**GitHub Create New Branch:**

Follow these steps to create a branch:

1. Navigate to the repository folder on your computer.
2. In Terminal on macOS or Git Bash on Windows, type git pull or git pull origin main and press Enter.
3. Type git checkout -b Simple\_Leaflet\_Map and press Enter.

* git checkout lets us navigate between branches.
* -b indicates we are creating a new branch.
* The name of the new branch follows -b.

After pressing Enter, your terminal or Git Bash should return the following:

Switched to a new branch 'Simple\_Leaflet\_Map'

Now we are in the Simple\_Leaflet\_Map branch. Confirm this by typing git branch and pressing Enter. The output in Terminal or Git Bash should look as follows, with an asterisk next to the branch name:

\* Simple\_Leaflet\_Map

main

**Add, Commit, and Push to a Branch:**

* First, check to see if it’s in current branch (i.e. Simple\_Leaflet\_Map branch), then type: git branch, and hit enter. If the following output is returned with an asterisk next to the branch name, you're in the Simple\_Leaflet\_Map branch:

\*Simple\_Leaflet\_Map

main

* Then, type git status and enter, it will show the new files/folders you’re working on.
* To see all the folders and files in the Simple\_Map folder, type git status -u and press Enter.
* Then, type git add . and hit enter
* Then, type git status and hit enter
* Then, type git commit -m “add message.”
* Then, type git push. If the output return:

fatal: The current branch Simple\_Leaflet\_Map has no upstream branch.

* To push the current branch and set the remote as upstream, use:

git push –set-upstream origin Simple\_Leaflet\_Map

* To push the folders and files to the Simple\_Leaflet\_Map branch for the first time, type git push –set-upstream origin Simple\_Leaflet\_Map in the command line and press Enter.
* Check the repo in github, it will show 2 branches:

Main

Simple\_Leaflet\_Map with files/folders in it

**Compare Branch to the main Branch:**

After the code is reviewed and approved, then it’s merging the branch to the main branch. Before to do that, compare the changes between the working branch and main.

To merge your Simple\_Leaflet\_Map branch with the main branch, we will:

1. Compare the changes between the Simple\_Leaflet\_Map branch and the main branch so that we can merge them.
2. If we are able to merge the Simple\_Leaflet\_Map branch into the main branch, we need to create a pull request on GitHub.
3. After we review the pull request, we can merge the branch into the main branch.

Let's compare the changes between our Simple\_Leaflet\_Map branch and the main branch:

1. Navigate to your Mapping\_Earthquake GitHub repository.
2. To start merging your branch into the main branch, compare the branches to confirm they can be merged.
3. You have two ways to compare your branch with the main branch:

* Click the green "Compare & pull request" button, or
* Click the gray "New pull request" button.

1. If you click the "New pull request" button, a new page will open with two buttons: "base: main" and "compare: main." Compare the Simple\_Leaflet\_Map branch with the base or main branch.
2. Click on the "compare: main" button and select the branch you want to compare (the Simple\_Leaflet\_Map branch).

**Create a Pull Request:**

After “Compare & pull request” button is selected, select the branch to compare:

Base: main <- compare: Simple\_Leaflet\_Map

A new page called "Open a pull request" will launch.

The page shows the five key elements of a pull request:

1. Near the top of the page is a green checkmark and text stating "Able to merge." If you're unable to merge, GitHub will display an explanatory message.
2. Next is the commit message you made.
3. "Reviewers" and "Assignees" are members, such as yourself, with access to the repository. You can assign the pull request to yourself. Labels, Projects, and Milestones are completed by team members and owners of the repository. Click on a gear wheel to modify any of these items.
4. In the "Leave a comment" field, describe what you're adding to the main branch in the pull request:

Adding the following folders and files to create a mapping GeoJSON points Leaflet map.

- Mapping\_GeoJSON\_Points/index.html

- Mapping\_GeoJSON\_Points/static/css/style.css

- Mapping\_GeoJSON\_Points/static/js/logic.js

1. After entering a comment, click the green "Create pull request" button.

Once a reviewer approves your pull request in a comment, you can click the green Merge pull request button.

If your code looks good, click the "Merge pull request" button.

**Merge Branch Into main Branch:**

After clicking the "Merge pull request" button, the page will refresh with two options: "Confirm merge" or "Cancel."

Click the green "Confirm merge" button. The page refreshes to confirm, in three places, that the pull request has been merged into the main branch.

This page gives the option of deleting the branch. However, don't delete it so that future interns can use it to create a simple Leaflet map.

Navigate to the main branch on GitHub to confirm all the folders and files have been merged from the Simple\_Leaflet\_Map branch.

Next, pull the latest changes on the main branch on our computer because the main branch on our computer is not up to date. Follow these steps.

1. Open the terminal or Git Bash and type git checkout main and press Enter.
2. Type git status and press Enter. The output might tell you that you are up to date, which can be misleading, or it might say you are "1" commit behind the Simple\_Leaflet\_Map branch.
3. Type git pull and press Enter. The output should look like the following:
4. To confirm that the files are in the main branch on your computer, if you have a Mac type open . to view the directory in Finder