# Assignment: 7-Day Food Log

Assignment Description: Now that you understand how to use GitHub, create a repository for a weekly food journal. The repository should include 7 days (Sunday thru Saturday) worth of food logs. Each day should have their own branch as well as the weekly guide (master branch). In order to upload your food journal, use a text editor of some sort or an Excel spreadsheet.

Students will be able to:

- Create a private repository
- Clone repository to local path on their computer
- Use GitBash, GitHub.com, GitHub desktop, and/or the terminal to make changes
- Use git commands (cd Documents, git clone, git add, git commit -m"", git push, git checkout -b, git branch, git merge, etc)
- Merge all 7 days of food logs into the weekly branch (master branch)

Create a Repository
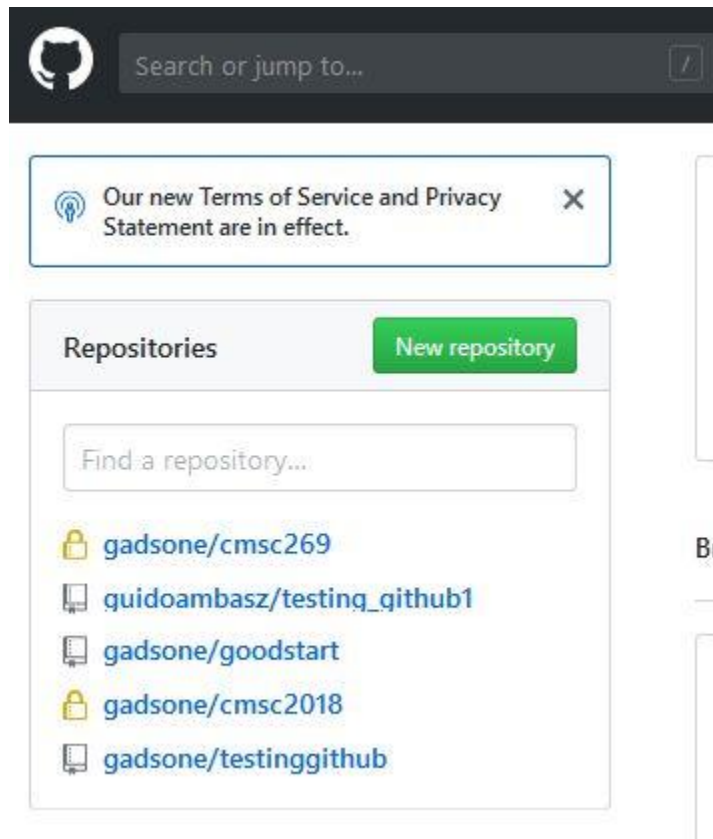
Log into GitHub.com and **select "New repository"** to create Weekly Food Journal repository.

[1]

Create a new repository

A repository contains all the files for your project, including the revision history.

Owner **Repository name**

🟪 gadsone ▾ / weekly-food-journal ✓

Great repository names are short and memorable. Need inspiration? How about **super-duper-spork**.

Description (optional)

This repository will keep track of the foods and calories consumed from August 5 to August 11.

○ 📘 **Public**
Anyone can see this repository. You choose who can commit.

◉ 🔒 **Private**
You choose who can see and commit to this repository.

☑ **Initialize this repository with a README**
This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: None ▾ | Add a license: None ▾ ⓘ

Create repository

Name your repository weekly-food-journal (or something similar). Add a description so that collaborators know what the repository's purpose is. The description will also show in README, so make sure to check the initialize repo with a README. Click ***"create repository."***

You have successfully created a private repository for your weekly journal. Next, you must clone your repository so that a local path to it will appear on your computer.

Clone Repository Using GitBash
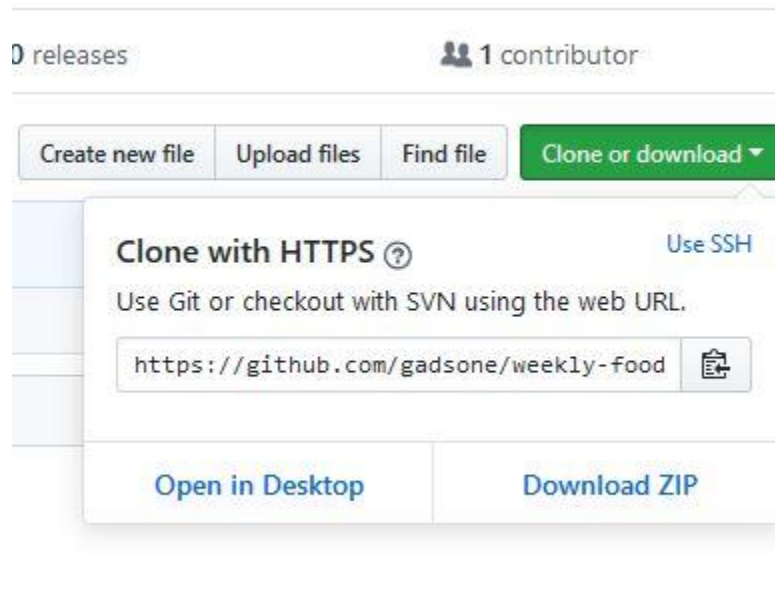
In order to create a local path on your computer, **click "Clone or download"** in your weekly-food-journal repository on GitHub.com and copy the link (see screenshot above).
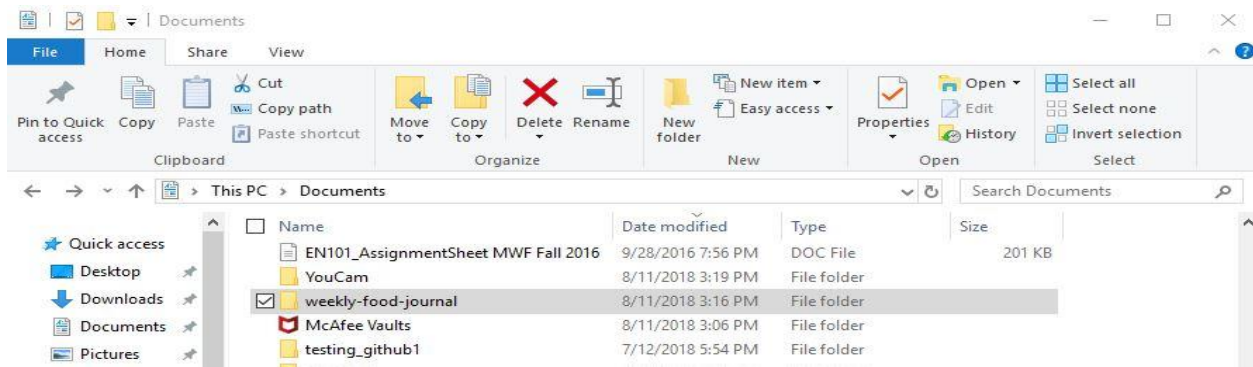
Open up GitBash and change your directory by typing in cd Documents. This will immediately place your repository locally in the documents folder. Next, type in *git clone <paste link here>* to store your local repository on your computer. You may need to type in *cd <repository name>* in order for the branch to change to master or the weekly guide.

[4]

After, locate your repository folder on your computer. If you entered in your command line **cd Documents**, it should be in your documents folder.

Add Files
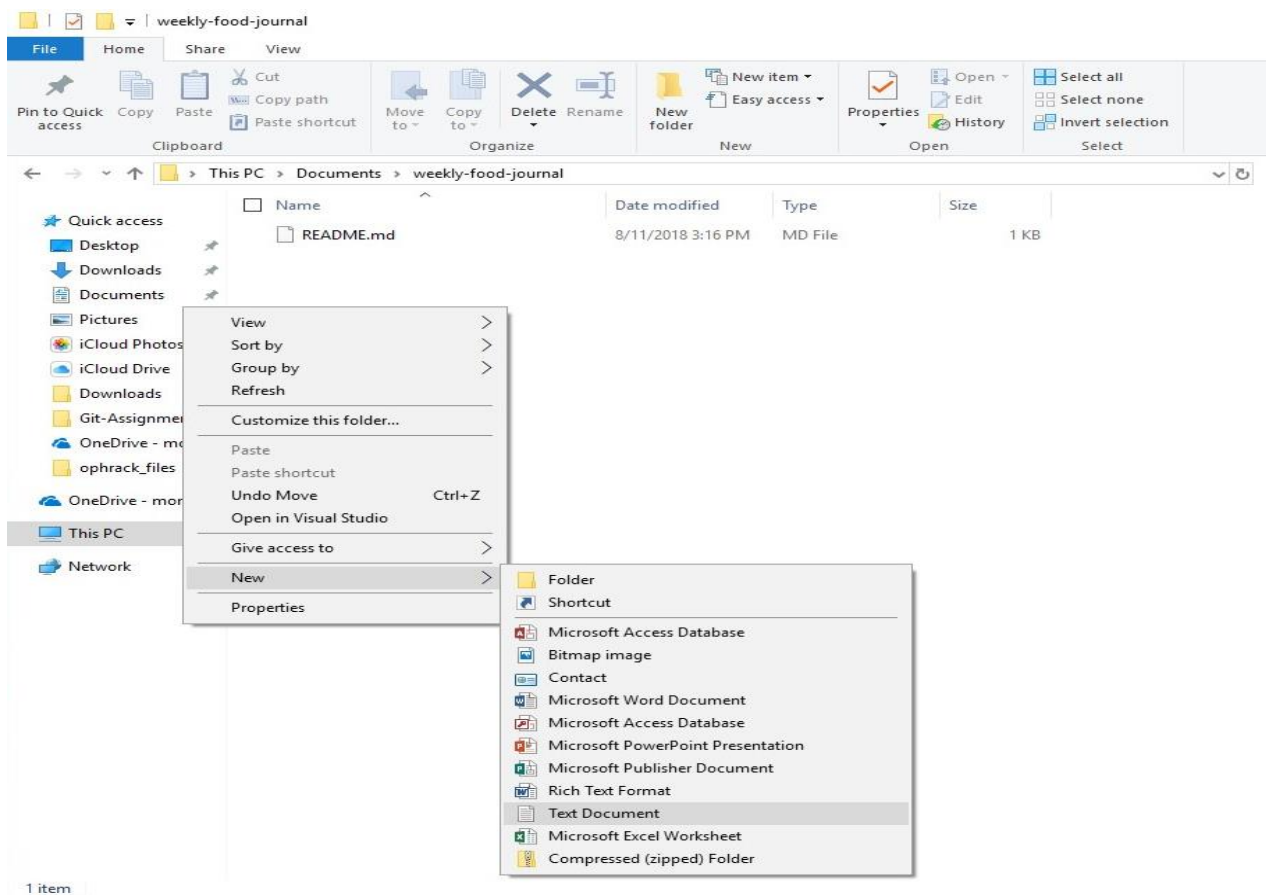
Open your repository folder and add and/or create files. In this case, you need to add a weekly guide file, so use a text editor like Notepad (Text document).

```
Guide-For-Weekly-Food-Journal - Notepad                                   —   □   ✕

File  Edit  Format  View  Help
For the next week, record daily food consumption. Include food name, igredients, measurements of ingredients, calories
count per ingredient and/or food, and any notes (optional).

Daily calorie limit: 1,400 calories (NOTE: ONLY EXCEED THIS LIMIT DURING THE WEEKENDS)

        Daily Entry Example:

    Sunday, July 29, 2018

            Breakfast:

            1 Peach --                      50 calories         Notes: none
            1/2 of Honey Cheerios w/ Skim milk   150 calories    Notes: none

            Mid-morning Snack:

            Chobani's Vanilla Yougurt       100 calories        Notes:

            Lunch:

            Tuna Salad                                          Notes: See ingredients
            1/2 cup of tuna                 170 calories        Notes: Use entire can
            1/2 cup of spinach              140 calories        Notes: none
            Dash of pepper                  0 calories          Notes: No calories for this

            Mid-Afternoon Snack:

            Assorted nuts package           100 calories        Notes: Can just be one nut source

            Dinner:

            1/2 cup of Spinach              140 calories        Notes: none
            1/2 cup of beets                180 calories        Notes:
            Tangerine                       80 calories         Notes:

            Dessert (optional):

            Banana                          70 calories         Notes:

            (NOTE: IT MAY BE HELPFUL TO CREATE AN EXCEL SPREADSHEET WHEN RECORDING WHAT FOODS TO EAT DAILY)

Create a new branch for each day of the week. Use GitBash, GitHub.com, GitHub Desktop, or the terminal to do so.
```

Create a Guide for weekly food journal. You should add what you will be recording in the journals (calories, food names, ingredients, measurements, and notes). You should also state your calorie limit if there is any. Use the example above as a guide to record your food logs daily. It may help to use an Excel spreadsheet for each day instead of a text editor. Save the file once you are done.

[6]

Then, type **git status** into the command line to see if there were any changes made locally. The status of the repository shows that Guide-For-Weekly-Journal.txt has been created but not added to our repository on GitHub.com.

Git Add and Git Commit

By typing in *git add <file name created here>* into the command line, the file will be used I git commit which save the changes preparing it to be pushed into your online repository.

After you git add, *git commit -m "enter message here"* so that changes are recorded. In this case, you created a new file and GitBash has recorded that you made this change based on a guide for daily entry.

Type in *git push* to save all of your changes to you online repository.

You should be able to see the file in your repository online once you *git push.*

Git pull allows any changes made online (GitHub.com) to be transferred locally (GItBash). This is will be extremely useful since you may want to use both GitBash and GitHub.com interchangeably. You can create pull requests, merge branches, and even commit changes online. Do not forget to transfer all of these changes made online to your local repository by typing in *git pull* into the command line.

Create branches (Method 1)
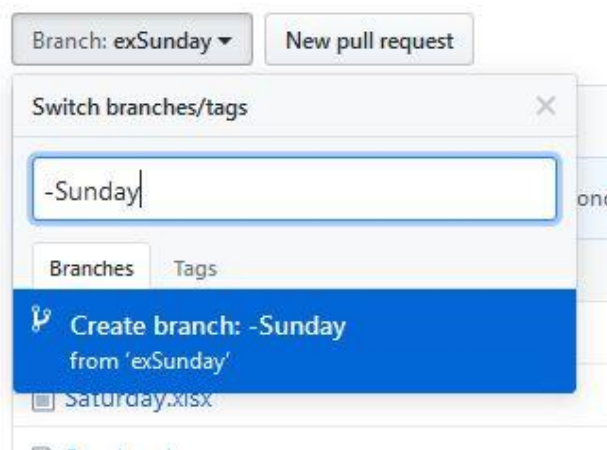
Using either GitBash or online repository, you can create multiple branches. These branches allow you to track changes over time. As mentioned before, GitHub is a version control software meaning these branches each represent a version (v1., v1.0, v1.0.2, etc.) In this case our branches each represent a day of the week. **On Github.com change and/or create a branch by clicking on the "Branch: master" button and typing in Sunday.** (Note: Type in Sunday not -Sunday – this was provided for screenshot purposes).

Since you created a branch online instead of using the command line, *git pull* all your changes into the local repository. The new branch Sunday should transfer there.

Git branch -a lists all of the branches located in the local repository and notes the branch you are currently in with an *asterisk (*).*

```
egadson@LAPTOP-O7L7K9HH MINGW64 ~/Documents/weekly-food-journal (master)
$ git checkout Sunday
Switched to branch 'Sunday'
```

In order to switch out of the master branch or any branch, type in ***git checkout <enter branch name you would like to switch to here>.***

Create branches (Method 2)

```
egadson@LAPTOP-O7L7K9HH MINGW64 ~/Documents/weekly-food-journal (Sunday)
$ git checkout -b Monday
Switched to a new branch 'Monday'
```

You can also create a branch using the command line. Type ***git checkout -b <branch name>*** which will switch over from Sunday branch to Monday branch in this case. Using this command will allow both a creation of a branch and the switch so it may be faster than using your online repository.

```
egadson@LAPTOP-O7L7K9HH MINGW64 ~/Documents/weekly-food-journal (Monday)
$ git commit -m "new branch added"
On branch Monday
nothing to commit, working tree clean
```

Type in ***git commit -m "new branch added"*** to commit your changes.

```
egadson@LAPTOP-O7L7K9HH MINGW64 ~/Documents/weekly-food-journal (Monday)
$ git push --set-upstream origin Monday
Total 0 (delta 0), reused 0 (delta 0)
To https://github.com/gadsone/weekly-food-journal.git
 * [new branch]      Monday -> Monday
Branch 'Monday' set up to track remote branch 'Monday' from 'origin'.
```

Next, type in ***git push –set-upstream origin <branch name>*** so that the changes on your online repository.

For Tuesday thru Saturday use either Method 1 or Method 2 when creating those branches.

[10]

Update README and Upload Food Log

For each branch you create, you should update the README.md file. <u>Go to your online repository and select Sunday under</u> ***"Branch: "*** Select the pin tool in the README.md box.
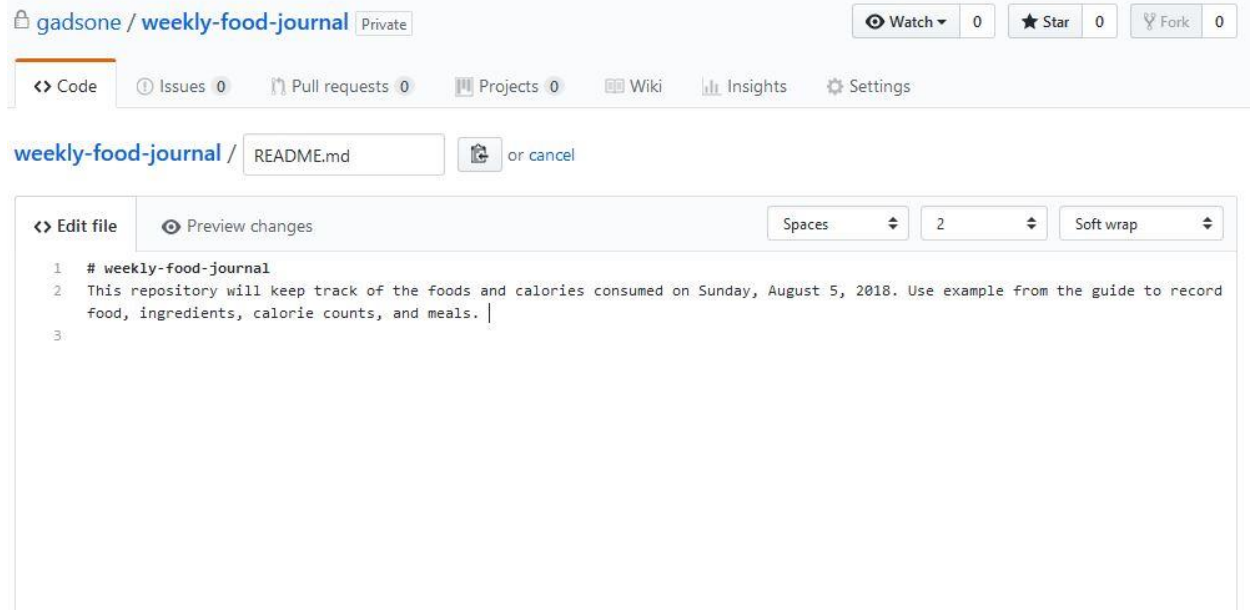
You can keep the same README file or edit it adding that the branch and repository represent different days for food journaling. Feel free to add more description into the file.
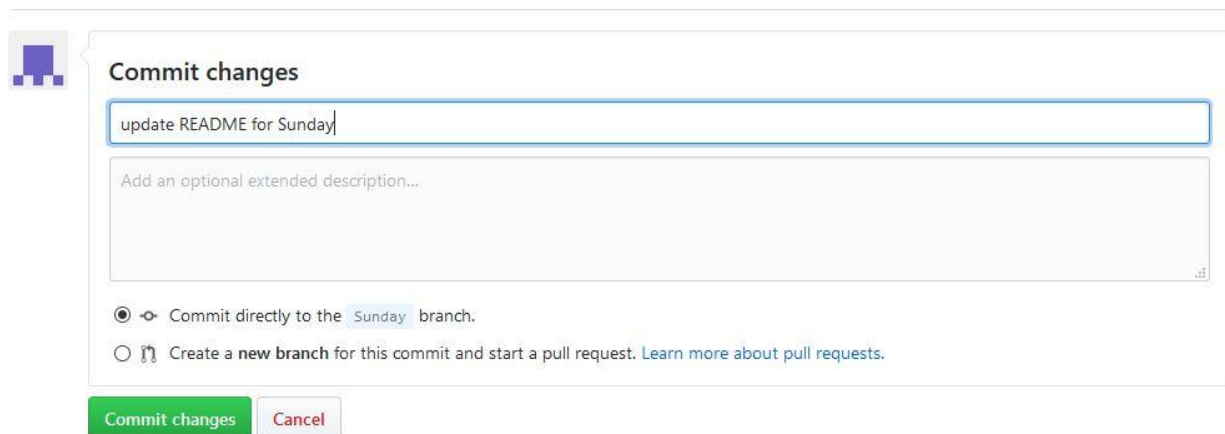
Scroll down to commit your changes to the online repository by adding a description. Try to make a habit of adding descriptions especially when you are collaborating with others on an assignment. Once you are done, **click "commit changes."**

```
egadson@LAPTOP-07L7K9HH MINGW64 ~/Documents/weekly-food-journal (Sunday)
$ git pull
remote: Counting objects: 3, done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.
From https://github.com/gadsone/weekly-food-journal
   9ab4e61..68511b3  Sunday      -> origin/Sunday
There is no tracking information for the current branch.
Please specify which branch you want to merge with.
See git-pull(1) for details.

    git pull <remote> <branch>

If you wish to set tracking information for this branch you can do so with:

    git branch --set-upstream-to=origin/<branch> Sunday
```

Do not forget to *git pull* your changes to your local repository.

Before uploading a file in your online repository, **create a text document or any other document with food journal for Sunday.** An excel spreadsheet is used above because it easier to organize meals, calories, and notes. It is also easier to calculate total calories for the day. Save the file as the day of the week.

[13]

After, **click "Upload files"** in your online repository under the Sunday branch in order to add Sunday's food log.

weekly-food-journal /

Drag additional files here to add them to your repository

Or choose your files

Sunday.xlsx   ✕

Commit changes

Sunday food log on excel

Add an optional extended description...

◉ ⦁ Commit directly to the `master` branch.

○ ⑃ Create a **new branch** for this commit and start a pull request. Learn more about pull requests.

Commit changes   Cancel

Then, **select "choose your files"** and upload Sunday's log from your computer. (Note: The log does not have to be in your local repository if you choose to upload the file online). Add a message before committing your changes. **Click "Commit changes."**

Update README files and upload food logs for Monday thru Saturday repeating these steps.

[15]

Merge Branches

```
egadson@LAPTOP-07L7K9HH MINGW64 ~/Documents/weekly-food-journal (master)
$ git merge Wednesday
Auto-merging README.md
CONFLICT (content): Merge conflict in README.md
Automatic merge failed; fix conflicts and then commit the result.
```

Once you have uploaded all of your food logs from Sunday thru Saturday for any week, it is best merge all of these branches into your master branch. If you receive an error like this, it is best create a new pull request and resolve the conflict in the online repository.
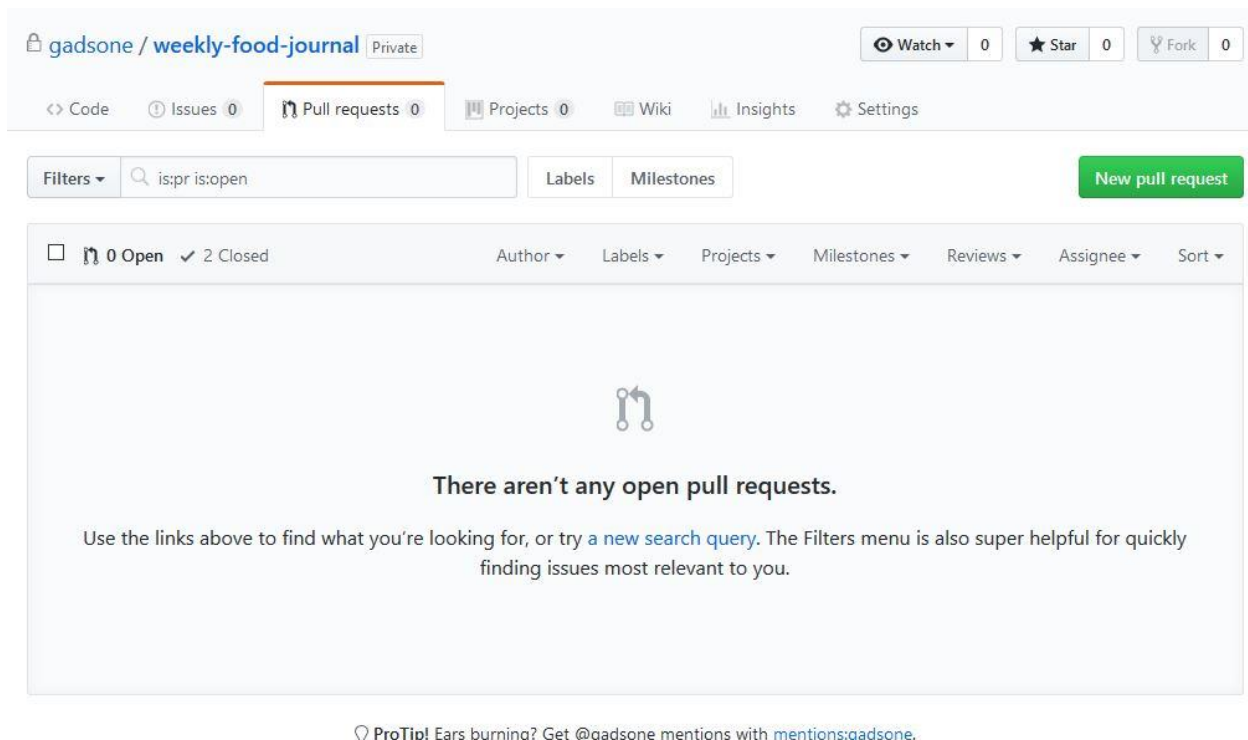
In the online repository, **click on the "Pull requests" tab and select the "New pull request" button. Pull requests** let's others see the changes you have made if you are collaborating with others. If you are not collaborating with others, *pull requests* allow for changes to go to GitHub.com that way you can view the history of these changes.

Under the weekday titled branch with the merge error, **click "Create pull request"** since those will show that they cannot merge automatically due to README edits.

Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also compare across forks.

base: Wednesday ▾ ← compare: master ▾ ✗ Can't automatically merge. Don't worry, you can still create the pull request.

README update

Write | Preview   AA B i ❝ <> ⌖ ⚟ ⚟ ⚟   @ 🔖 ↩

Leave a comment

Attach files by dragging & dropping, selecting them, or pasting from the clipboard.

🅼🅳 Styling with Markdown is supported   **Create pull request**

Reviewers ⚙
No reviews

Assignees ⚙
No one—assign yourself

Labels ⚙
None yet

Projects ⚙
None yet

Milestone ⚙
No milestone

○ 4 commits   📄 2 files changed   💬 0 commit comments   👥 1 contributor

Type in that it is a README update before **clicking "create pull request."**

Add more commits by pushing to the **master** branch on **gadsone/weekly-food-journal**.

⚠ **This branch has conflicts that must be resolved**
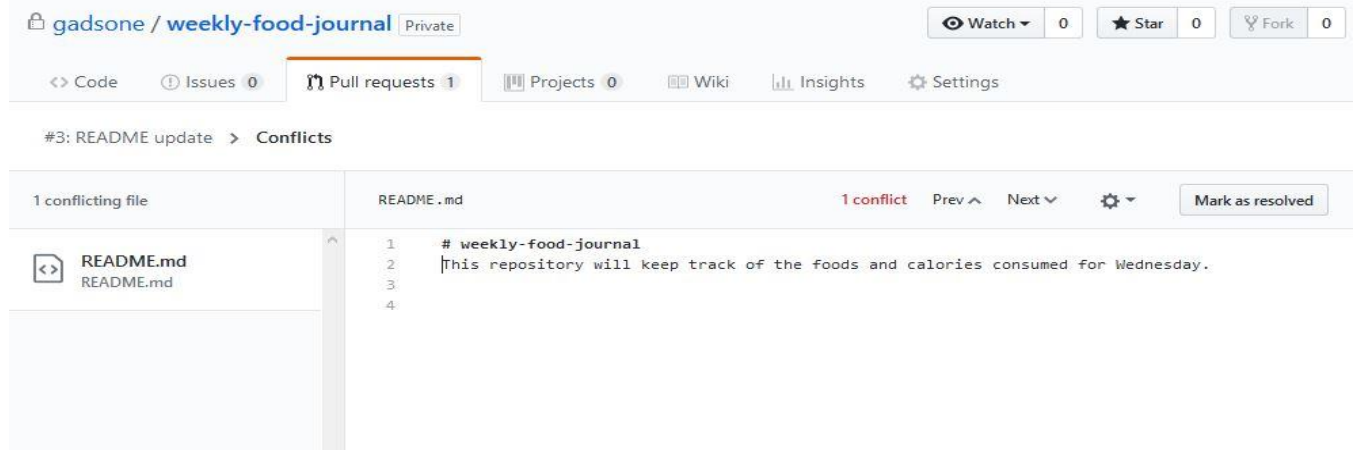Use the web editor or the command line to resolve conflicts.
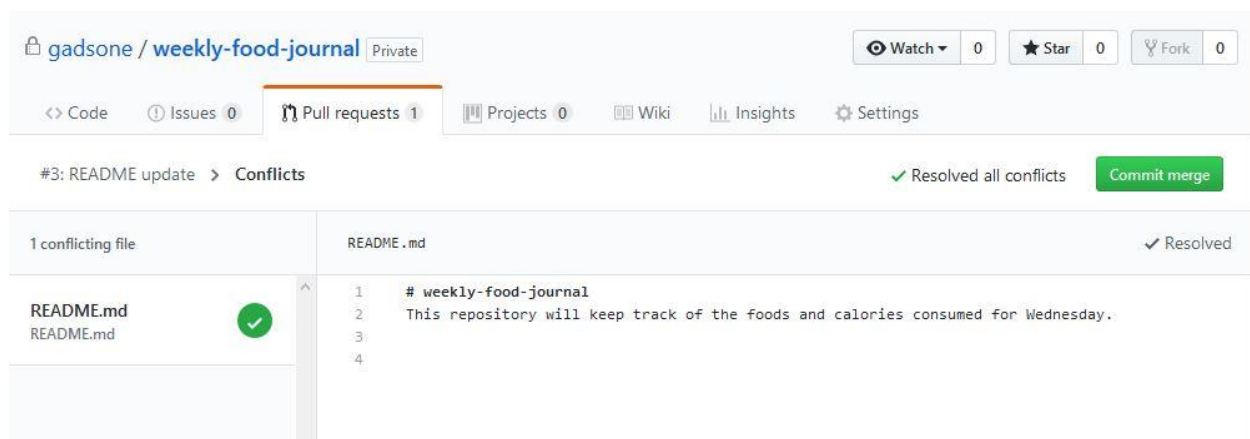
Resolve conflicts

Conflicting files

README.md

Merge pull request ▾   You can also open this in GitHub Desktop or view command line instructions.

Next, **click "Resolve conflicts"** to view and fix README.md file as shown in screenshot.

[18]

The README file will open up. Make changes and decide what you would like to include in the README file. Remove ===== and >>>>> symbols before being satisfied with changes. Click "Marked as resolved" when done.

After, **click "Commit merge"** to save changes to the online repository and solve the conflict with the README file.

Select "Merge pull request" which will merge the Wednesday branch into the master branch.

Select "confirm merge" which will merge README file into master branch.

Check to see if all of your branches have merged into your master branch. **A purple "merged" sign should show next to each branch.**

Since all of your branches are merged, you can ow delete the branches. This is helpful if it becomes disorganized and harder to track changes.

Congratulations! You have made your first repository for a 7-Day Food Log.

[21]