changes

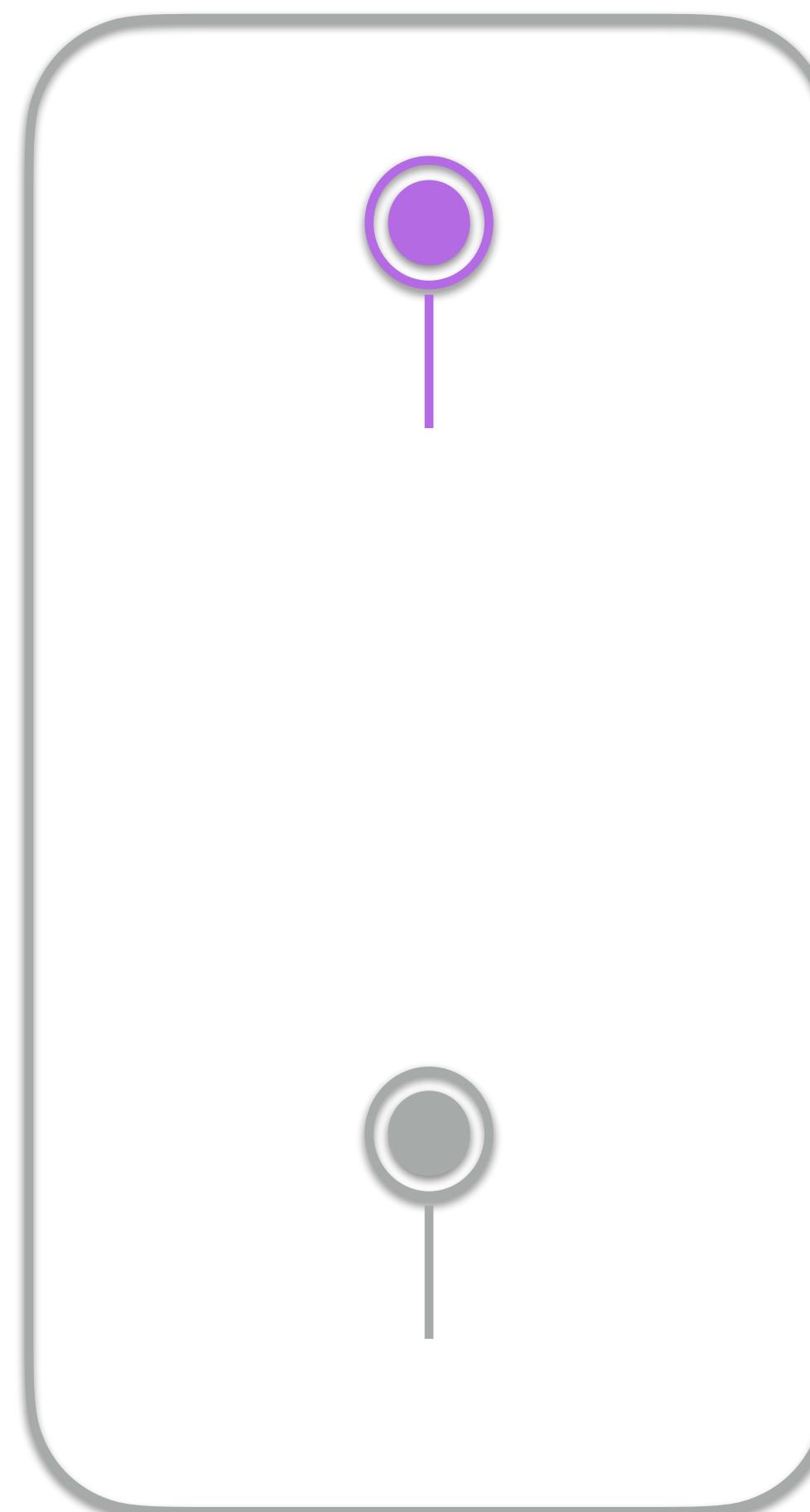
There are no uncommitted changes in this repository. Here are some friendly suggestions

View stashed changes

View your stashed changes
You have 3 changes in progress that you have not yet committed.
When a stash exists, access it at the bottom of the Changes tab to the left.

View stash

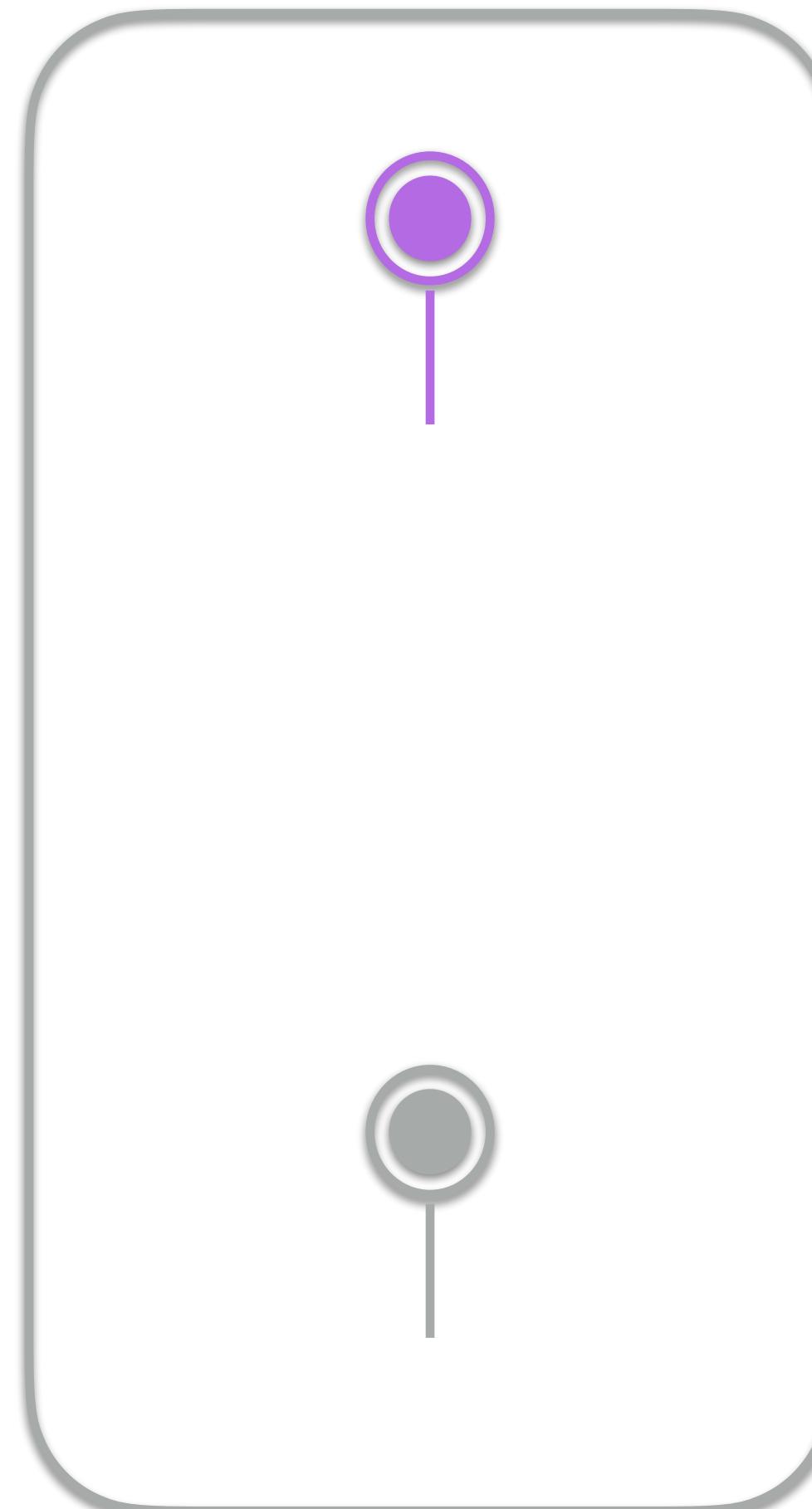
Remote commit and local changes



The screenshot shows a Git interface with the following details:

- Current Repository:** example
- Current Branch:** main
- Last fetched:** Fetch origin (2 minutes ago)
- Message:** Successfully merged origin/main into main
- Changes:** 3 changed files
 - 3 changed files
 - .gitignore
 - printsomestuff.py
 - README.md
- Stashed Changes:** A blue button labeled "Stashed Changes" with a right arrow.
- Summary (required):** A summary card for a user named "John Doe".
- Description:** A text input field for a commit message.
- Conflict:** A yellow callout box pointing to the README.md file in the changes list, indicating a conflict.
- Buttons:** Discard and Restore. The Restore button is highlighted in blue.
- Yellow Callouts:**
 - "Re-apply stashed changes" points to the Stashed Changes button.
 - "Restore will move your stashed files to the Changes list." points to the Restore button.

Remote commit and local changes



The screenshot shows a Source Control interface with a dark theme. The left sidebar has icons for file, repository (with 3), search, and more. The main area shows a file named 'README.md' with a conflict. The conflict text is:

```
# example
This repo is for ma
Here is some more text that I added for an example.
Accept Current Change | Accept Incoming Change | Accept Both Changes | Compare Changes
<<<< Updated upstream (Current Change)
```

A yellow callout bubble points to the conflict text with the text "Resolve conflicts". Another yellow callout bubble points to the "Accept Current Change" button with the text "Stage file to mark conflict resolution."

At the bottom, the graph shows a merge commit from 'main' branch into the current branch, with a local update to 'readme' file.

Remote commit and local changes

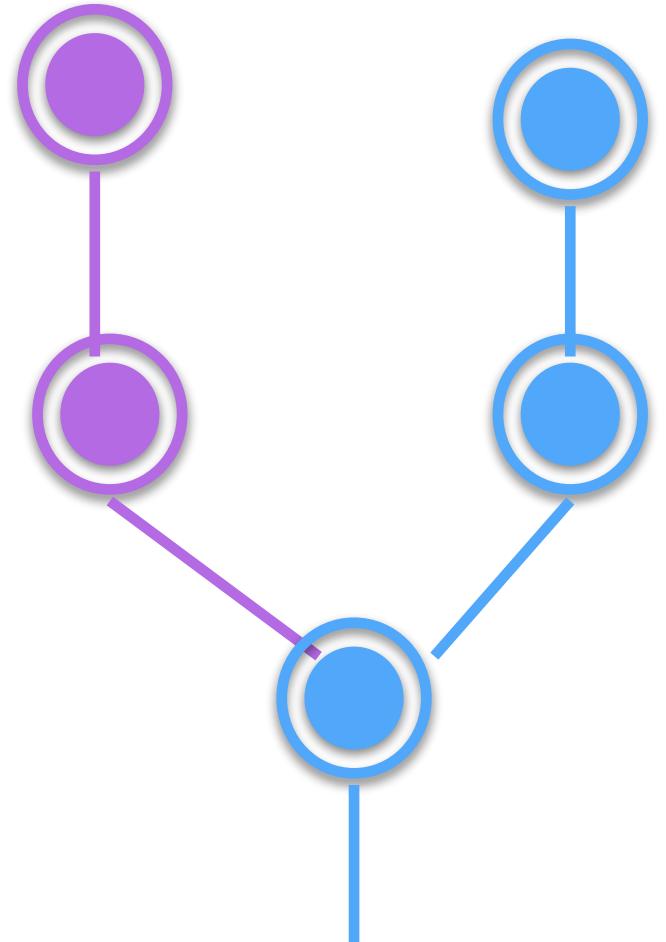


- You can **stash changes** to allow **pull**
- You can **re-apply changes** from stash
 - Need to delete from stash later
 - Pop: re-apply and delete
- Resolve conflict
- **Stage resolved files**



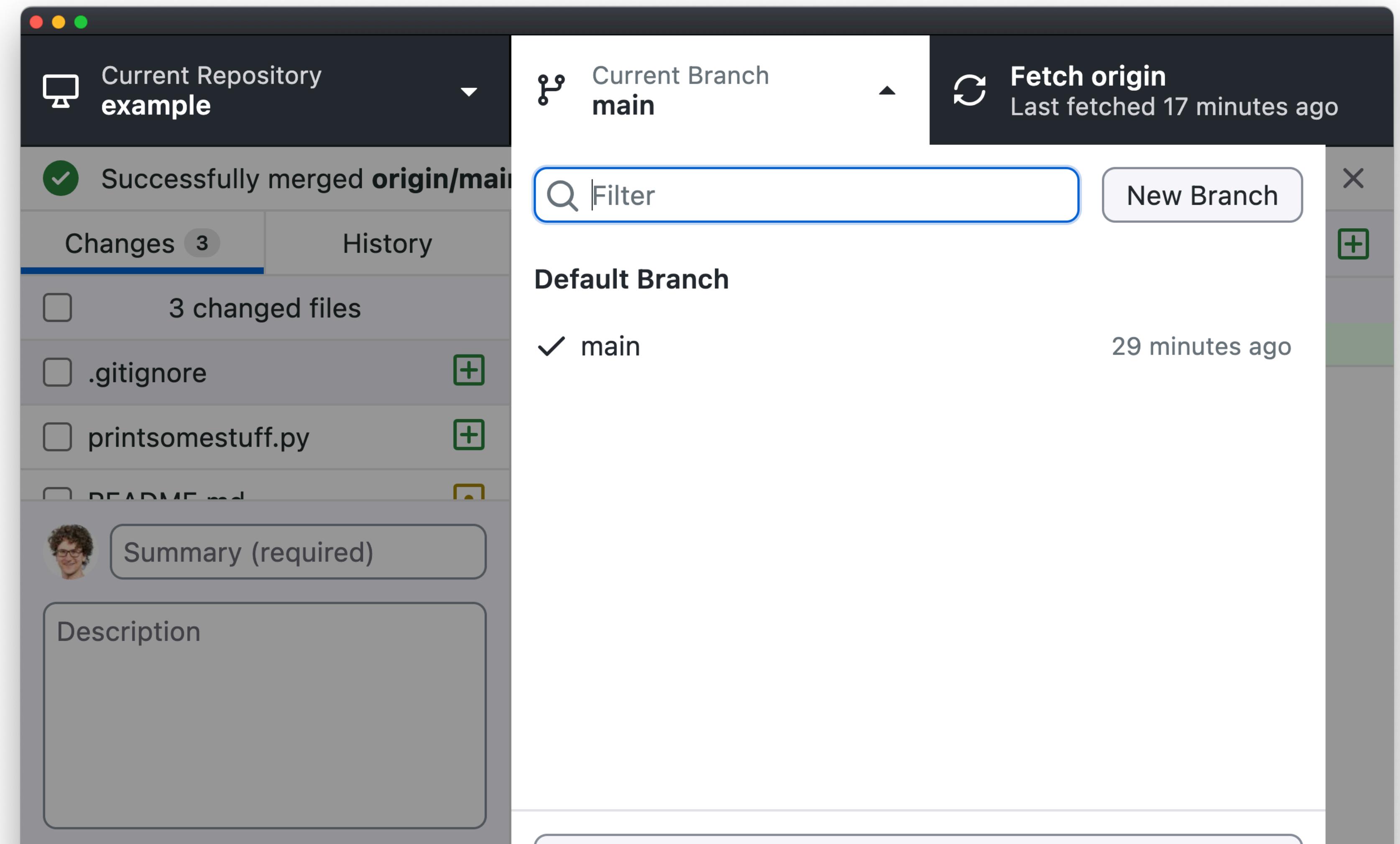
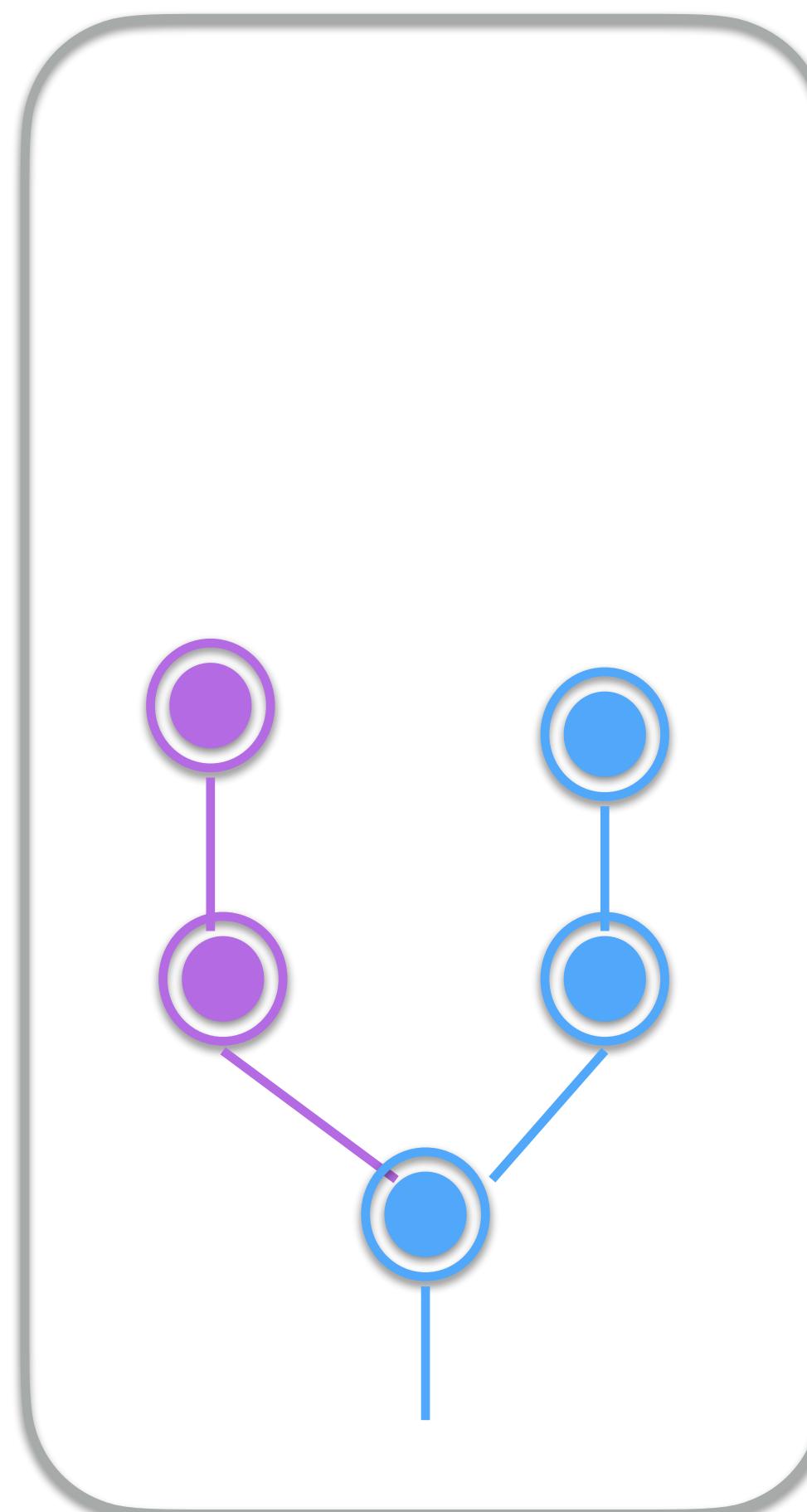
Branches

A branch is a version of the repository

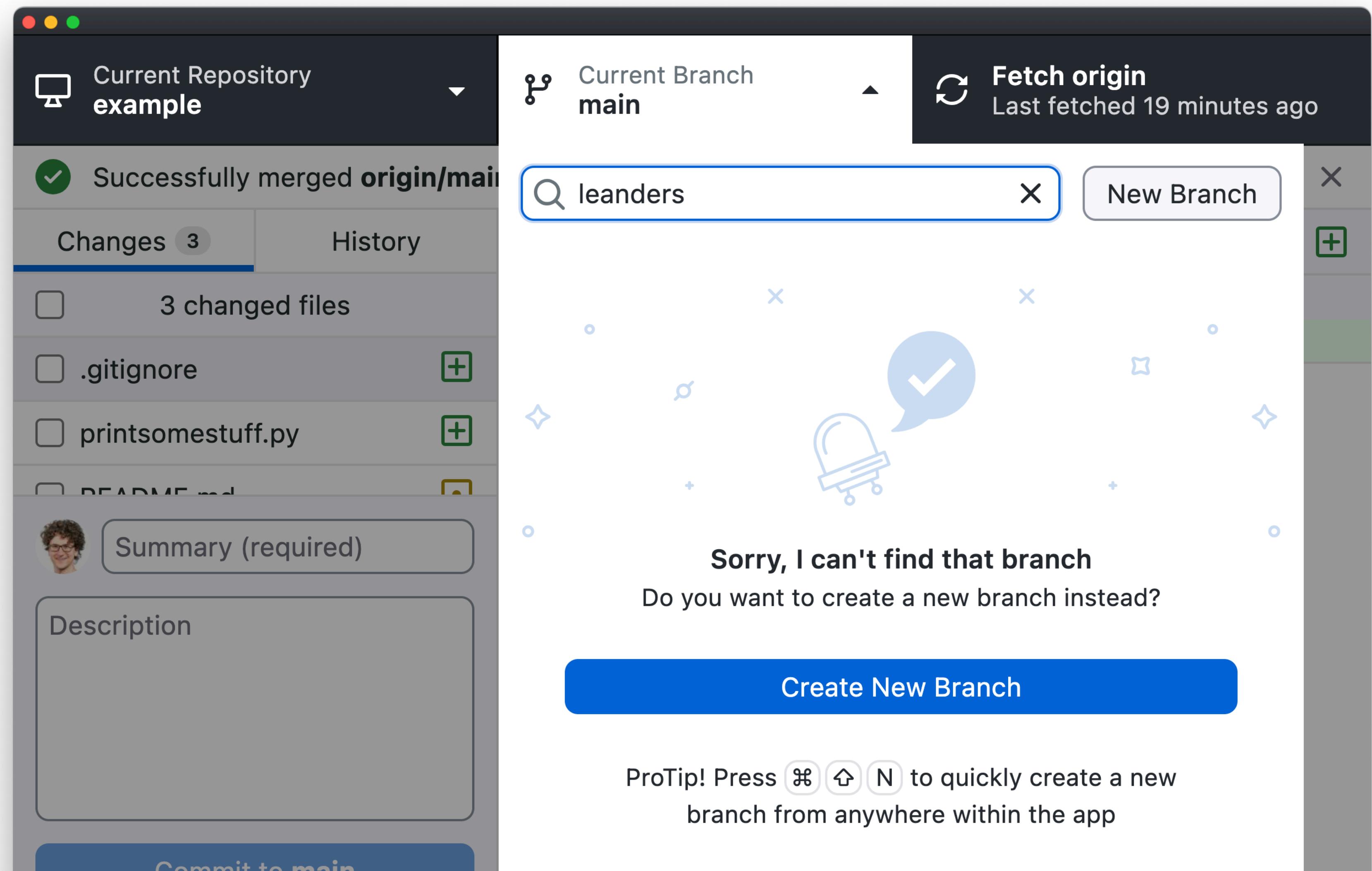
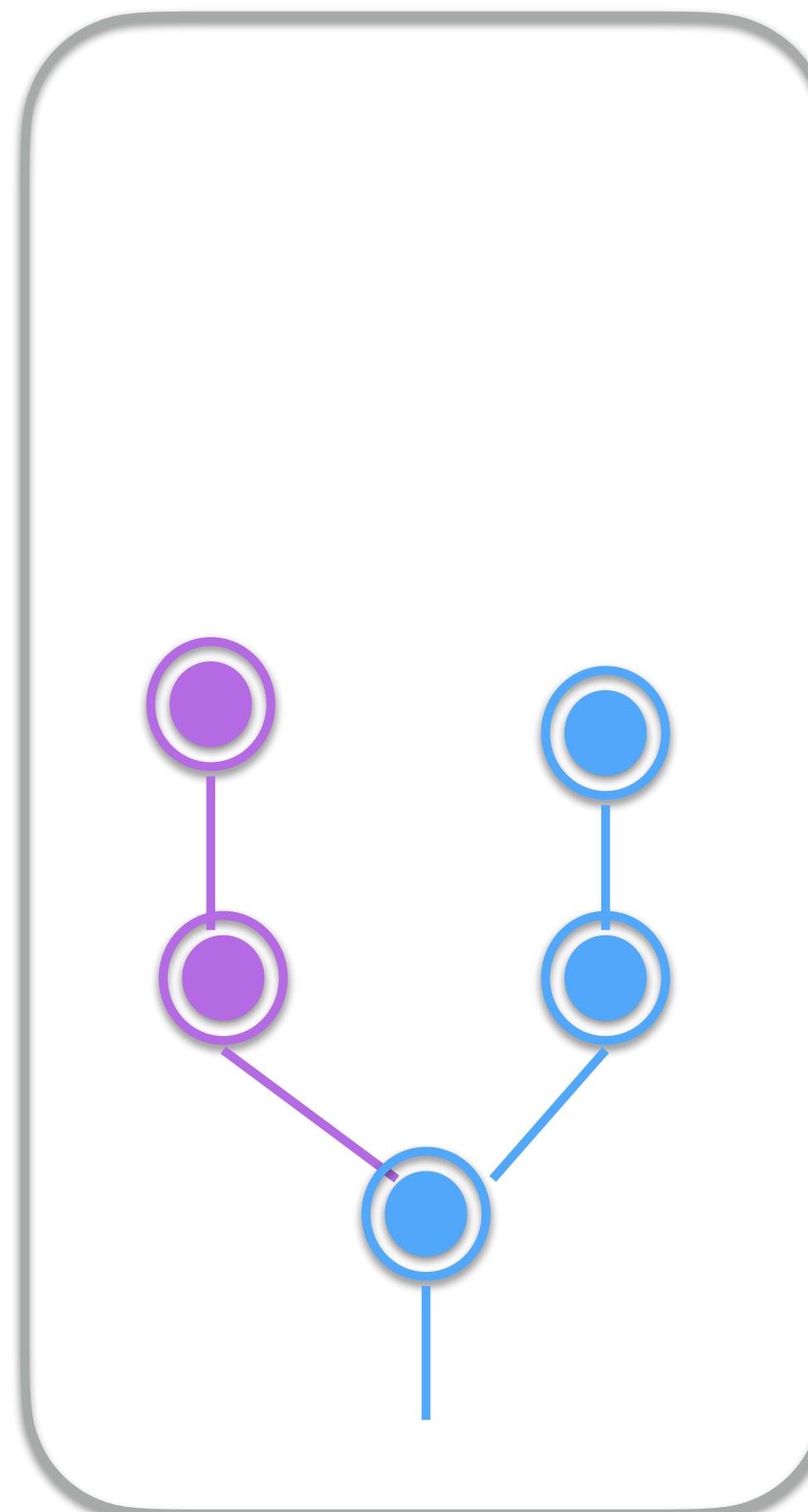


- Idea:
- Every developer (or feature) has one branch
- Working features are merged into main branch
- Main branch is merged into other branches

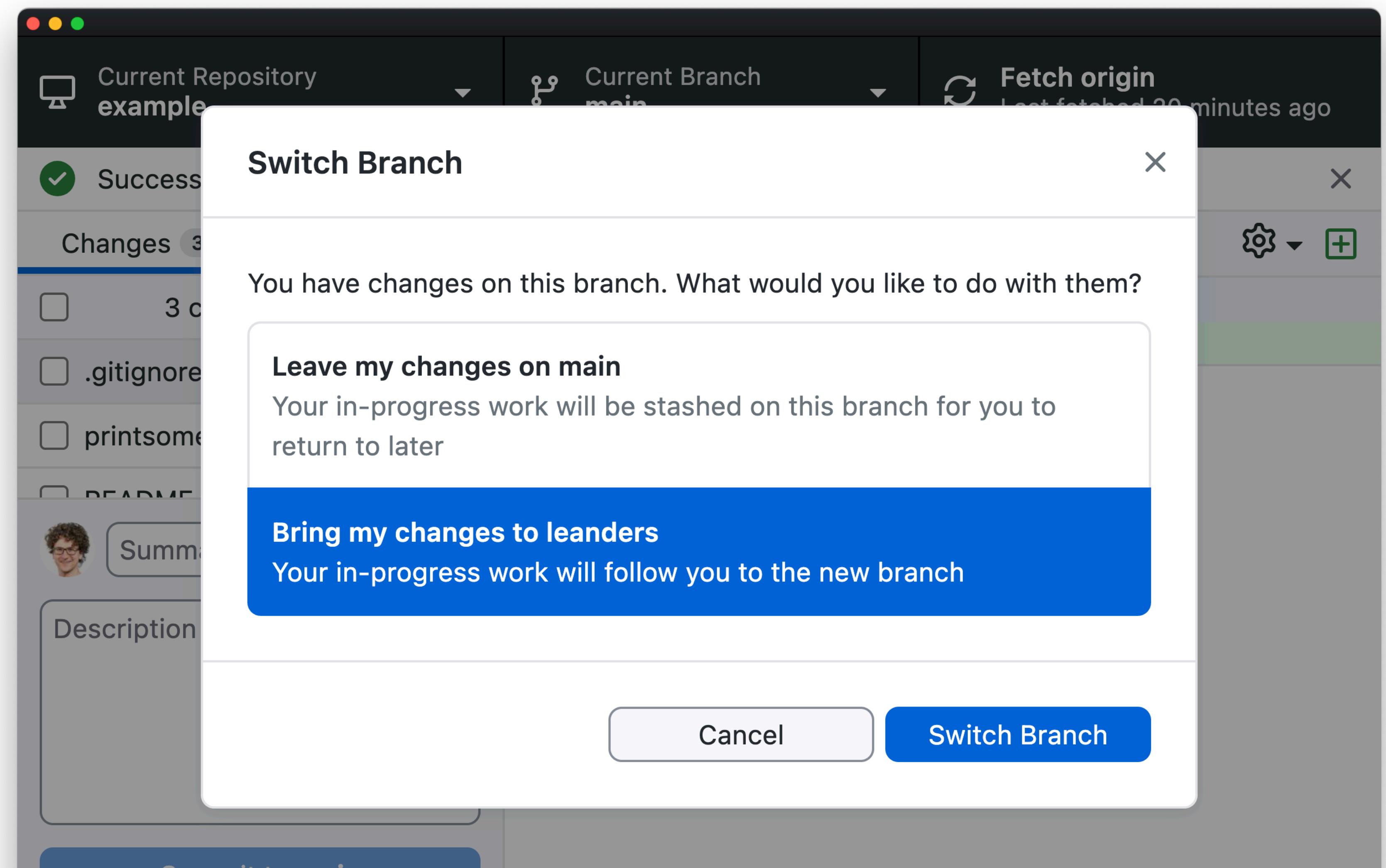
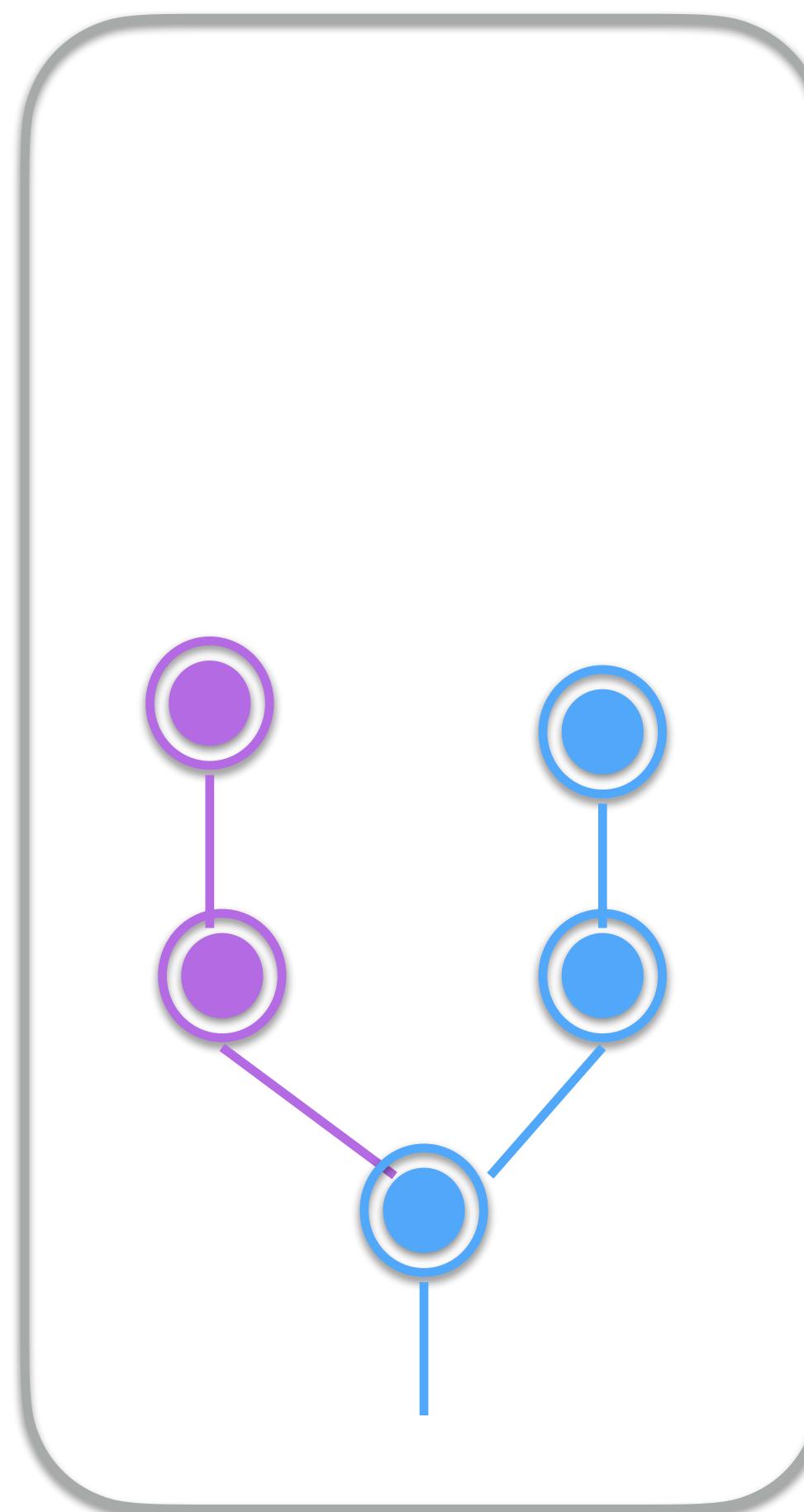
A branch is a version of the repository



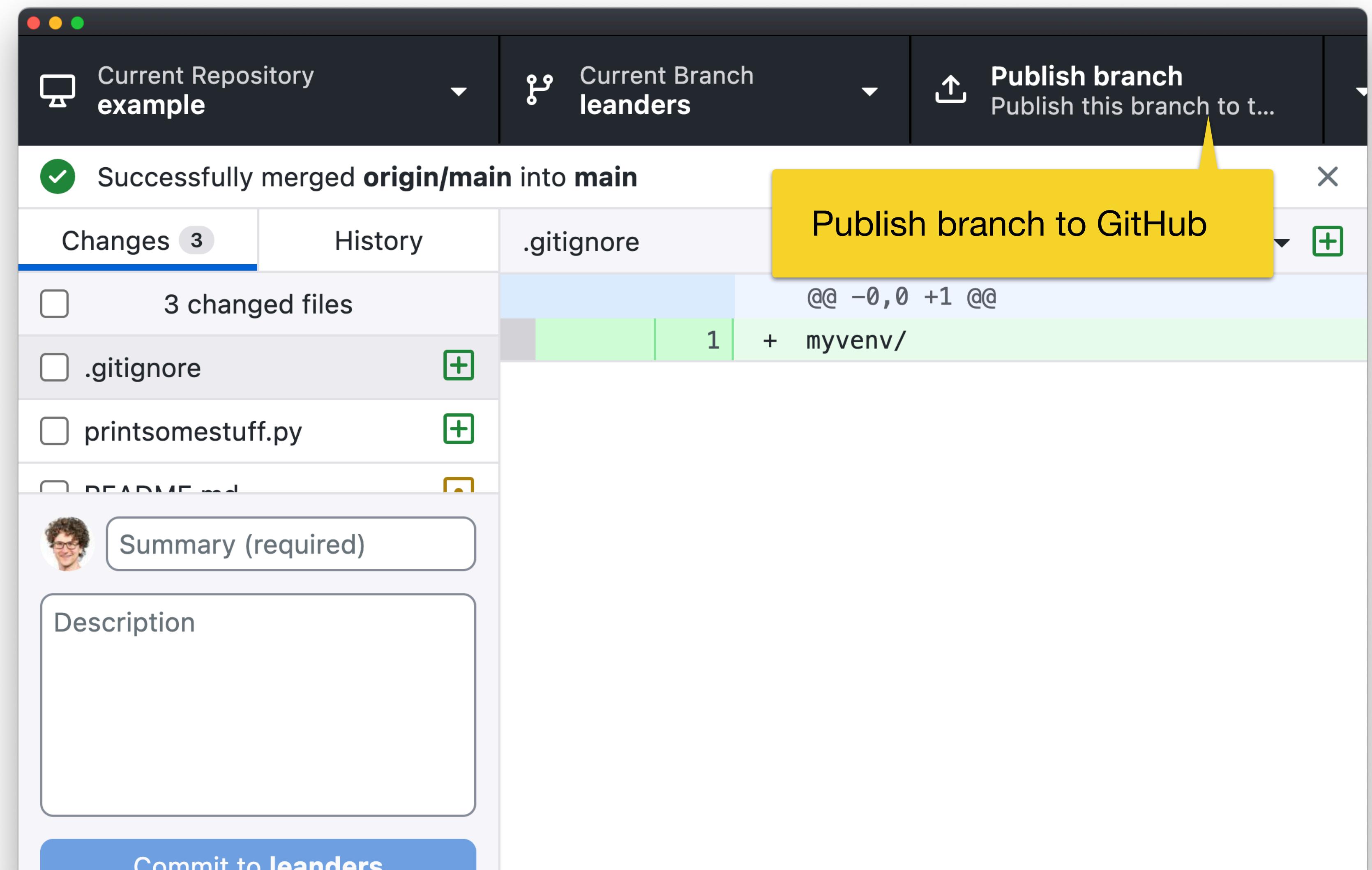
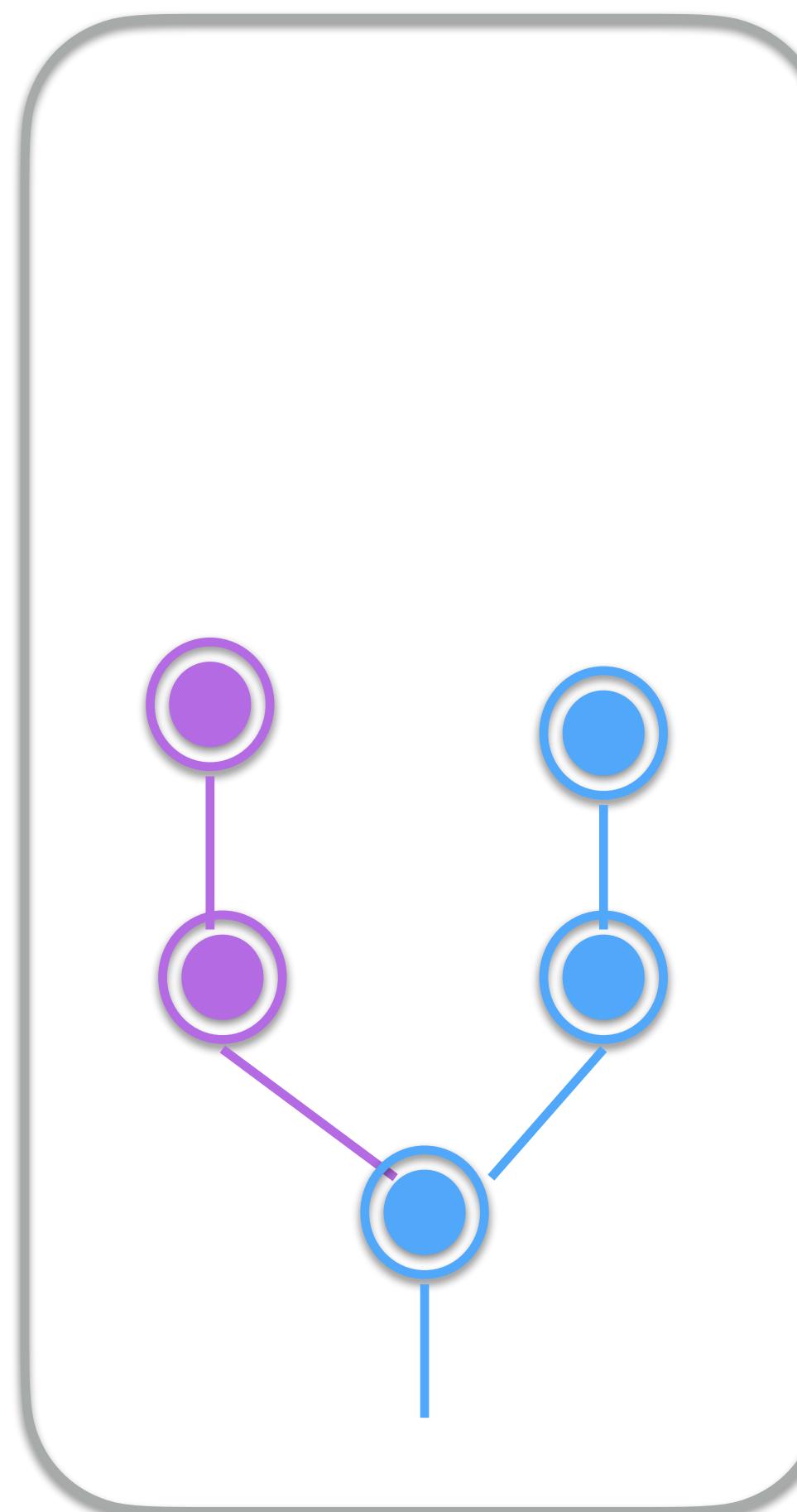
A branch is a version of the repository



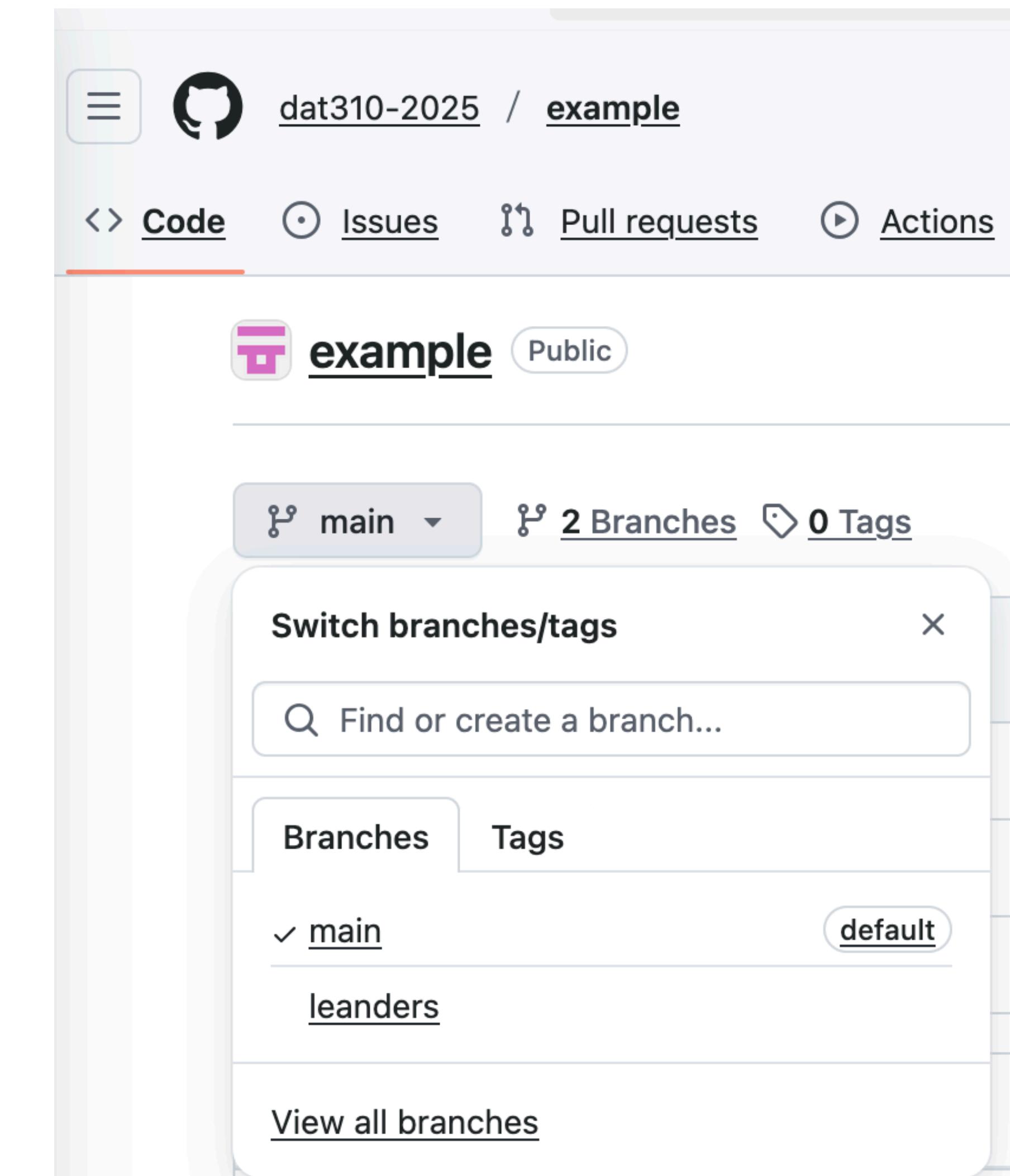
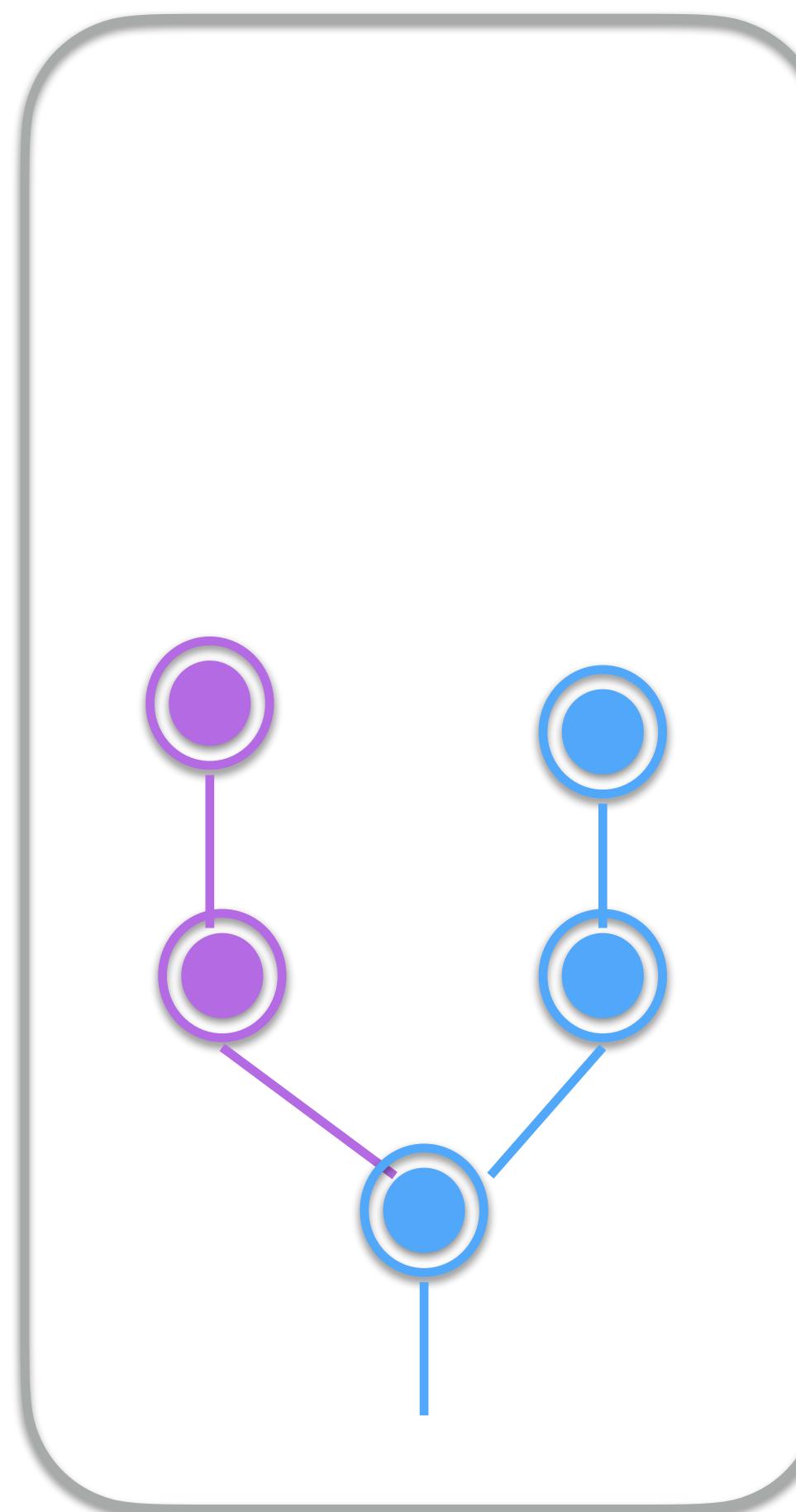
A branch is a version of the repository



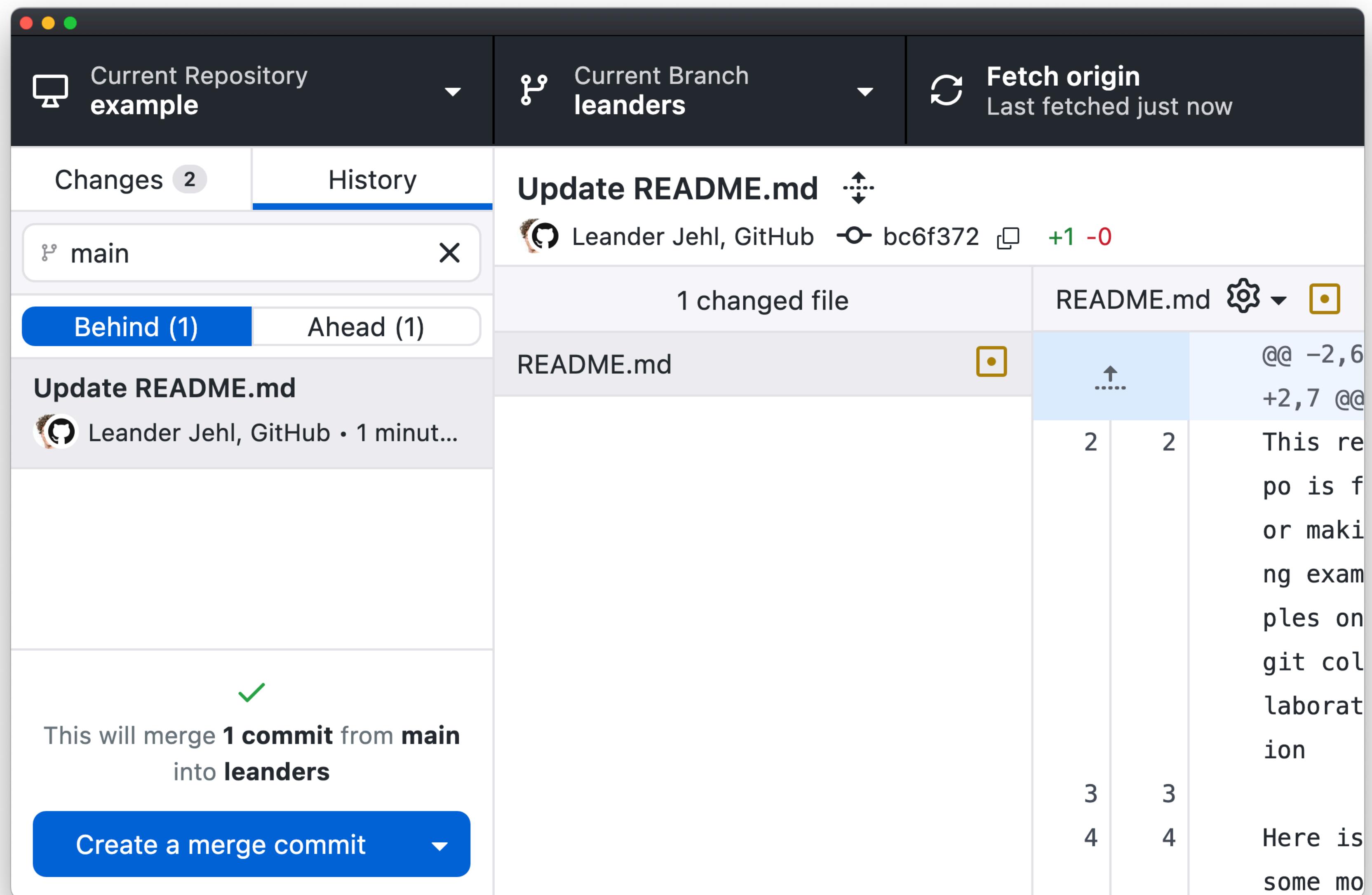
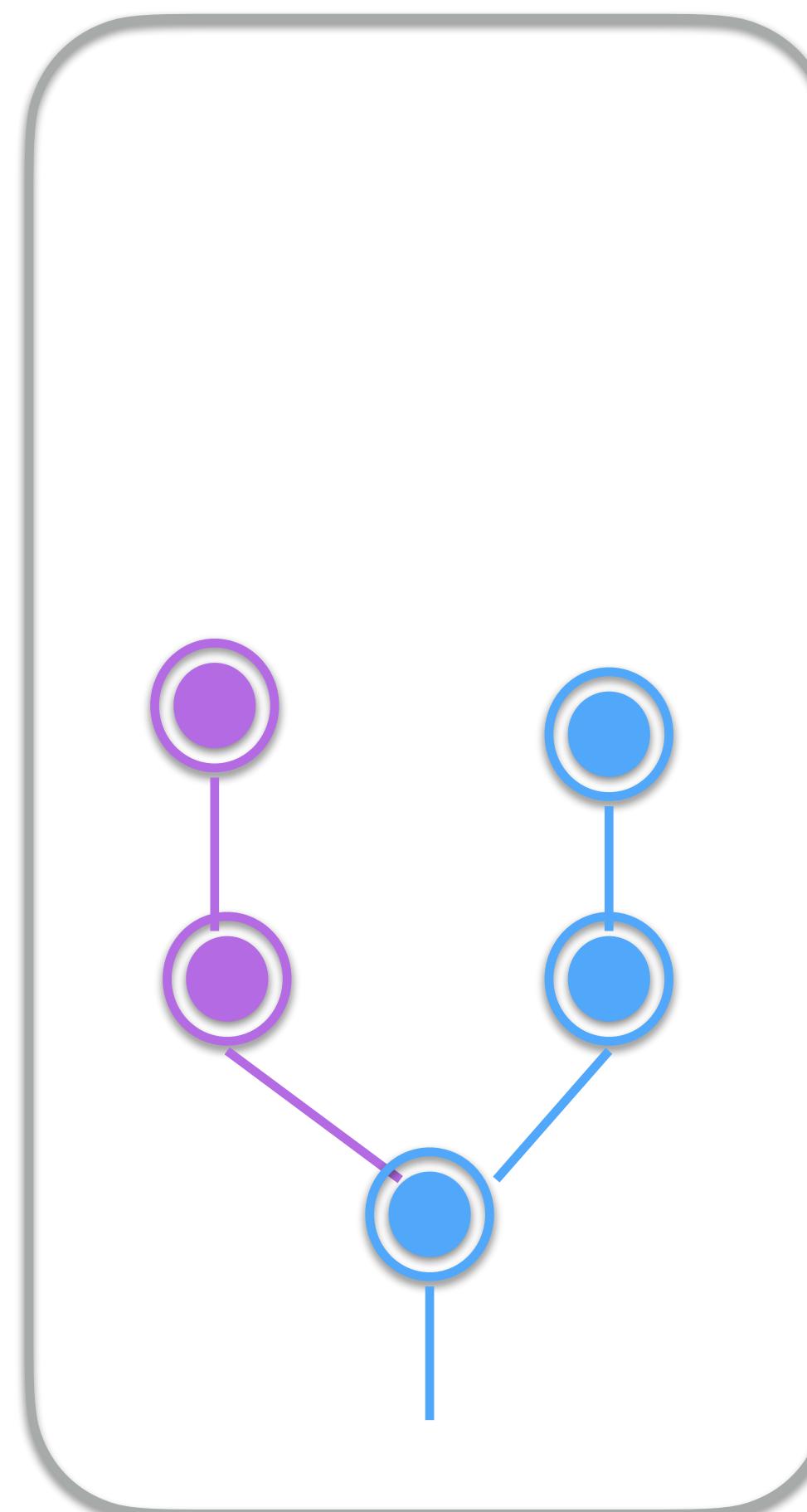
A branch is a version of the repository



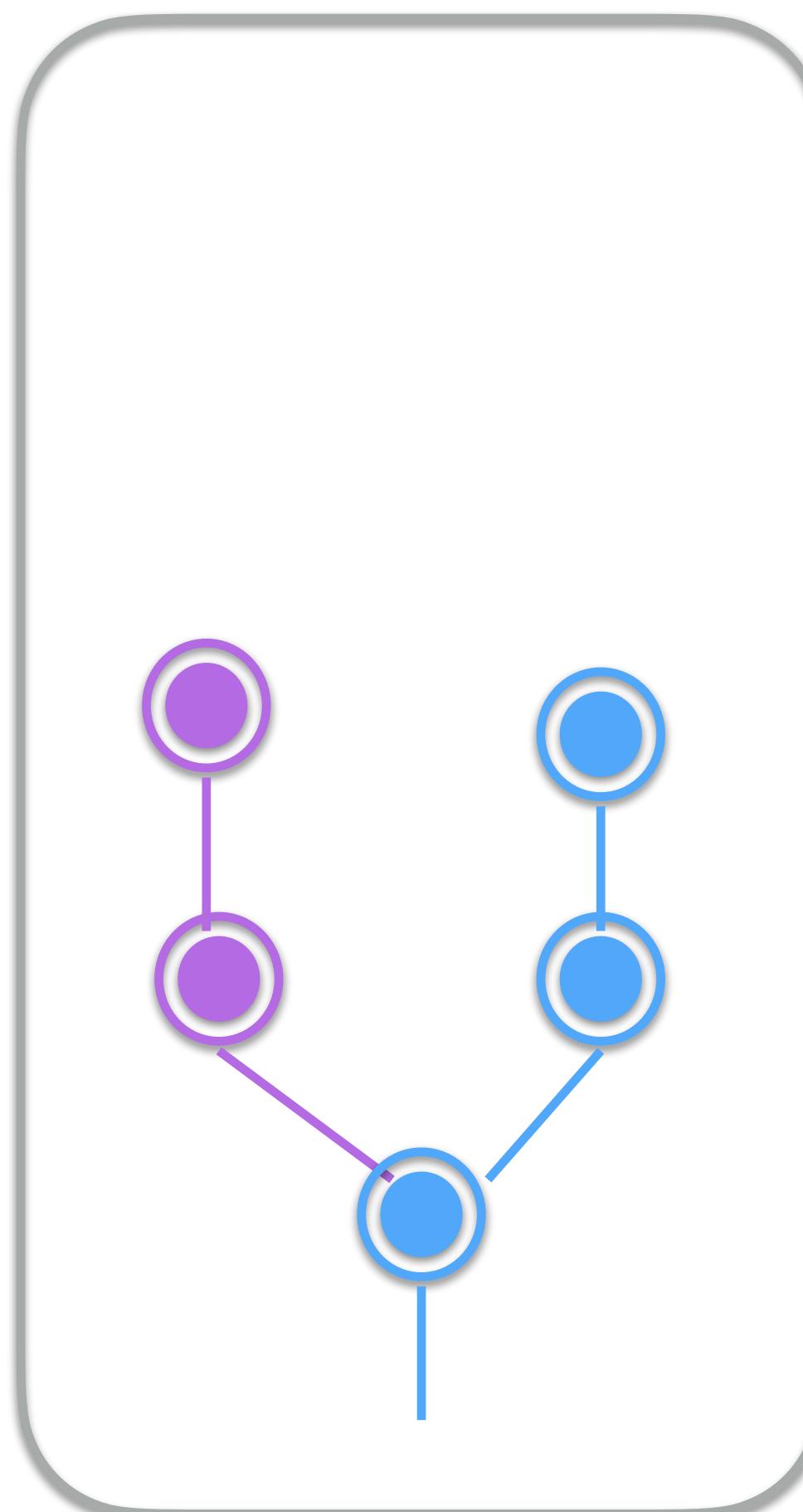
A branch is a version of the repository



A branch is a version of the repository

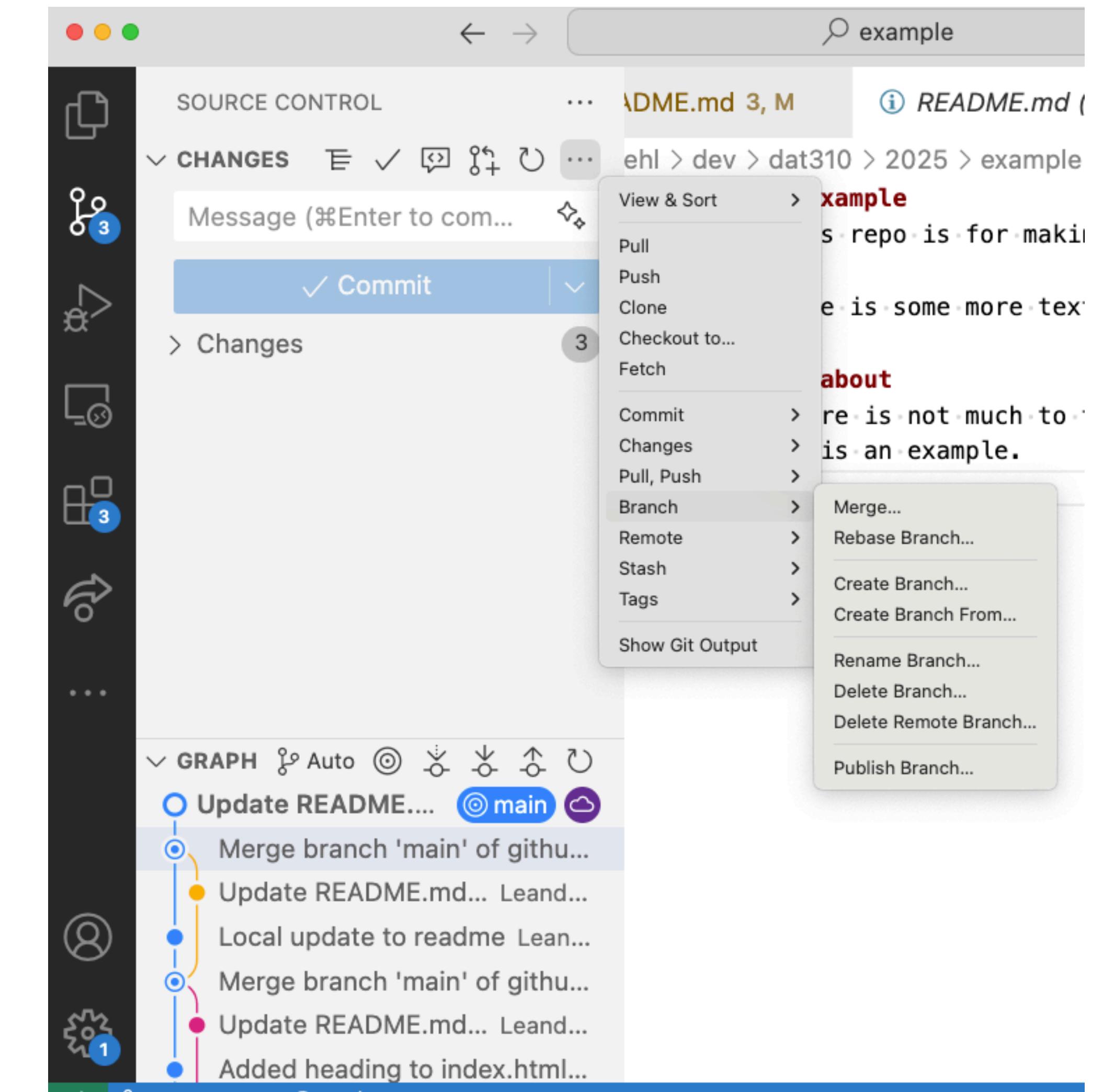


A branch is a version of the repository

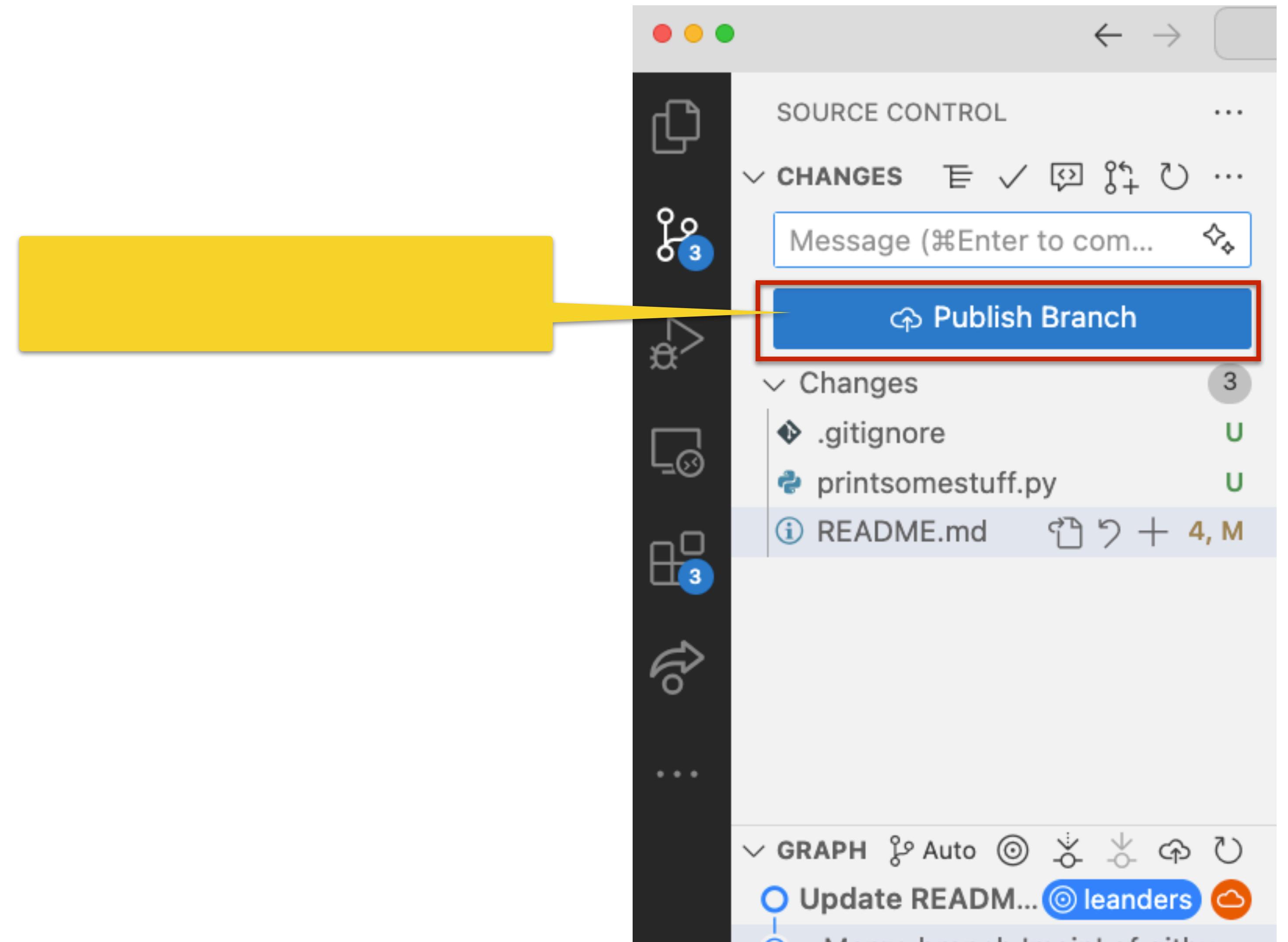
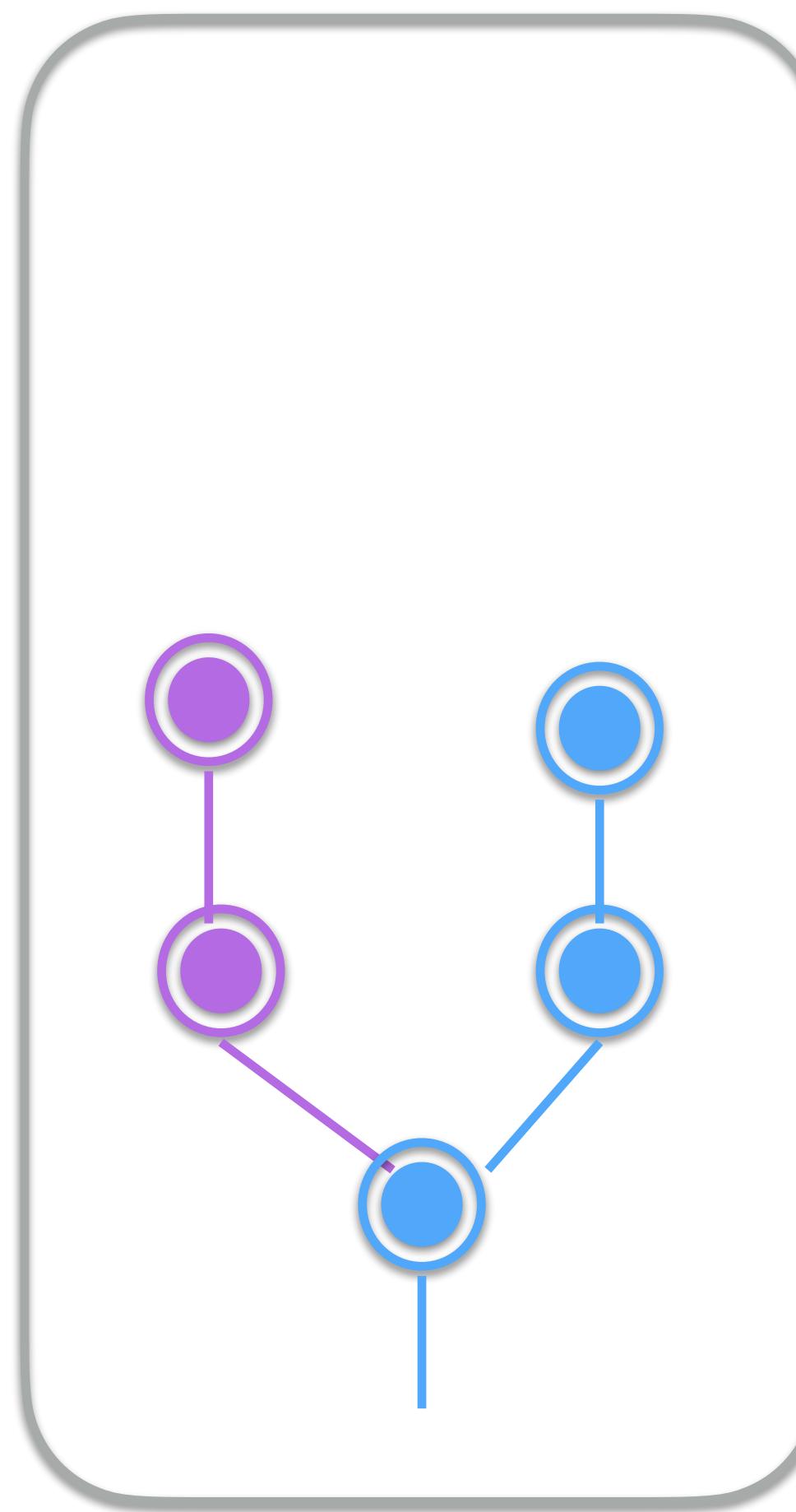


VSCode

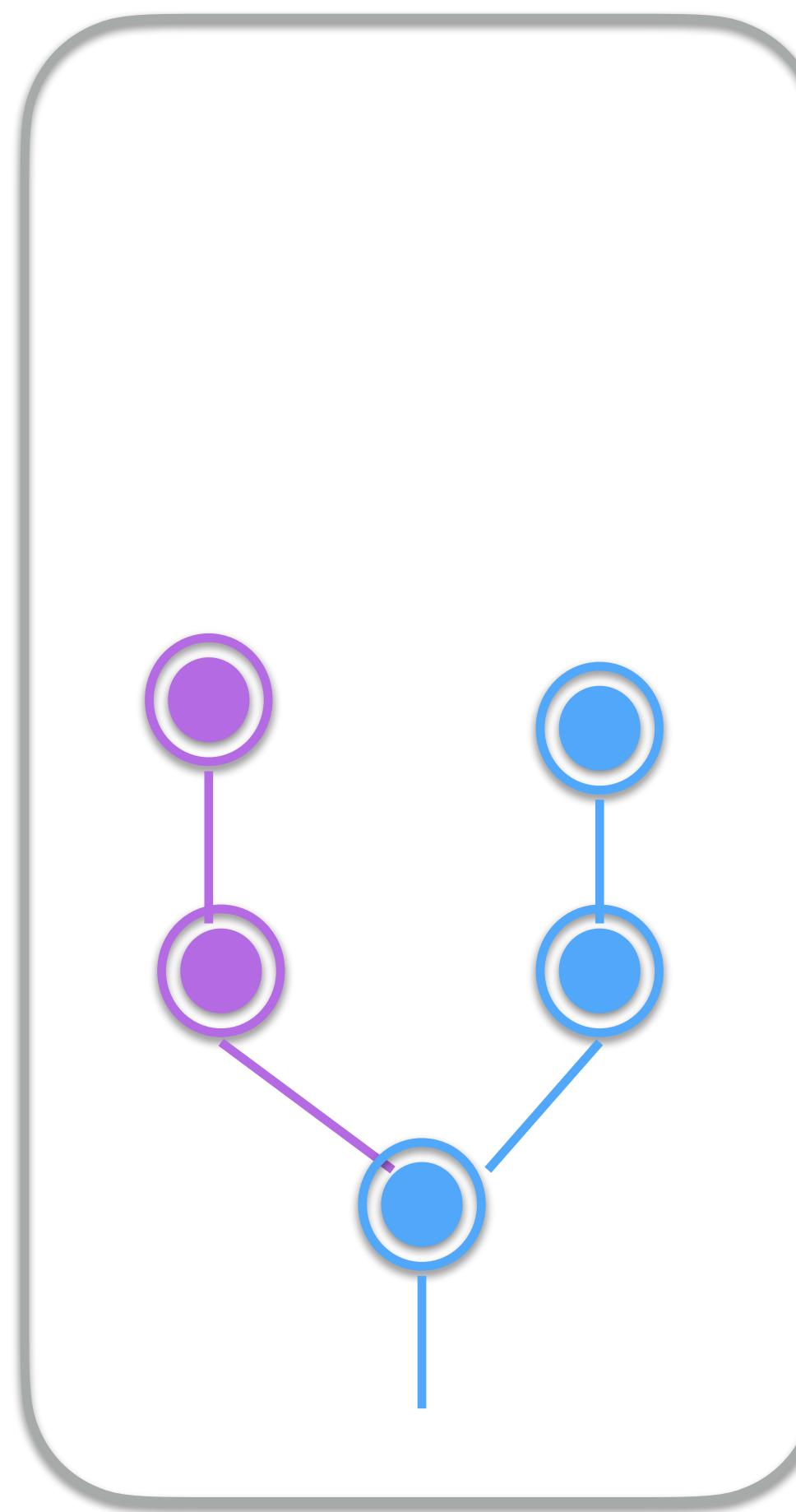
- ...
- Branch
- Create Branch



A branch is a version of the repository



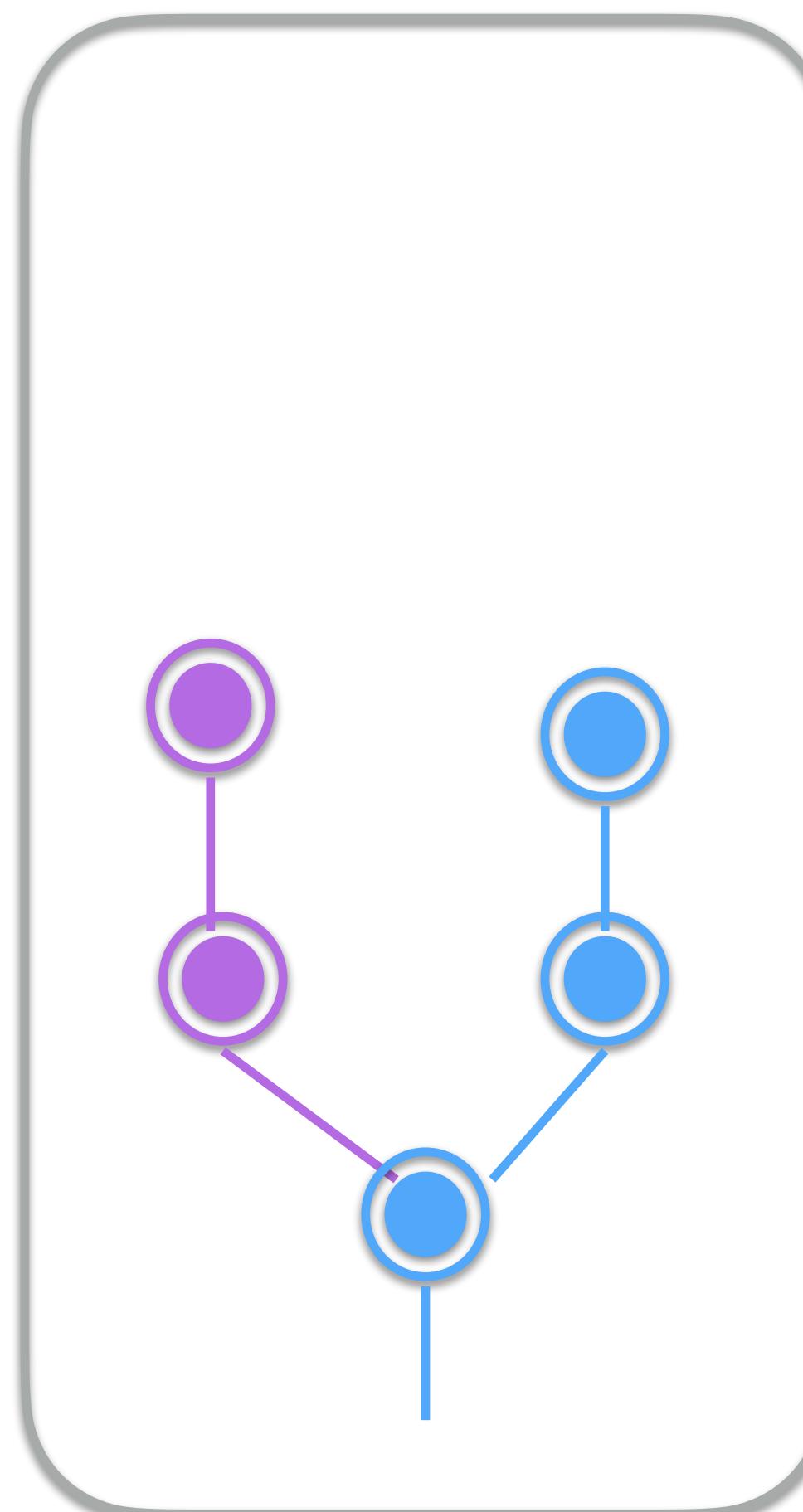
A branch is a version of the repository



✓ GRAPH ⚏ Auto ⌂ ⌃ ⌄ ⌅ ⌆ ⌇ ⌈ ⌉

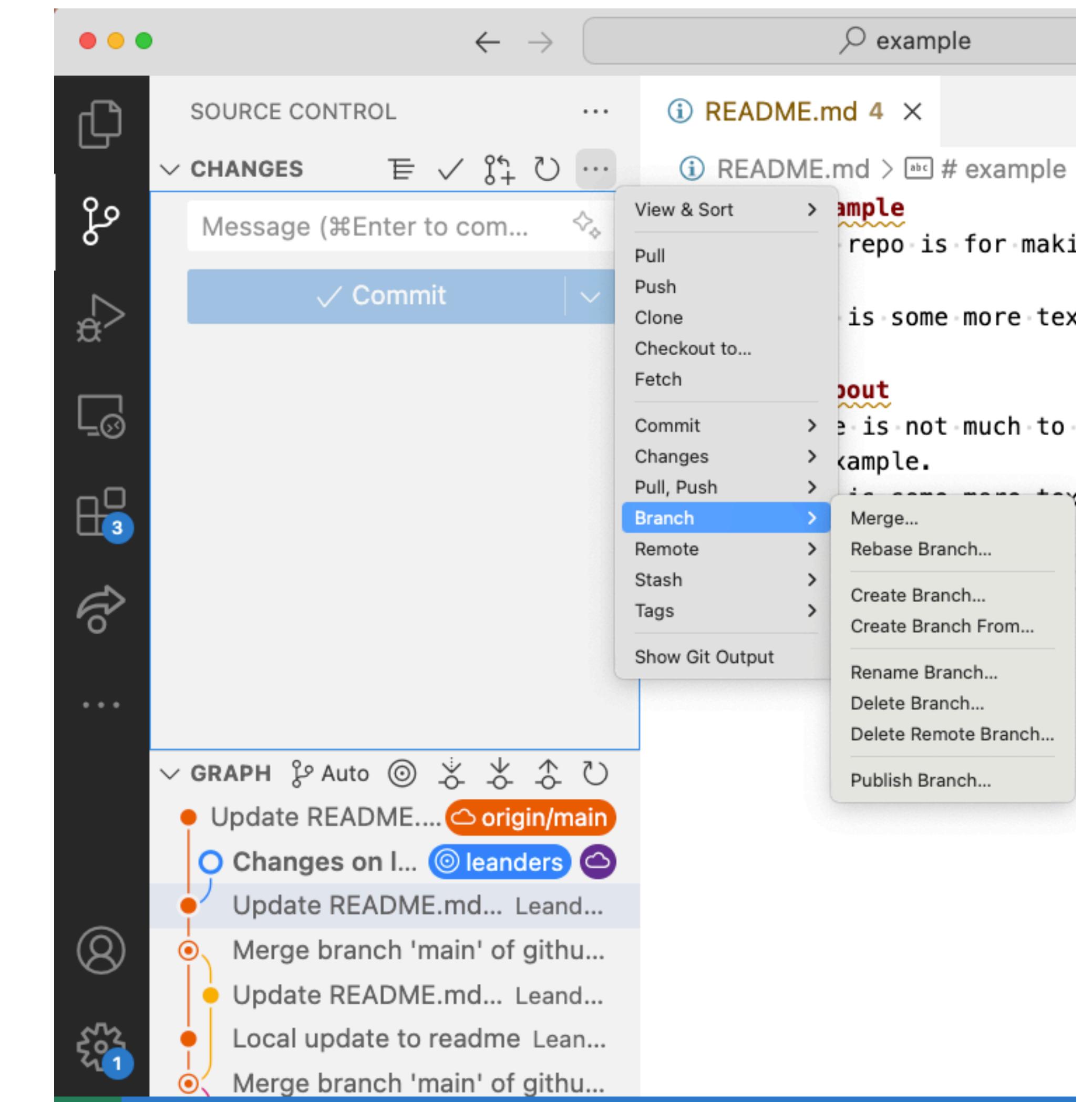
- Update README.... origin/main
- Changes on I... leanders
- Update README.md... Leand...
- Merge branch 'main' of githu...
- Update README.md... Leand...
- Local update to readme Lean...
- Merge branch 'main' of aithu...

A branch is a version of the repository

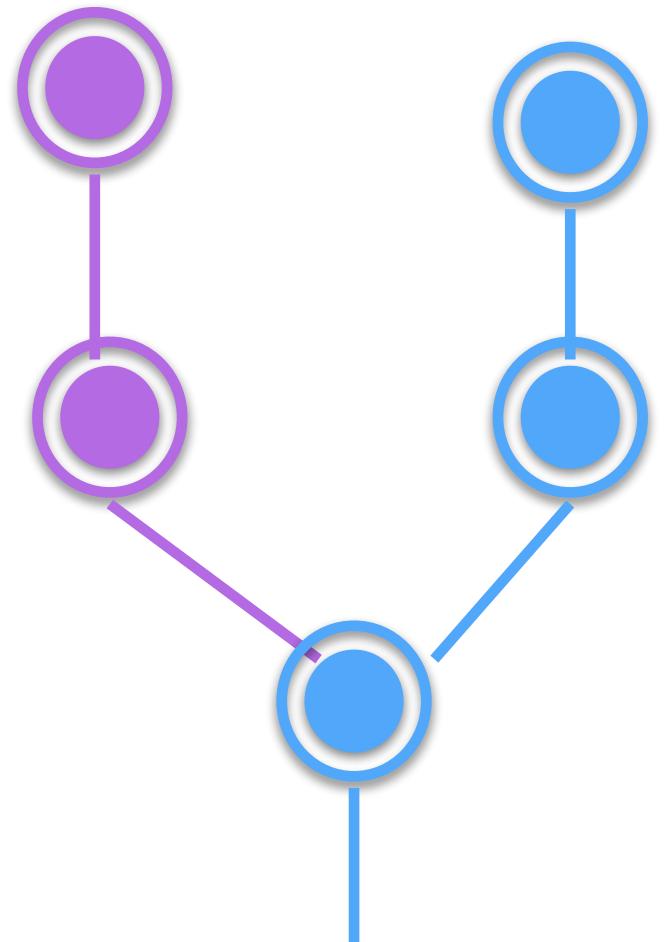


VSCode

- ...
- Branch
- Merge



A branch is a version of the repository



- Idea:
- Every developer (or feature) has one branch
- Working features are merged into main branch
- Main branch is merged into other branches
- Always merge first main into feature branch,
 - Resolve possible conflicts
 - then merge feature branch into main