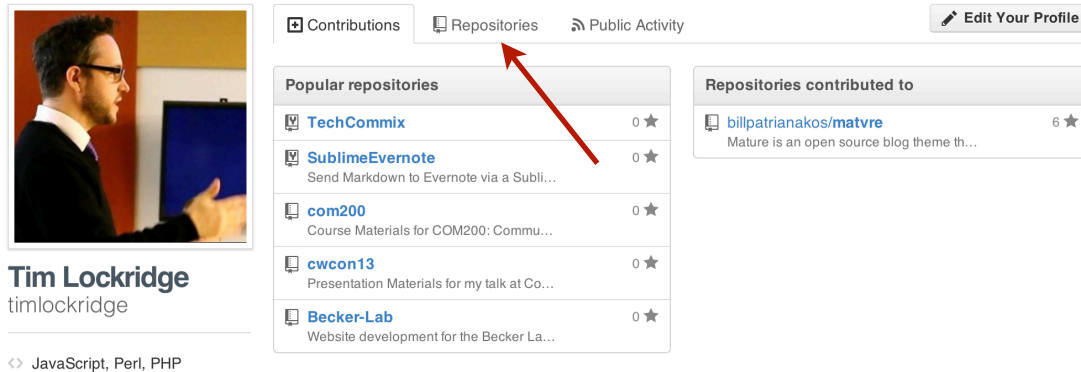
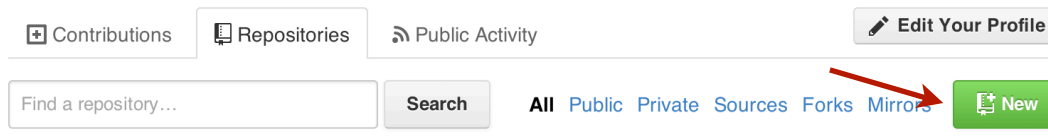


# Github: Configuring & Cloning

1. Log into [www.github.com](https://www.github.com)
2. On your account page, click repositories





3. Click the “new” button to create a repository



4. Name the repository "Coding Check-Ins" and click “create repository”

**Owner** **Repository name**

PUBLIC   timlockridge / coding-check-ins ✓

Great repository names are short and memorable. Need inspiration? How about [potential-octo-meme](#).

**Description (optional)**

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

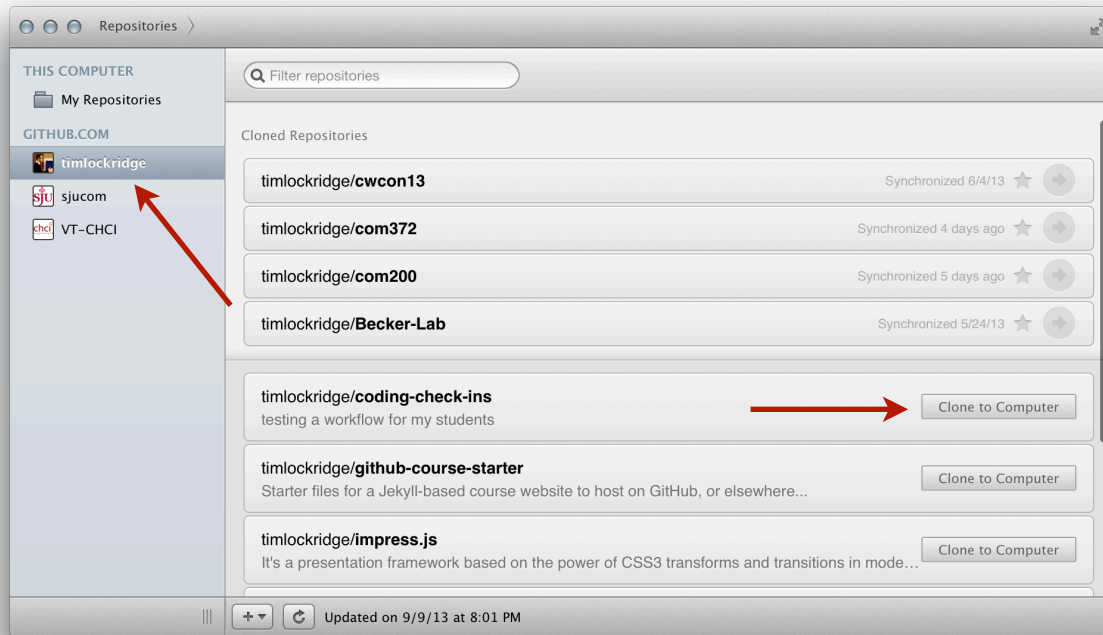
☐ **Private**  
You choose who can see and commit to this repository.

☐ **Initialize this repository with a README**  
This will allow you to `git clone` the repository immediately.

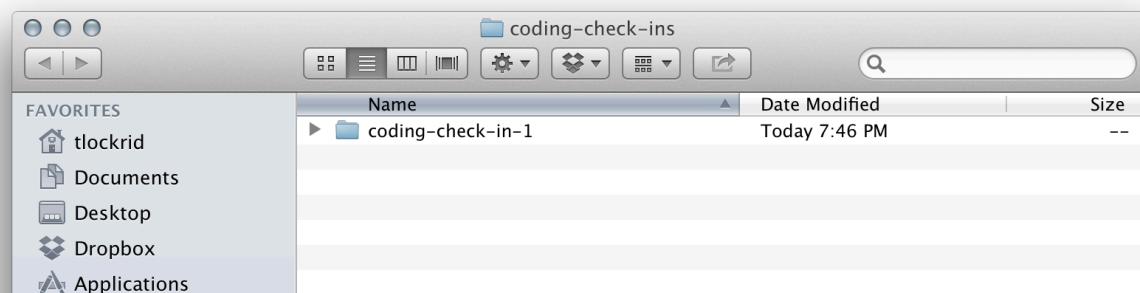
Add .gitignore: **None** | Add a license: **None** ⓘ

**Create repository**

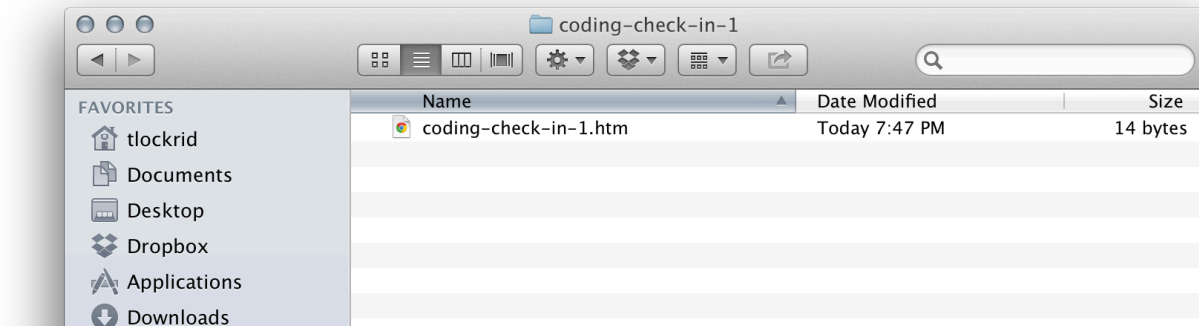
5. Open the Github App on your computer
6. In the left sidebar, beneath the “GITHUB.COM” category, click on your account (you may need to first click repositories in the top left) and click the “clone to computer” button next to your coding-check-ins repository.



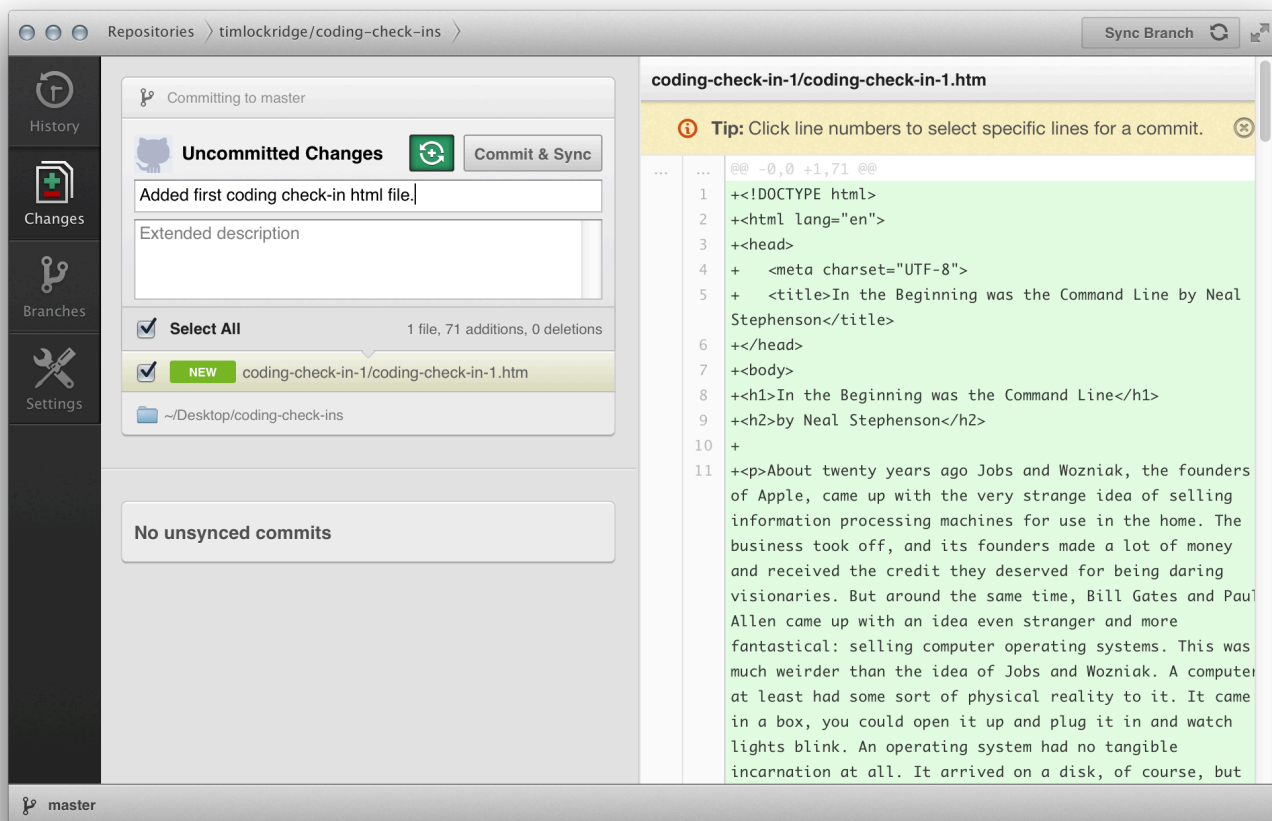
7. The Github app will clone the repository to your computer, but to do this it needs to know where you would like to save it. If you are working on your own computer, you might want to save this in a Documents or Dropbox folder. If you are working on a lab computer, you will likely want to store it on the desktop. Select a location and press “clone.” The Github app will clone the repository, creating a local coding-check-ins folder.
8. Go to the folder in which you told Github to clone the repository, and open the "coding-check-ins" folder. Inside it create a child folder called coding-check-in-1



- Open the coding-check-in-1 folder and copy into it your html file for this week's coding check-in.

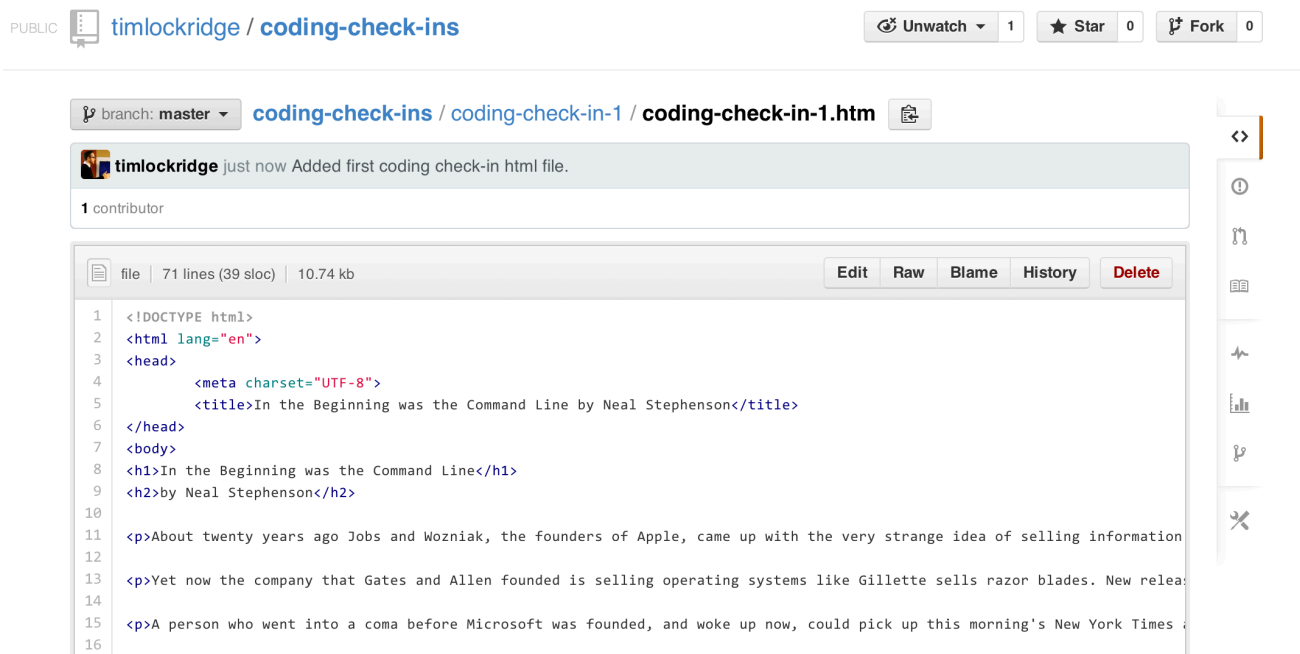


- Return to the Github app and double click on the coding-check-ins repository. It should have detected changes to the folder.
- In the "commit summary" form describe those changes. Make sure the sync button is pressed (it should be green) and press "Commit & Sync"



- **Important:** Your daily use of git/Github will mostly consist of a two step process. When you **commit**, you are documenting a series of changes you made. A typical commit might be *“Changed the color of the header”* or *“Adjusted the margins on all paragraphs.”* When you **sync**, you are updating github with a record of your commits. To streamline this process, the github app allows you to *commit & sync*—telling Github to store a history of your changes.

12. Go to github.com, click on your coding-check-ins repository, and see if your Github page is updated with your new commits.



After you have changed files in a local repository, the Github app will register those changes. Give the changes a meaningful summary, commit them, and then sync them back to github. If ever you lose your work (or have to work on a lab computer), you can clone the repository to a new computer and pick up where you left off!