

GitHub Introduction Labs

Fundamentals of collaboration

Revision 1.0 – 01/27/24

Tech Skills Transformations LLC / Brent Laster

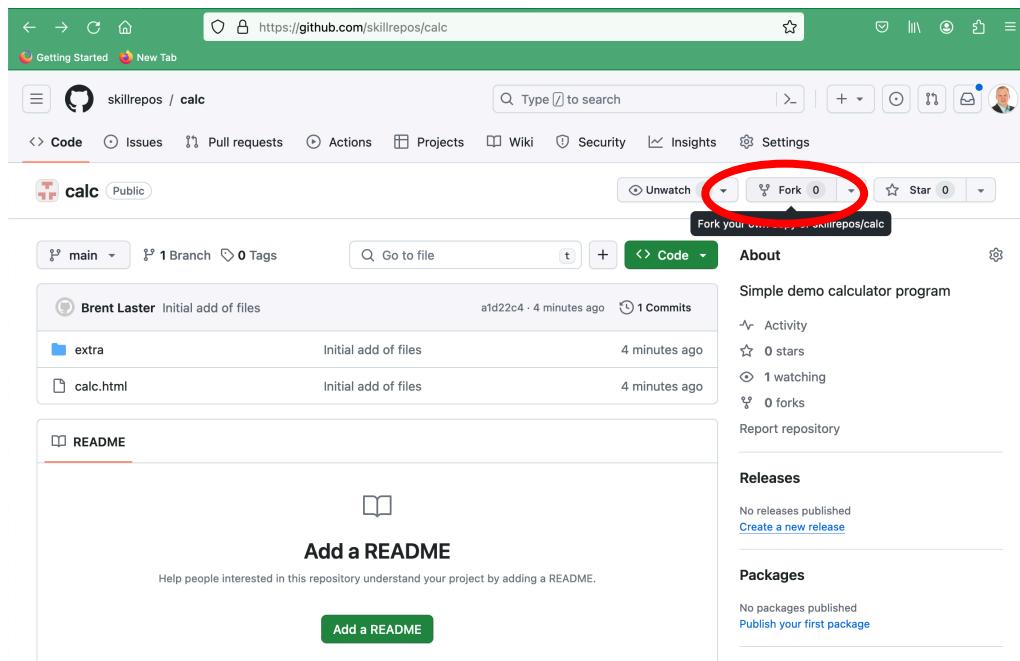
Setup and prerequisites

1. In order to do some of the labs in this class, you will need to have a personal access token (PAT) setup and also two separate GitHub userids, as well as a version of Git installed.
2. Git can be installed by going to <https://git-scm.org> and following the instructions there for your OS.
3. To create the second GitHub userid, just select another email address and sign up for the free tier at GitHub.com.
4. You can set up the PAT in advance by following the instructions [here](#) or do it as part of the first lab.
5. If you are doing the labs on Windows, it is recommended to use the Git Bash shell that can be installed with Git for Windows.

Lab 1 – Getting Started

Purpose: In this lab, we'll get a quick start learning about GitHub through forking a project, creating a new file and committing it.

1. Log in to GitHub with your primary GitHub account.
2. Go to <https://github.com/skillrepos/calc> and fork that project into your own GitHub space. Do this by clicking on the **Fork** button. On the next screen, **make sure to uncheck** the box next to **Copy the main branch only**. Then click the **Create Fork** button.



The screenshot shows the GitHub interface for creating a new fork of a repository named 'calc'. The 'Code' tab is selected. The 'Owner' dropdown is set to 'brentlaster' and the 'Repository name' field contains 'calc'. A yellow callout bubble points to the 'Owner' dropdown with the text 'uncheck'. A red circle highlights the checkbox labeled 'Copy the main branch only'. A green circle highlights the 'Create fork' button.

Create a new fork

A *fork* is a copy of a repository. Forking a repository allows you to freely experiment with changes without affecting the original project.

Required fields are marked with an asterisk (*).

Owner * Repository name *

brentlaster / calc

Description (optional)

Simple demo calculator program

Copy the main branch only
Contribute back to skillrepos/calc by adding your own branch. [Learn more.](#)

You are creating a fork in your personal account.

Create fork

- Now you'll be on your fork of the repo. Next, let's clone your repo down to your local system so we can make changes there. In your project, ensure you are on the **Code** tab, then click on the large green **<> Code** button. In the **Local** tab, select **HTTPS** under Clone and then click on the **copy icon** to copy your project's URL.

The screenshot shows the GitHub interface for the forked repository 'brentlaster/calc'. The 'Code' tab is selected. The repository details show it was forked from 'skillrepos/calc'. A red circle highlights the 'Code' tab. A red circle highlights the 'Local' tab in the clone options. A red circle highlights the 'HTTPS' link. A red circle highlights the 'Copy url to clipboard' button. A red circle highlights the 'Copy' icon in the clone options.

Code

Local

HTTPS

Copy url to clipboard

Copy

4. Open a terminal on your system and clone down the repository from GitHub. You can use the following command – just paste (or type) the URL you copied from the step above and then change to that directory. Then change into the local working directory.

```
$ git clone <url from repo>
```

```
$ cd calc
```

5. If not already set globally, configure your name and email. Best practice would be for your email to be the same as the one you're using for your userid on GitHub.

```
$ git config user.name "your name"
```

```
$ git config user.email <same email as you're using on GitHub>
```

6. After this you can run the command below and see that GitHub is setup as your remote repository.

```
$ git remote -v
```

7. Let's make a simple edit to a file so we can have a change to push back to GitHub. Edit the calc.html file and update the line in the file surrounded by <title> and </title> to customize it with your name. The process is described below.

Edit calc.html and change

<title>Calc</title>i

to

<title> **name's** Calc</title>

substituting in your GitHub user ID for “github_user_id”.

8. Save your changes and commit them back into the repository.

```
$ git commit -am "Updating title"
```

9. Several aspects of using GitHub rely on options you can set in the user **Settings** menu. To demonstrate this and in preparation for the next lab, we'll go to settings to create your Personal Access Token (PAT) that you'll need for securely pushing changes over to GitHub in place of a password.

To create your PAT, follow the instructions for creating a classic token at <https://docs.github.com/en/authentication/keeping-your-account-and-data-secure/managing-your-personal-access-tokens#creating-a-personal-access-token-classic>

(Alternatively, you can go directly to <https://github.com/settings/tokens/new>)

When setting up your token, ensure that you have the boxes checked for the first four scopes (*repo – delete:packages*) as shown below. **Also make sure to copy and save the token for future use.**

New personal access token (classic)

Personal access tokens (classic) function like ordinary OAuth access tokens. They can be used to interact with Git over HTTPS, or can be used to [authenticate to the API over Basic Authentication](#).

Note

intro class

What's this token for?

Expiration *

30 days The token will expire on Mon, Feb 26 2024

Select scopes

Scopes define the access for personal tokens. [Read more about OAuth scopes](#).

<input checked="" type="checkbox"/> repo	Full control of private repositories
<input checked="" type="checkbox"/> repo:status	Access commit status
<input checked="" type="checkbox"/> repo_deployment	Access deployment status
<input checked="" type="checkbox"/> public_repo	Access public repositories
<input checked="" type="checkbox"/> repo:invite	Access repository invitations
<input checked="" type="checkbox"/> security_events	Read and write security events
<input type="checkbox"/> workflow	Update GitHub Action workflows

When done, click on the green **Generate Token** button.

<input type="checkbox"/> read:ssh_signing_key	Read public user SSH signing keys
Generate token Cancel	

Make sure to save a copy of the token string from this screen - you won't be able to see it again.

The screenshot shows the GitHub 'Personal access tokens (classic)' page. On the left, there's a sidebar with 'GitHub Apps', 'OAuth Apps', and 'Personal access tokens' (which is expanded, showing 'Fine-grained tokens' and 'Tokens (classic)'). A 'Copied!' button is visible above a list of tokens, with a tooltip indicating the token has been copied to the clipboard.

9. Now, let's go ahead and push your change back into GitHub. We'll push to a new branch in preparation for the next lab. Go back to your terminal and enter:

```
$ git push -u origin main:dev
```

10. After this, you'll be prompted for username (your GitHub username) and then a sign-in/Private Access Token or password. Wherever it asks for a token or a password, you can just copy and paste in **the token you generated in GitHub prior to this lab**. An example dialog that may come up is shown below.



If instead, you are on the command line and prompted for a password, just paste the token in at the prompt. Note that it will not show up on the line, but you can just hit enter afterwards.

A terminal window showing a git push command. The command 'git push -u origin main:dev' is entered, followed by a password prompt. A context menu is open over the password field, with the 'Paste' option highlighted in blue.

```
developer@Bs-MacBook-Pro calc % vi calc.html
developer@Bs-MacBook-Pro calc % git commit -am "Updating title"
[main d9e79db] Updating title
 1 file changed, 2 insertions(+), 2 deletions(-)
developer@Bs-MacBook-Pro calc % git push -u origin main:dev
Username for 'https://github.com': brentlaster
Password for 'https://brentlaster@github.com':
```

NOTE: If you hit run into problems trying to push with the token, such as it saying invalid password, you may be getting caught by previously saved credentials. See the very end of this doc for some other options.

END OF LAB

Lab 2 – Pull requests

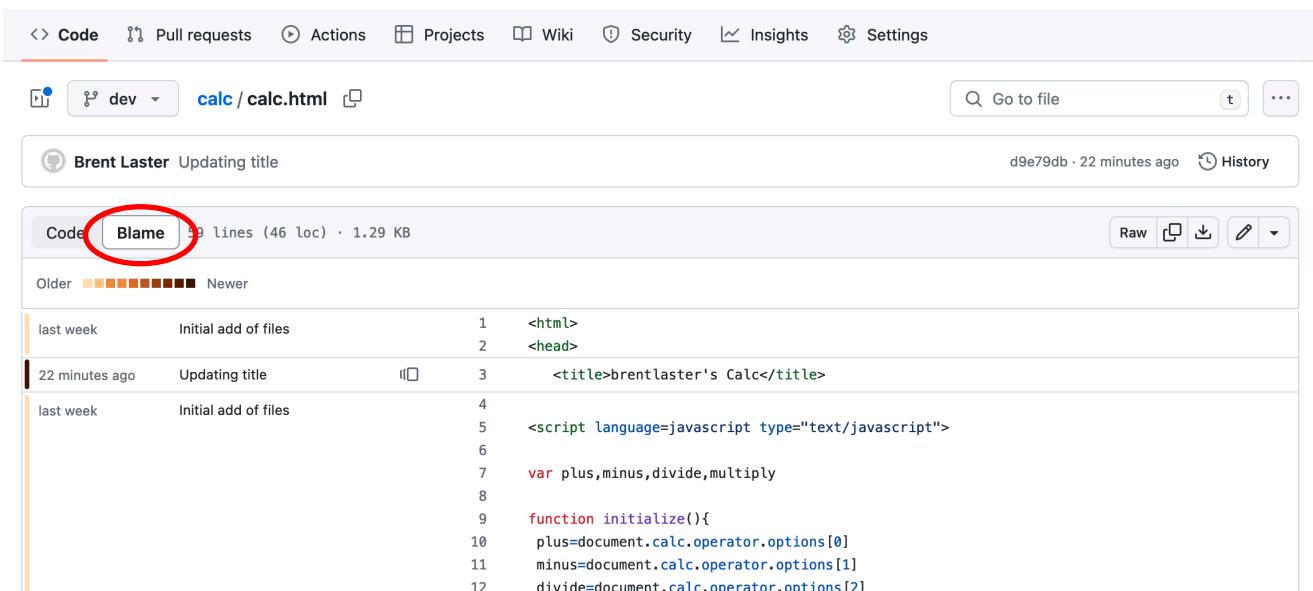
Purpose: In this lab, we'll see how to merge a change using a pull request.

- After the push is complete, you can switch back to the GitHub repo in the browser, change the branch to **dev** and click on the calc.html file to see the change. (If you don't see **dev** listed in the branch dropdown list, click on the **3 Branches** button next to the dropdown and you should be able to see it there. Alternatively, you can go to github.com/<github userid>/calc/tree/dev in the browser.)

The screenshot shows a GitHub repository interface. At the top, there are navigation buttons: 'dev' (selected), '3 Branches' (which shows 3 branches), and '0 Tags'. On the right, there are buttons for 'Go to file', 'Add file', and 'Code'. Below these are two main sections: a 'Switch branches/tags' modal on the left and the repository history on the right. The modal has a search bar 'Find or create a branch...' and tabs for 'Branches' (selected) and 'Tags'. It lists branches: 'main' (default), 'cspace', and '✓ dev'. Below the modal is a link 'View all branches'. The repository history shows a single commit: 'Updating title' by '61e42da · 8 hours ago' with '3 Commits'. There is also a 'Contribute' and 'Sync fork' button.

The screenshot shows the same GitHub repository interface, but now the 'dev' branch is selected. The top navigation buttons are 'dev' (selected), '3 Branches', and '0 Tags'. The repository history shows a commit from 'Brent Laster' updating the title. The file list at the bottom shows 'calc.html' (selected) and 'README'. A message at the top right says 'This branch is 1 commit ahead of skillrepos/calc:main.' There are 'Contribute' and 'Sync fork' buttons.

- Click on the file name to open the file in the browser. While you have the file open there, click on the **Blame** button in the gray bar at the top to see additional information about who made changes to the content.



Code **Blame** 59 lines (46 loc) · 1.29 KB

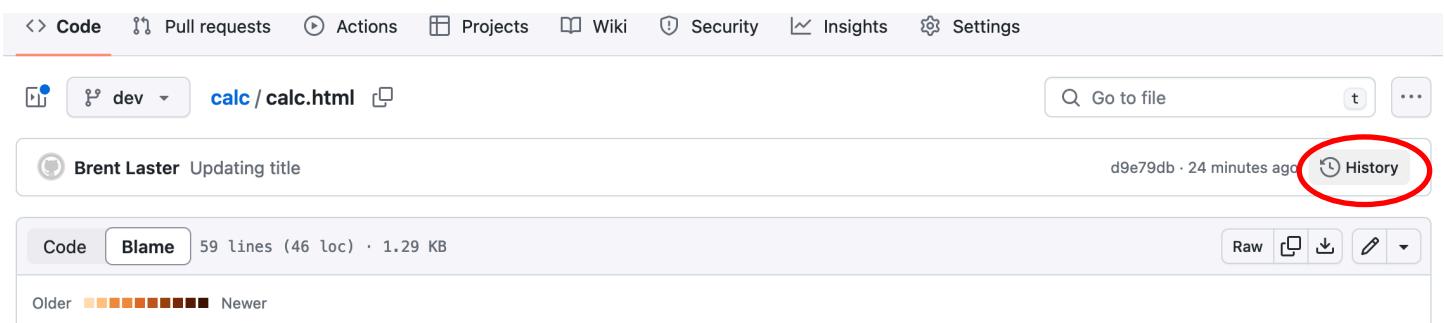
Older  Newer

```

last week   Initial add of files      1  <html>
2           <head>
22 minutes ago  Updating title    3  <title>brentlaster's Calc</title>
last week   Initial add of files      4
5           <script language=javascript type="text/javascript">
6
7           var plus,minus,divide,multiply
8
9           function initialize(){
10          plus=document.calc.operator.options[0]
11          minus=document.calc.operator.options[1]
12          divide=document.calc.operator.options[2]

```

3. Also, click on the *History* button (upper right) to see the change history for the file.

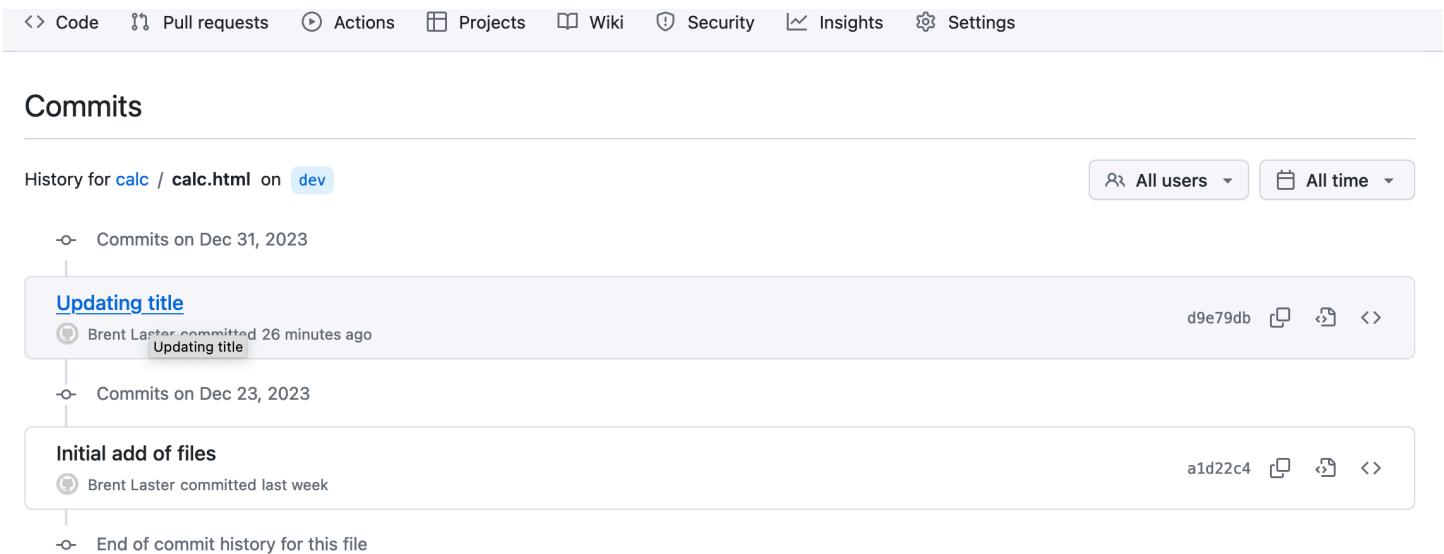


Code **Blame** 59 lines (46 loc) · 1.29 KB

Older  Newer

Brent Laster Updating title d9e79db · 24 minutes ago **History**

4. In the history screen, click on the commit message for your change. You'll then be able to see the differences introduced by your commit.



Commits

History for **calc** / **calc.html** on **dev** **All users** **All time**

- o- Commits on Dec 31, 2023
 - Updating title** **Brent Laster committed 26 minutes ago** **Updating title** **d9e79db**
- o- Commits on Dec 23, 2023
 - Initial add of files** **Brent Laster committed last week** **a1d22c4**
- o- End of commit history for this file

Updating title

Brent Laster committed 28 minutes ago

1 parent a1d22c4 commit d9e79db

Showing 1 changed file with 2 additions and 2 deletions.

Whitespace Ignore whitespace Split Unified

```

 4 calc.html
@@ -1,6 +1,6 @@
 1 1 <html>
 2 2 <head>
 3 - <title>Calc</title>
 3 + <title>brentlaster's Calc</title>
 4 4
 5 5 <script language=javascript type="text/javascript">
 6 6
@@ -56,4 +56,4 @@
 56 56
 57 57
 58 58 </body>
 59 - </html>
 59 + </html>

```

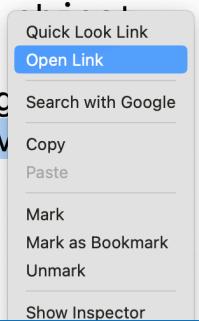
5. Let's now merge our change from the dev branch to main via a pull request. **Switch back to the terminal where you did the commit and push.**

In the output from the push, you should see a link (*highlighted in the screenshot below*). Right click and open that link. (Alternatively, you can go back to the main page of your repo and if you see a message there that looks like the second picture below, you can just click on the *Compare & pull request* button.)

```

Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 322 bytes | 322.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local
remote:
remote: Create a pull request for 'dev' on GitHub by visiting
remote:     https://github.com/brentlaster/calc/pull/new/dev
remote:
To https://github.com/brentlaster/calc.git
 * [new branch]      main -> dev
branch 'main' set up to track 'origin/dev'.

```



-- OR --

brentlaster / calc

Type ⌘ to search

Code Pull requests Actions Projects Wiki Security Insights Settings

calc Public

forked from [skillrepos/calc](#)

Pin Watch 0

dev had recent pushes 26 minutes ago

Compare & pull request

6. Depending on which option you chose in the step above, you may either be on a *Comparing Changes* screen or *Open a pull request* screen. In either case, we need to update the base repository in the gray bar at the top to make the merge go to your repo and **NOT to skillrepos/calc**. Click on the dropdown (small downward pointing arrow) and select **your repo** from the list.

Comparing changes

Choose two branches to see what's changed or to start a new pull request. If you need to, you can also [compare across forks](#) or [learn more about diff comparisons](#).

base repository: skillrepos/calc ▾ base: main ▾ ... head repository: brentlaster/calc ▾ compare: dev ▾

Choose a Base Repository

Filter repos

skillrepos/calc

brentlaster/calc

Reviewers

Assignees

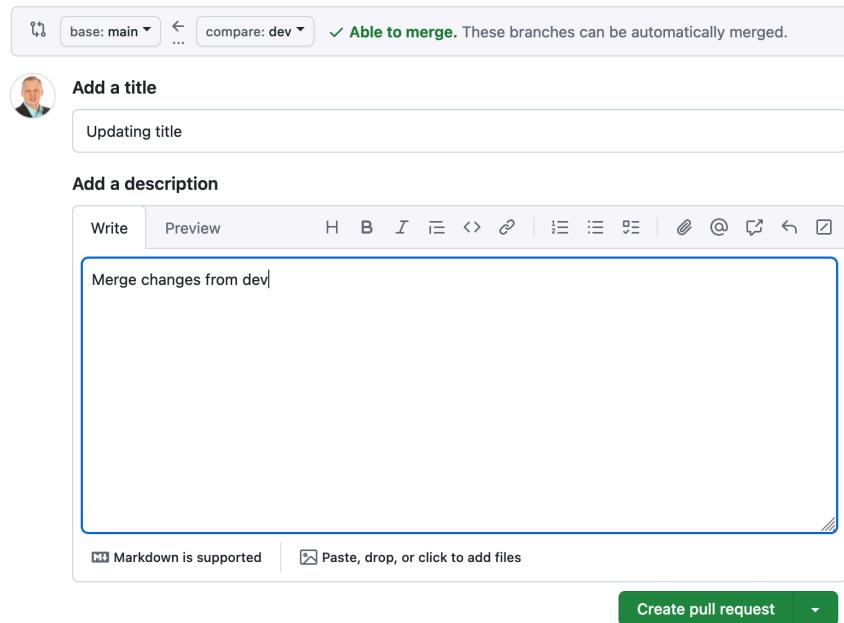
Write Preview

7. After making that change, the gray bar showing the base and compare should look like the screenshot below.

Choose two branches to see what's changed or to start a new pull request. If you need to, you can also [compare across forks](#)

base: main ▾ compare: dev ▾ ✓ Able to merge. These branches can be automatically merged.

8. Now, with your repo selected for the base, add an optional description if you want and then click on the **Create pull request** button.



9. At this point, you have created a new pull request. (Note that the *Pull Requests* tab at the top shows 1 pull request in the repo.) It will check for any conflicts for merging.

We haven't set up any CI processes or reviewers so there is nothing for those sections. Note the check in the middle section that says *This branch has no conflicts with the base branch*. You can look at the *Commits* or *File Changed* tabs if you want to see more details on the changes.

10. When you're ready, switch back to the ***Conversations*** tab. Then click on the ***Merge pull request*** button and then the ***Confirm merge*** button to complete the pull request. After that, the pull request will be completed and closed (shown in second screenshot). Afterwards, you can click on the button to delete the ***dev*** branch if you want.

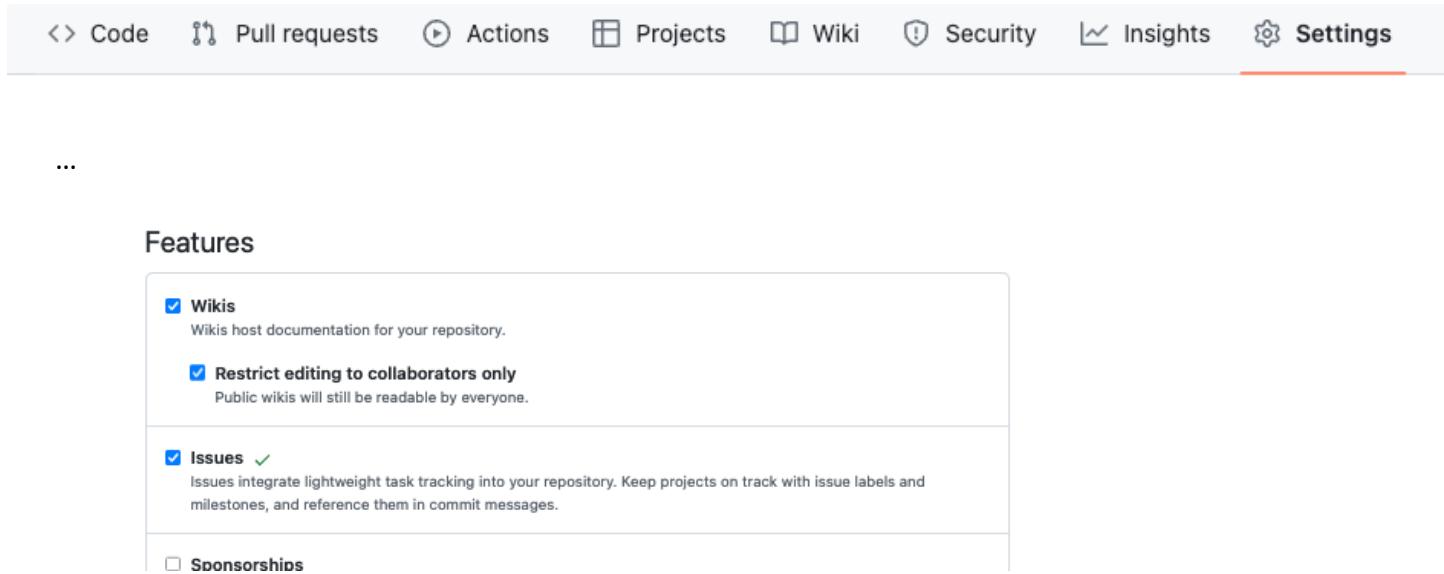
The screenshot shows a GitHub pull request interface. At the top, there's a green button labeled "Open" with a "Updating title #1" message. Below it, a message says "brentlaster wants to merge 1 commit into `main` from `dev`". A commit history shows a single commit titled "Updating title" with hash `d9e79db`. The commit message includes the instruction "Add more commits by pushing to the `dev` branch on [brentlaster/calc](#)". A user profile picture of a man is shown next to a comment bubble containing "Merge pull request #1 from brentlaster/dev". Another comment bubble below it says "Updating title". A note states "This commit will be authored by bclaster@nclasters.org". At the bottom of the modal are "Confirm merge" and "Cancel" buttons. Outside the modal, there's a "Merged" button, a "Revert" button, and a "Delete branch" button. A success message at the bottom says "Pull request successfully merged and closed" and "You're all set—the `dev` branch can be safely deleted."

END OF LAB

Lab 3: Creating GitHub issues

Purpose: In this lab, you'll create an issue, assign it to a user, and add labels for it.

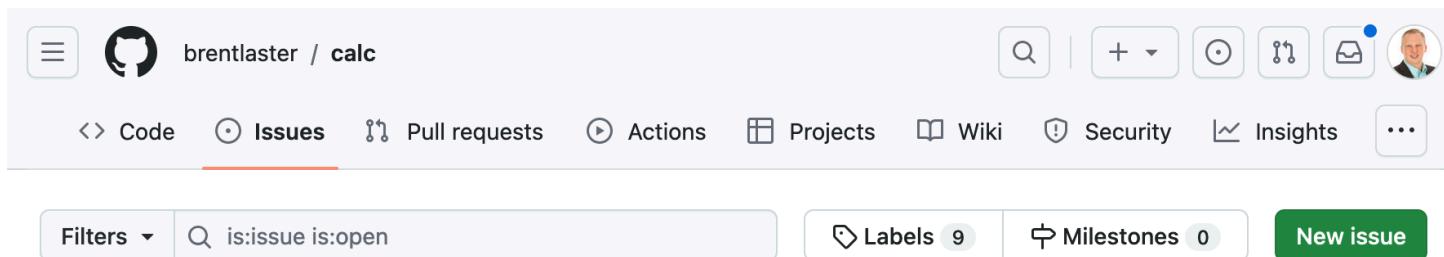
1. We'd like to have a *README* file in our project to make it more standard. So, let's create an issue to document that. First, ensure that the repository has the *Issues* feature turned on. On the main repo page, go to the repository's **Settings** tab, and then scroll down until you see the **Features** section. Then, check the box for **Issues**.



The screenshot shows the GitHub repository settings page. At the top, there are tabs for Code, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. The Settings tab is selected. Below the tabs, there is a section titled "Features". Inside this section, there are several checkboxes:

- Wikis: Wikis host documentation for your repository.
- Restrict editing to collaborators only: Public wikis will still be readable by everyone.
- Issues: Issues integrate lightweight task tracking into your repository. Keep projects on track with issue labels and milestones, and reference them in commit messages. This checkbox is highlighted with a green checkmark.
- Sponsorships

2. Now, click on the **Issues** tab at the top of the repository page, then the **New issue** button on the right. Then fill in the title with something like “Needs README”. For the description, you can enter something like “Please add a README file :book:”. (:book: will be changed to an emoji.) Then click the **Submit new issue** button.



The screenshot shows the GitHub repository issues page. At the top, there are tabs for Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and ... The Issues tab is selected. Below the tabs, there is a search bar, a filter dropdown set to "Filters", a search input with the query "is:issue is:open", a labels counter (9), a milestones counter (0), and a green "New issue" button. The Issues tab is underlined with a red line.

Add a title

Needs README

Add a description

Please add README file :book:

Assignees: No one—assign yourself

Labels: None yet

Projects: None yet

Milestone: No milestone

Development: Shows branches and pull requests linked to this issue.

Helpful resources: GitHub Community Guidelines

Submit new issue

3. Take note of what number is assigned to the issue – you will need it later. (It will probably be #2 for you)

Needs README #2

Open brentlaster opened this issue now · 0 comments

brentlaster commented now

Please add README file :book:

Add a comment

Write Preview H B I E <> ⌂ ...

Add your comment here...

4. Assign the issue to yourself by clicking on the **Assign yourself** link under the **Assignees** section on the right.

The screenshot shows a user interface with a top navigation bar containing 'Wiki', 'Security', 'Insights', and a three-dot menu. Below this is a toolbar with 'Edit' and a green 'New issue' button. The main area features a light blue sidebar with a three-dot menu icon. To its right, the word 'Assignees' is centered above a gear icon. A blue box highlights the text 'No one—[assign yourself](#)'.

5. Add the documentation label to the issue by clicking on **Labels** and selecting the **Documentation** one.

The screenshot shows a user interface with a sidebar on the left containing sections for 'Assignees' (with 'brentlaster' listed) and 'Labels'. A modal window titled 'Apply labels to this issue' is open, showing a 'Filter labels' input field and a list of labels. The 'documentation' label is selected, indicated by a checked radio button and a blue circle. Other labels listed are 'duplicate' (unchecked).

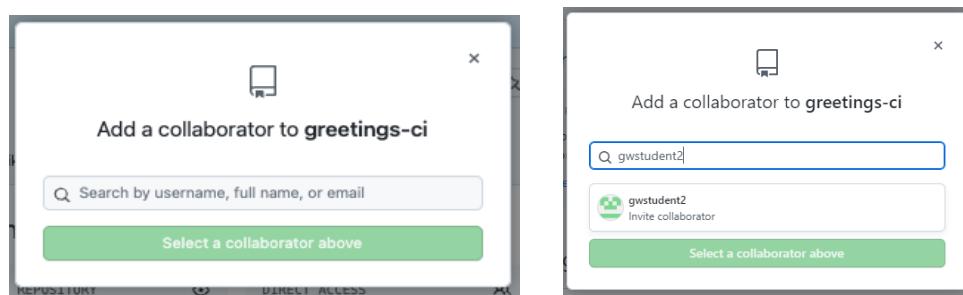
6. After this, if you click on the **Issues** tab at the top, and look at your issue, it should look like the following.

The screenshot shows a browser window for 'github.com/brentlaster/calc/issues'. The top navigation bar includes 'Code', 'Issues 1', 'Pull requests', 'Actions', 'Projects', 'Wiki', 'Security', and 'Insights'. The 'Issues' tab is active. Below the navigation is a search bar and filter options. The main content area shows a single open issue. The issue title is 'Needs README documentation' with a blue label. The issue description is '#2 opened 15 hours ago by brentlaster'. The issue has a small profile picture next to it.

7. In preparation for the next lab, we need to add your second GitHub userid as a *collaborator* to this repository. Go to the repository's **Settings** tab and then select **Collaborators** on the left under **Access**. Then click the **Add people** button.

The screenshot shows the GitHub repository settings page. The top navigation bar includes links for Code, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. The 'Settings' tab is active. On the left, a sidebar titled 'General' lists various access options: Collaborators (selected), Moderation options, Code and automation (Branches, Tags, Actions, Webhooks, Environments, Pages), Security (Code security and analysis, Deploy keys, Secrets), and others like Help support and Direct access. The main content area is titled 'Who has access'. It shows that the repository is a 'PUBLIC REPOSITORY' and that there are '0 collaborators have access to this repository. Only you can contribute to this repository.' A 'Manage' link is provided. Below this is the 'Manage access' section, which displays a message 'You haven't invited any collaborator' and a large 'Add people' button.

8. In the dialog box that pops up, enter the other GitHub userid you have and then click on the specific id or click on **Select a collaborator above**. Then, click on **Add <userid> to this repository**. That userid should then receive an email with the invite which you can accept.



9. **Make sure to respond to the email and accept the invitation!** (You will need to sign in as the invited id in a different browser or a private tab or sign out/sign in, and then view and accept the invitation.). If you sign in as the secondary id and go to <https://github.com/<primary github userid>/calc> you can also view the invitation via clicking on the button.

The screenshot shows a GitHub repository page for 'brentlaster / calc'. At the top, there's a search bar with the placeholder 'Type ⌘ to search'. Below the search bar are navigation links: Code (selected), Issues (1), Pull requests, Actions, Projects, Security, and Insights.

The repository details show the owner's profile picture, the repository name 'calc' (Public), and a note that it was forked from 'skillrepos/calc'. To the right, there's a 'Watch' button with a count of 0.

A prominent message box at the top states '@brentlaster has invited you to collaborate on this repository' with a 'View invitation' button, which is circled in red.

Below the message box is another navigation bar with links: Code, Issues (1), Pull requests, Actions, and Projects.

The main content area shows two user profile pictures: the invitee (@brentlaster) and the inviter (SkillRepos). A message reads 'brentlaster invited you to collaborate'.

Below the message are two buttons: 'Accept invitation' (circled in red) and 'Decline'.

Information about what the invitee can see is listed under a lock icon:

- Owners of calc will be able to see:
 - Your public profile information
 - Certain activity within this repository
 - Country of request origin
 - Your access level for this repository
 - Your IP address

Below this, a question asks 'Is this user sending spam or malicious content?' with a 'Block brentlaster' link.

The screenshot shows the same GitHub repository page after accepting the invitation. The navigation bar at the top is identical.

The main content area now displays a message: 'You now have push access to the brentlaster/calc repository.' with a close button (X).

END OF LAB

Lab 4: Setting up a pull request with reviewers

Purpose: In this lab, you'll use a pull request with a reviewer and an associated issue to make a change.

- Now, we'll address adding the README itself per the issue we previously created. If you're not signed in as your original/primary GitHub userid, sign in as that id now. In the **Code** tab of the *calc* repository, click on the green button to add a README.md file.

The screenshot shows the GitHub interface for the 'calc' repository. The top navigation bar includes tabs for Code, Issues (1), Pull requests, Actions, Projects, Wiki, and Settings. The repository card shows it's public and forked from 'skillrepos/calc'. Below the card, a message indicates the branch is 2 commits ahead of 'skillrepos/calc:main'. A list of recent commits by 'brentlaster' is shown, including 'Merge pull request #1 from br...' (yesterday), 'Initial add of files' (last week), and 'Updating title' (yesterday). The main content area is titled 'README' and contains a large 'Add a README' button, which is highlighted with a red oval.

- This will bring up the editor in GitHub. Enter the text below in the new file text input area for README.md. Fill in your github userid in both places instead of github-userid. (Notes: Do this on a single line. Also, there is no space between the "]" and "("). And since we don't have a calculator emoji, we're using an abacus emoji. Finally, if you cut and paste from this doc, that may add an image link at the end of the line that has to be removed.)

This is a simple calculator :abacus: program. :question: can be directed to [@github-userid](<https://github.com/github-userid>)

Code Issues Pull requests Actions Projects Wiki Security Insights

calc / README.md in main

Cancel changes Commit changes...

Edit Preview Spaces 2 Soft wrap

```
1 This is a simple calculator :abacus: program. :question: can be directed to @brentlaster
2
```

3. Click on the Preview tab (next to Edit) to see how this will render once committed.

Code Issues Pull requests Actions Projects Wiki Security Insights

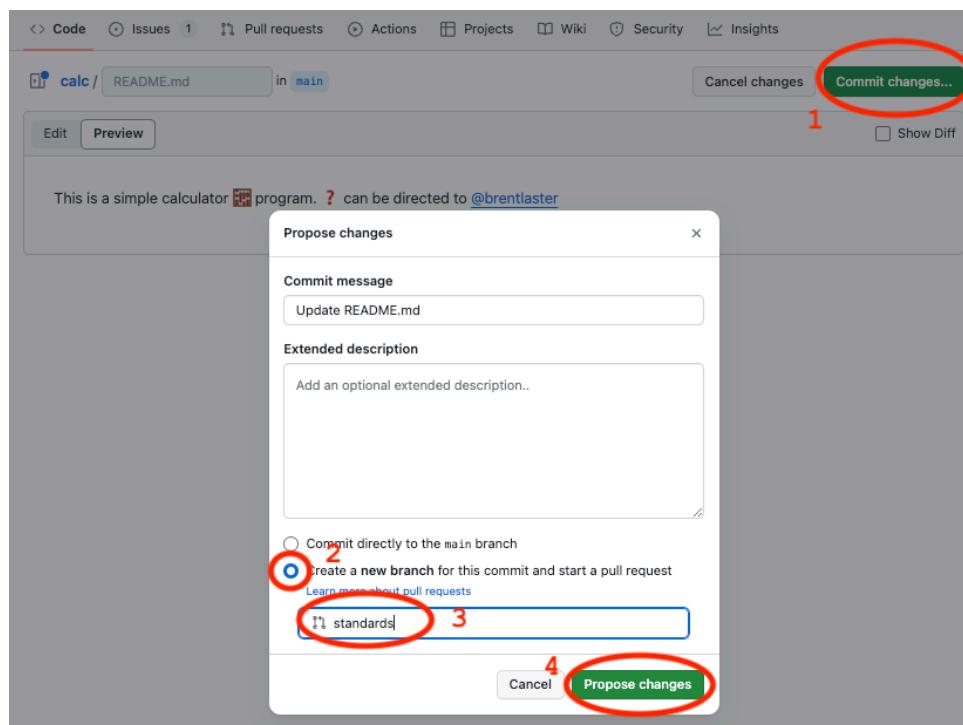
calc / README.md in main

Cancel changes Commit changes...

Edit Preview Show Diff

```
This is a simple calculator :abacus: program. ? can be directed to @brentlaster
```

4. Now let's commit these changes to a new branch and open a pull request to merge them. click on the green **Commit changes...** button in the upper right corner. In the dialog, enter a comment if you want and select the option to **Create a new branch...**. You can change the generated branch name if you want. In this case, I've changed it to "standards". Then click **Propose changes**.



5. At this point, you'll see a screen showing you the changes and what's being compared at the top. This should only be branches in the same repo, not different repos. It should also show a green checkmark with "Able to merge." next to it. We're going to create a pull request to be reviewed. Click on the **Create pull request** button.

The screenshot shows the GitHub 'Comparing changes' interface. At the top, there are dropdown menus for 'base: main' and 'compare: standards'. A green checkmark indicates 'Able to merge. These branches can be automatically merged.' Below this, a message encourages discussing changes with others, followed by a prominent green 'Create pull request' button, which is circled in red. The main content area shows a commit history from 'Commits on Jan 1, 2024', featuring a single commit to 'README.md' by 'brentlaster'. The commit message includes a link to a GitHub profile. A detailed view of the README.md file shows one addition and no deletions, with the added text: 'This is a simple calculator :abacus: program. :question: can be directed to [@brentlaster] (<https://github.com/brentlaster>)'.

6. You'll now be on the screen to create the pull request. Let's add your secondary GitHub id as a reviewer. In the upper right, click on the **Reviewers** link, then select your other id from the list. (You can just make sure it's checked and hit ESC or type it into the field.) Make sure your other userid shows up in the Reviewers section now.

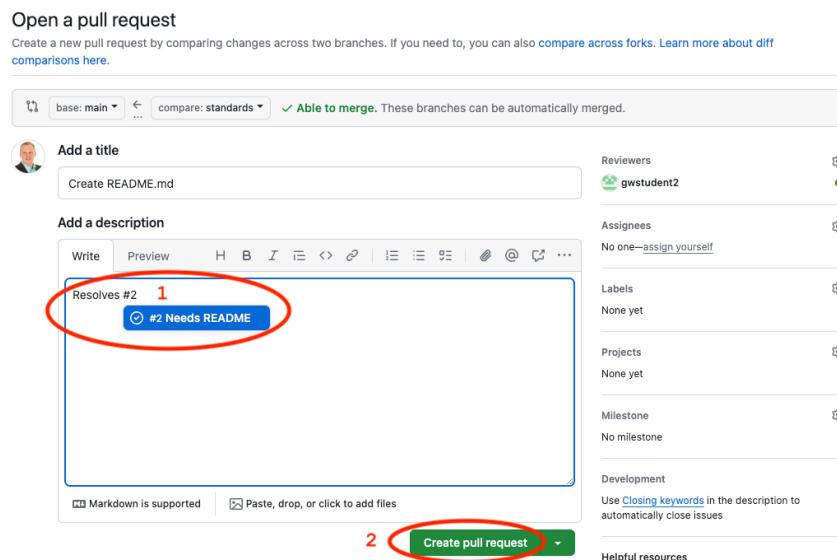
The screenshot shows the 'Open a pull request' page. The top bar includes dropdowns for 'base: main' and 'compare: standards', and a green 'Able to merge' status. On the left, there are fields for 'Add a title' (with a placeholder 'Create README.md') and 'Add a description' (with a text area containing 'Add your description here...'). On the right, the 'Reviewers' section is highlighted. It shows a list with one user selected: 'gwstudent2', which is circled in red with the number '1'. Below this, other fields include 'Labels' (None yet), 'Projects' (None yet), 'Milestone' (No milestone), and 'Development' (with a note about closing keywords). At the bottom, there is a 'Create pull request' button.

7. Also, we can add in a description that will automatically close the associated issue when we resolve this pull request. Click in the “Add your description here...” field and enter

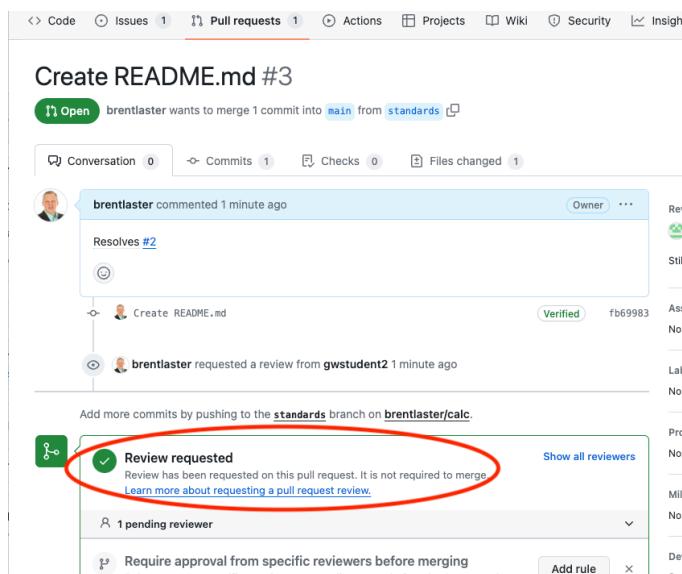
Resolves #2

If you have a different issue number, change the 2 to your issue number.

Then click on the “Create pull request” button.



8. Afterwards, you'll be on the screen for the open pull request. Around the middle of the screen, you can see the conditions that need to be satisfied before the pull request can be merged. This includes the pending review you have from your secondary GitHub user id.



Lab 5: Completing a pull request with reviewers

Purpose: In this lab, we'll complete the pull request we started in the last lab.

1. In a separate browser or a private tab, log in to your secondary GitHub userid (the one you added as a collaborator and a reviewer). After you log in, you can either go to your notifications to see the item about the requested review or go to <https://github.com/pulls/review-requested>. Then click on the commit message for the pull request.

The screenshot shows the GitHub Notifications interface. At the top, there's a header with a search bar, a plus sign button, and a mail icon circled in red with the number '1'. Below the header, a blue bar indicates 'Inbox' with '1' notification. The main area lists notifications under categories: Saved (0), Done (✓), Filters (Assigned, Participating, Mentioned, Team mentioned, Review requested), and a Repository section (brentlaster/calc, 1 notification). A prominent notification for 'Create README.md' from 'brentlaster/calc #4' is highlighted with a red circle and the number '2'. This notification includes a 'Select all' checkbox, a 'Dismiss' button, and a 'Get started' button. Below the notification, a 'ProTip!' suggests creating custom filters. At the bottom right, there are 'Prev' and 'Next' buttons.

- OR -

The screenshot shows a browser window with the URL <https://github.com/pulls/review-requested> circled in red. The page title is 'Pull Requests'. The search bar contains the query 'is:open is:pr review-requested:gwstudent2 archive'. Below the search bar, it says '1 Open' and '1 Closed'. A single pull request is listed: 'brentlaster/calc Create README.md'. This item is also circled in red. A note below the list says '#4 opened 14 minutes ago by brenlaster'. At the bottom, a 'ProTip!' suggests adding 'no:assignee' to see everything that's not assigned.

2. This will open up the pull request. There is a button at the top to “Add your review”. Click on that.

The screenshot shows a GitHub pull request page for a repository named 'brentlaster / calc'. The pull request is titled 'Create README.md #4' and is currently 'Open'. A message at the top states 'brentlaster requested your review on this pull request.' A green button labeled 'Add your review' is circled in red. Below the message, the pull request summary indicates 'brentlaster wants to merge 1 commit into main from standards'. The commit count is 1, and there are 0 checks and 1 file changed. The commit author is 'brentlaster' and it was made 18 minutes ago. There is no description provided. The review section shows 'Reviewers' with 'gwstudent2' listed.

3. We could click on any of the lines and add a comment if we wanted, but since this is simply adding a README file, it looks ok. However, since this is about standards, let's make a suggestion to also add a license for the repo. Select the **Review changes** button and add a comment to that effect. Then select the **Approve** option, and then **Submit review**.

The screenshot shows the same GitHub pull request page as before, but now with a review dialog open. The dialog title is 'Finish your review'. In the text area, the comment 'Suggest adding a license file too.' is entered and circled in red. A red number '1' is placed above the 'Review changes' button, which is also circled in red. A red number '2' is placed below the text area. In the review options section, the 'Approve' radio button is selected and circled in red. A red number '3' is placed next to it. A red number '4' is placed above the 'Submit review' button, which is also circled in red. The right side of the screen shows the standard GitHub sidebar with 'Viewed' and other repository details.

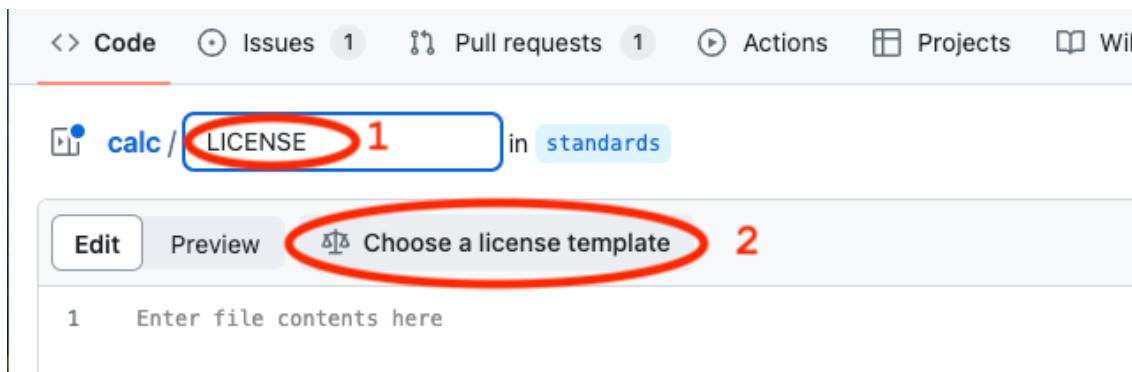
4. Go to the session with your original GitHub userid or log out of the other one and log back in if you need to. Go to the **Pull requests** menu at the top, find the pull request and click on the commit message. Then you should see a screen like below.

The screenshot shows a GitHub pull request page for a repository named 'calc'. The pull request is titled 'Create README.md #4' and is currently open. The commit message is 'Create README.md'. A comment from 'gwstudent2' suggests adding a license file. The pull request has 1 commit and 1 file changed.

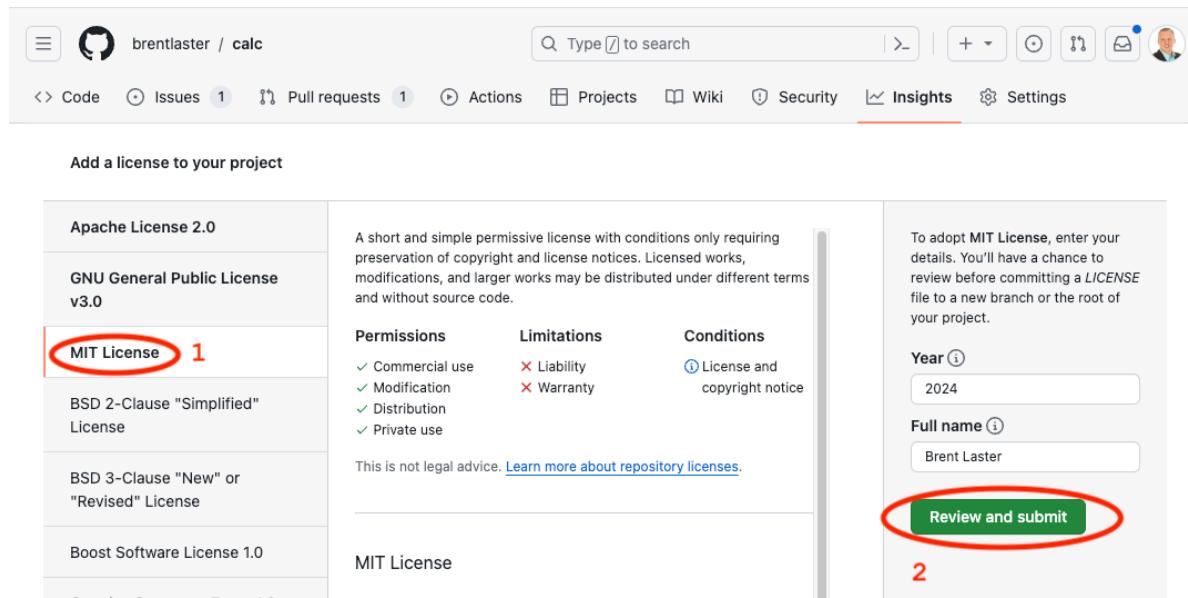
5. Since there was a suggestion to add a license file, that sounds like a good idea, so let's do that. Click on the **Code** tab at the top, then select the **standards** branch from the branch dropdown, then select the "+" sign and the option to **+ Create new file**.

The screenshot shows the GitHub repository 'calc'. The 'Code' tab is selected (step 1). The 'standards' branch is selected in the dropdown (step 2). A context menu is open over a commit, with the '+ Create new file' option highlighted (step 3). Step 4 indicates the option to upload files.

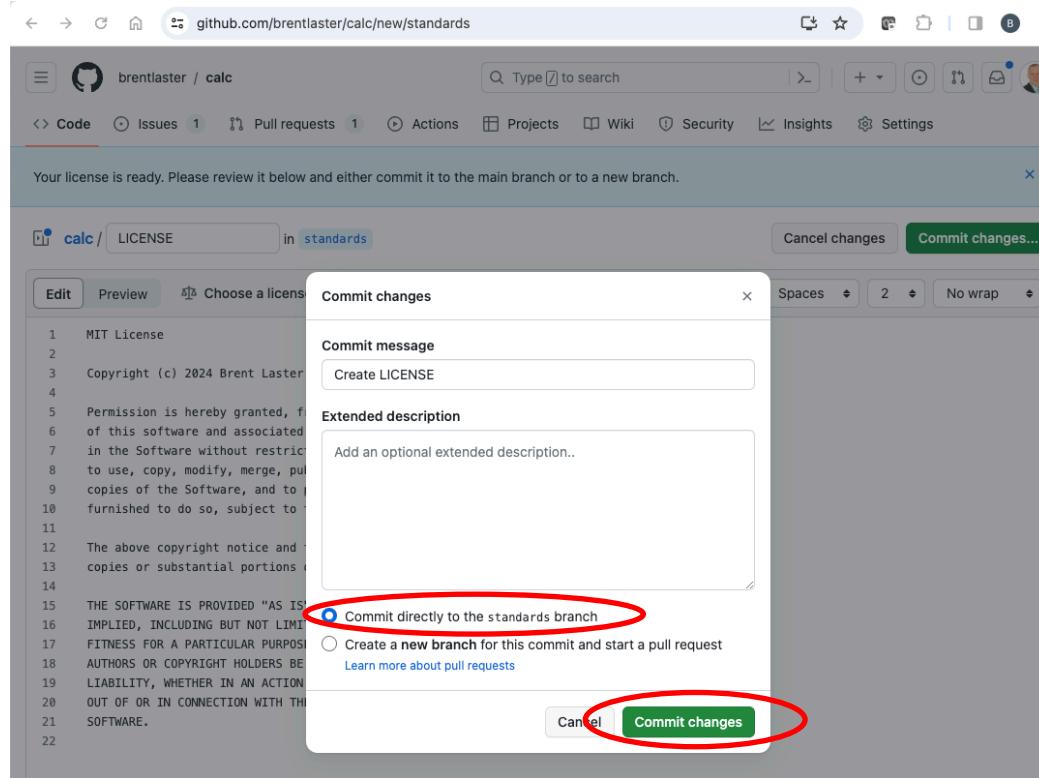
6. In the next screen, there will be a text entry area for the name of the file. Type in “LICENSE” for the name. Then, an option will display that says ***Choose a license template***. Click on that option. You will be asked about discarding changes. It’s ok in this case, so click on “OK”.



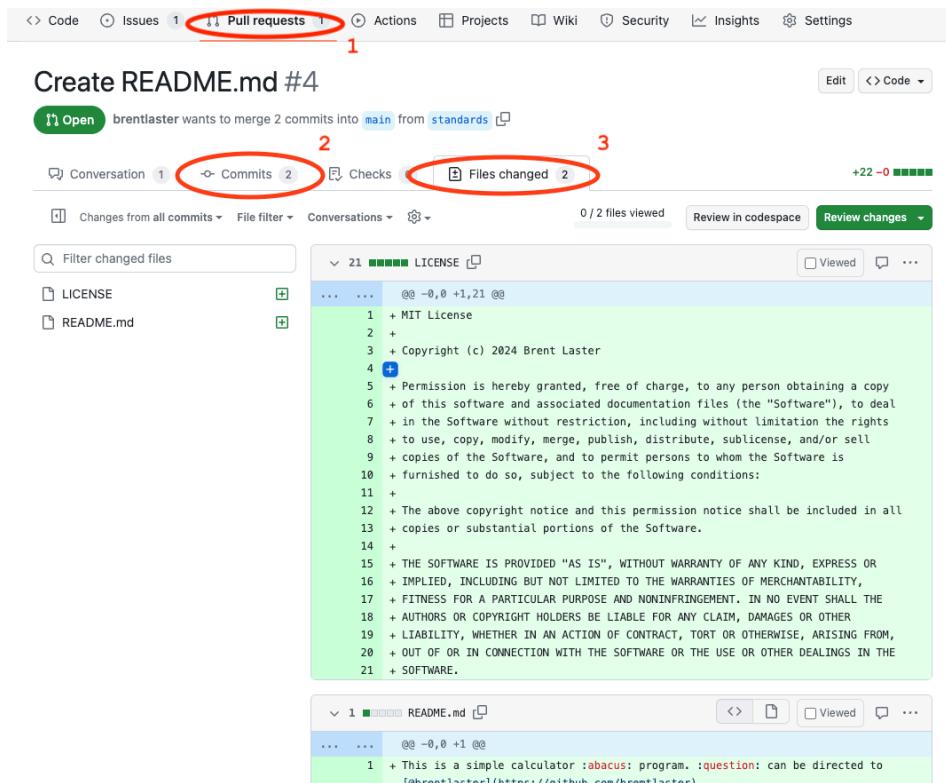
7. On the next screen, you’ll be able to pick the license you want. You can select the “MIT License” or another one if you prefer. Once done, click the ***Review and submit*** button on the right.



You’ll have an opportunity to review the license. When ready, just click on the ***Commit Changes*** buttons to commit the file to the *standards* branch. Be sure to leave it on the *standards* branch so it will be added to the existing pull request.



8. Go back to the pull request by selecting **Pull requests** at the top and selecting the one open pull request. You can look at the changes currently in the pull request by clicking on the **Commits** tab and also the **Files changed** tab.



9. Click back on the **Conversation** tab in the pull request and go ahead and **merge and close** (confirm merge) the pull request. After completing the merge, you should be able to click on the **Issues** tab and see that your issue has been automatically closed. You can click on the **Closed** list and then open the issue to see the automatically generated log of comments and actions if you want.

END OF LAB

That's all - THANKS!

Other options for making changes in repo vs https (if the https approach doesn't work for you) – choose one of A,B, or C if and only if the https push did not seem to work...

A. Resetting credential helpers: Especially on Windows, if you are pasting in your token for the password, but still getting an error message referencing password authentication, you may be running into issues because you have previous credentials stored in the *credential helper*.

One of the things you can try in this case is resetting the stored credentials via:

```
$ git config --global credential.helper store
```

Then you do your push as per the lab. It will probably pop up a text entry box for you to add your username in and another to paste in your password (PAT) and then will replace your credentials with those and complete the push.

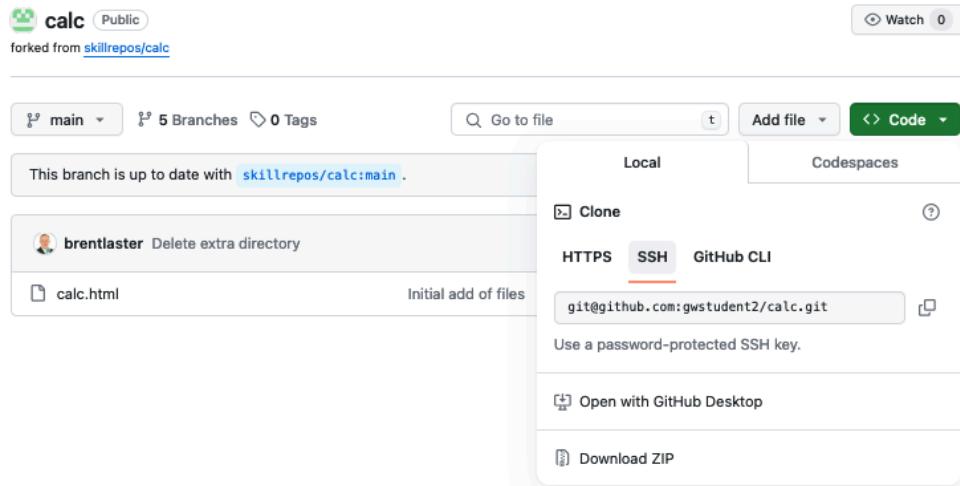
(Note: If you prefer to disable the global credentials helper entirely, you can try

```
$ git config --unset --system credentials.helper
```

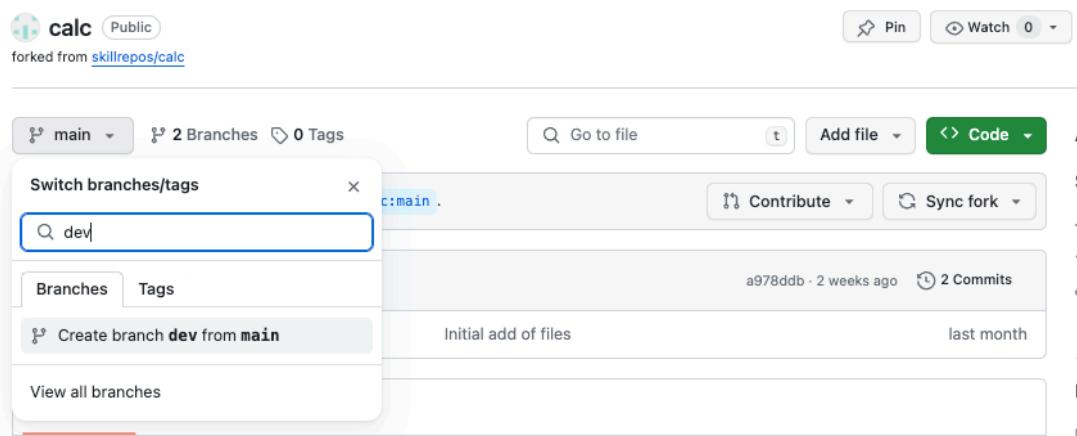
This may or may not work depending on if you have access to do this.)

B. SSH keys: If you are familiar with using ssh and have keys, you can add them into GitHub and use those. Ref <https://docs.github.com/en/authentication/connecting-to-github-with-ssh/adding-a-new-ssh-key-to-your-github-account> for more details.

If you go this route, when you get the remote URL from the browser, select the SSH tab.



C. Commit directly in GitHub: Another option is to commit directly to GitHub in the browser. To do this, first create a *dev* branch in the repo. Click on the branch dropdown under the title of the repo. In the *Find or create a branch* field, type **dev**. Then click on **Create branch dev from main**.



In the *dev* branch, click on the *calc.html* file and open it up.

This branch is up to date with [skillrepos/calc:main](#).

brentlaster Delete extra directory · a978ddb · 2 weeks ago · 2 Commits

[calc.html](#) Initial add of files · last month

[README](#)

Click on the pencil icon to edit the file.

```

calc / calc.html
Brent Laster Initial add of files · a1d22c4 · last month · History
Code Blame 59 lines (46 loc) · 1.27 KB · Code 55% faster with GitHub Copilot
1 <html>
2 <head>
3 <title>Calc</title>

```

Make the changes noted in Lab 1 in the file.

When done editing, click on the **Commit changes...** button in the upper left, then in the dialog that comes up, you can leave all the options as they are, and then click on the **Commit changes** button to commit/push the file.

A screenshot of a web browser displaying a GitHub repository page for 'gwstudent / calc'. The URL in the address bar is 'github.com/gwstudent/calc/edit/dev/calc.html'. The page shows the file 'calc.html' in the 'dev' branch. A modal dialog box titled 'Commit changes' is open over the code editor. The 'Commit message' field contains the text 'Update calc.html'. Below it, the 'Extended description' field has the placeholder 'Add an optional extended description..'. At the bottom of the dialog, there are two radio button options: 'Commit directly to the dev branch' (selected) and 'Create a new branch for this commit and start a pull request', with a link 'Learn more about pull requests'. There are 'Cancel' and 'Commit changes' buttons at the bottom right of the dialog.

```
1 <html>
2 <head>
3     <title>Brent's Calc</title>
4
5 <script language="javascr
6
7 var plus,minus,divide,mul
8
9 function initialize(){
10     plus=document.calc.operat
11     minus=document.calc.operat
12     divide=document.calc.operat
13     multiply=document.calc.operat
14 }
15
16 function calculate(){
17     a = parseInt(document.calc.an
18     b = parseInt(document.calc.an
19     if (plus.selected)
20         document.calc.answer.value= a+
21     if (minus.selected)
22         document.calc.answer.value= a-
23     if (divide.selected)
24         document.calc.answer.value= a/b
25     if (multiply.selected)
26         document.calc.answer.value= a*b
27 }
```