CPSC1520 – JavaScript 3 Exercise: Events! Functions! JS Resources!

Introduction

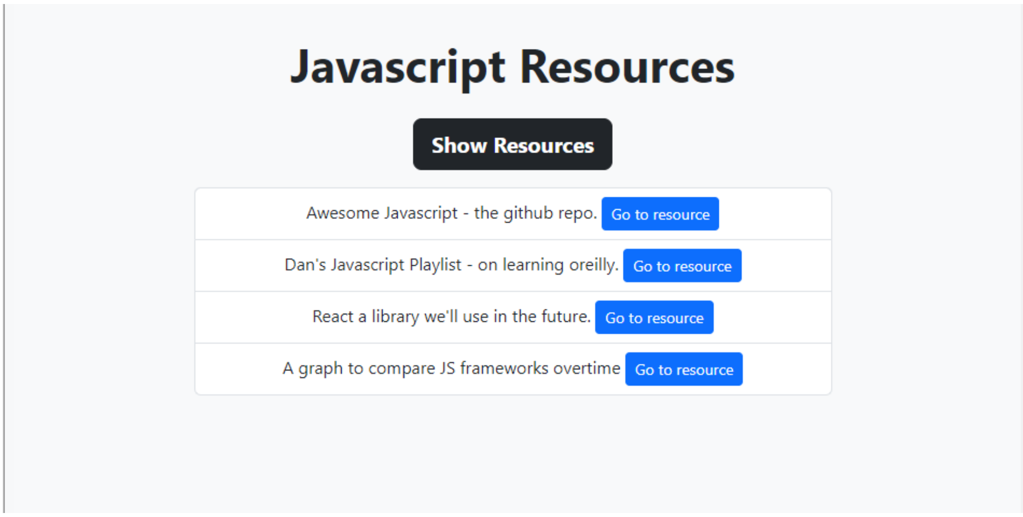
Events in JavaScript are the backbone of getting input from the user and making pages reactive. In class, we've learned how to add event listeners (and handlers), use the event object, prevent default event actions from happening, and a whole bunch more.

In this exercise, we will use this knowledge to keep our "More JavaScript Resources" page open and make it more user-friendly by highlighting links as our users hover over them.

Events Exercise

Exercise Step 1 – Show the Javascript Resources

1. Link your JavaScript file.
2. Select the "div" with the class "javascript-resources" and the "button" with the id of "show-resources.
3. Add an event listener on the button that will listen to the "click" event.
4. In the event handler from the last step remove the class "d-none" from the "div" with the class "javascript-resources" that you selected in the first step. The result should look like the following image.



Javascript Resources

Awesome Javascript - the github repo.

Dan's Javascript Playlist - on learning oreilly.

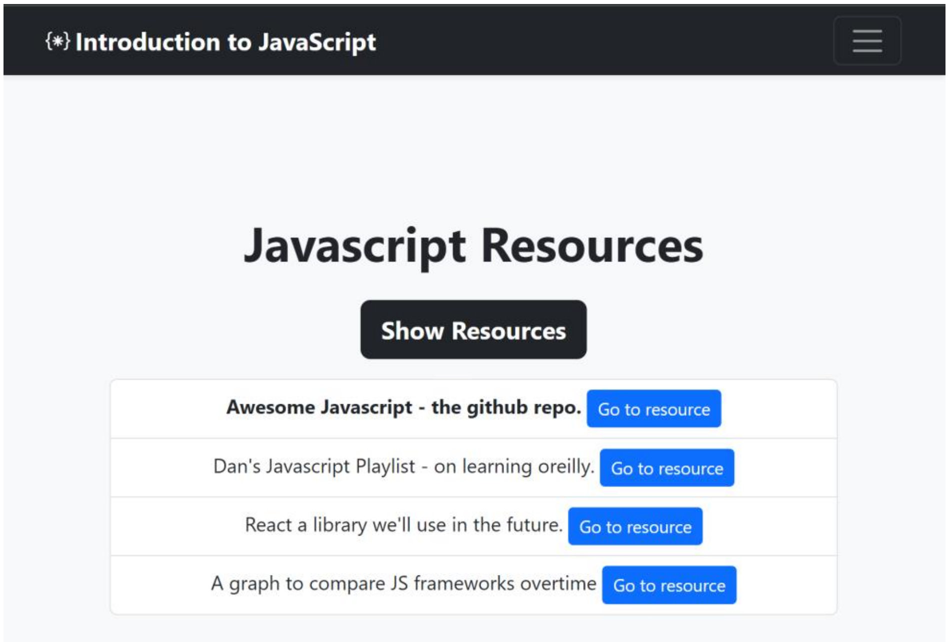
React a library we'll use in the future.

A graph to compare JS frameworks overtime

Note: If you refresh the page the resources should be hidden again.

Exercise Step 2 – Add font boldness on the links when you hover over an item.

1. Add an event listener on element selected in the previous step that will listen to the mouseover event. Ensure that you're passing in the event object to the event handler function.
2. Use console.log to print out the "event.target" to see the link list item elements are the event.target returned
3. On the event.target use the method "classList.add" to add the class named **"fw-bold"**.
    a. When you only hover over the the first link item you should see something like this.



Javascript Resources

Awesome Javascript - the github repo.

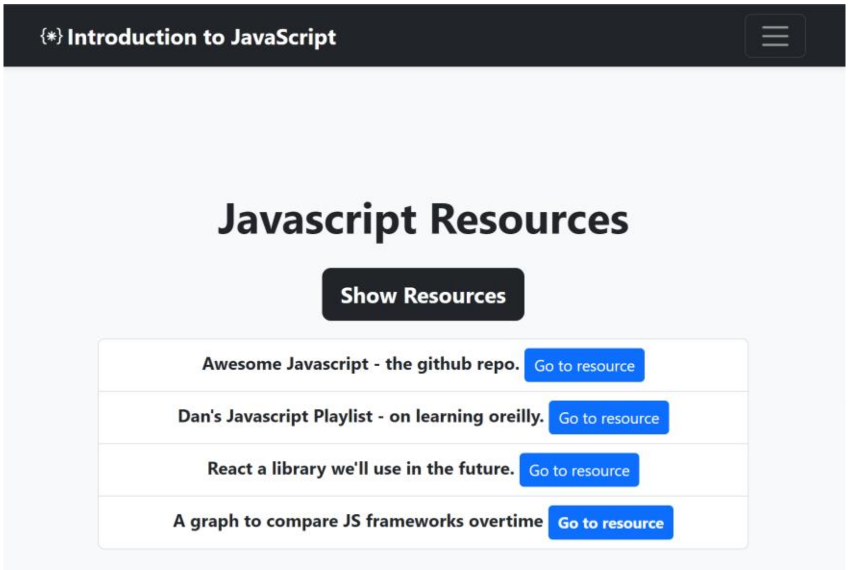
Dan's Javascript Playlist - on learning oreilly.

React a library we'll use in the future.

A graph to compare JS frameworks overtime

Exercise Step 3 – Remove the font boldness on the links when you move your mouse away

1. You see an issue with your application. The idea is that you only want to bold the text of the item you are hovering over. In the image below, all links get highlighted when you hover your mouse over the other items in the list. You need to fix this (probably with another event listener of mouseout).
    a. Image of current functionality.



Javascript Resources

Awesome Javascript - the github repo.

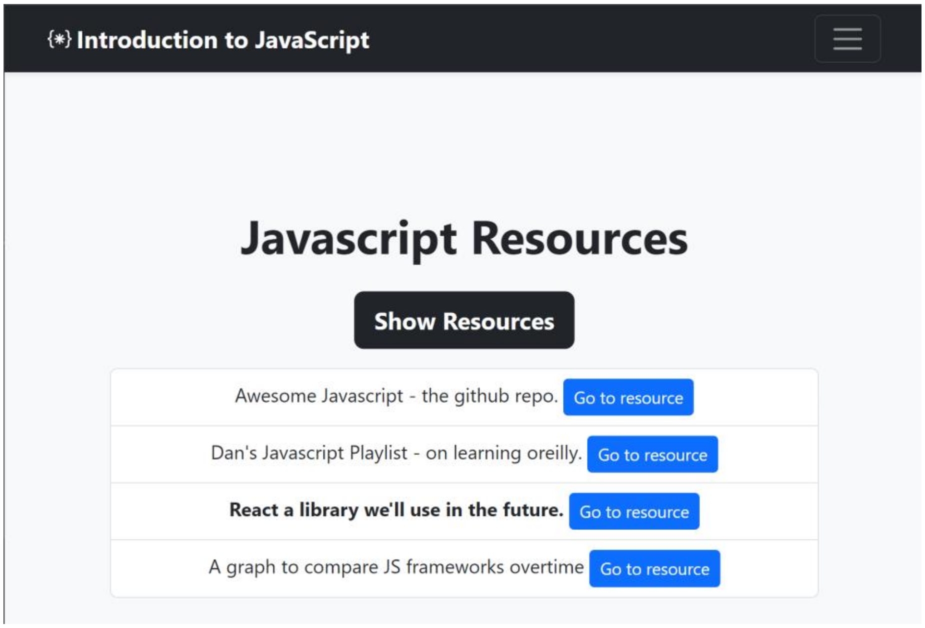
Dan's Javascript Playlist - on learning oreilly.

React a library we'll use in the future.

A graph to compare JS frameworks overtime

2. Add another event listener on the div list group selected in the first step (the last section) that listens to mouseout events.

3. In the event handler, on the event.target use the method “classList.remove” to remove the class named “**fw-bold.**” The handler will now remove the class when you move away from an item in the list.
    a. When you **test the third item** (after hovering over the other items) it should now look something like this.



Javascript Resources

Awesome Javascript - the github repo.

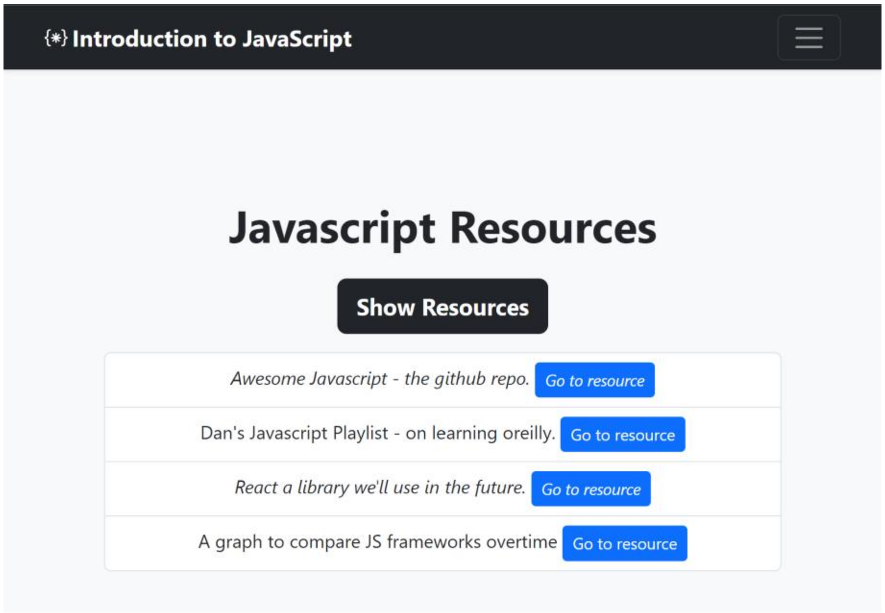
Dan's Javascript Playlist - on learning oreilly.

React a library we'll use in the future.

A graph to compare JS frameworks overtime

Exercise Step 4 – Italicize a clicked item.

1. Add an event listener to the list to listen to the “click” event.
2. In that event handler, use the same techniques in the previous steps to add the class “*fst-italic*” on the “event.target”.
    a. The result when you click the first and third items is as shown below.



Javascript Resources

Awesome Javascript - the github repo.

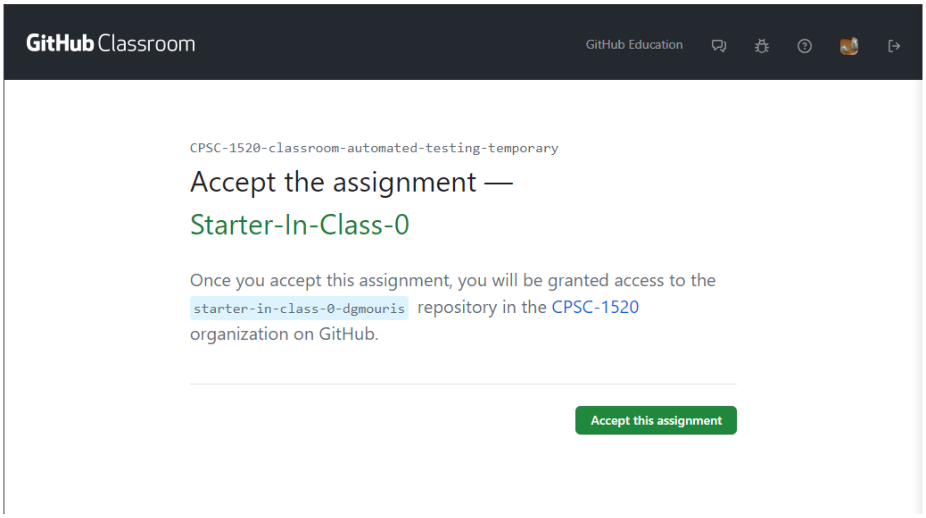
Dan's Javascript Playlist - on learning oreilly.

React a library we'll use in the future.

A graph to compare JS frameworks overtime

Exercise Step 4 – Push up your code to github (accepting this assignment)

1. Open the link given and accept the assignment. Your link should look something like this. Note the image will be different because you'll accept the assignment specified.

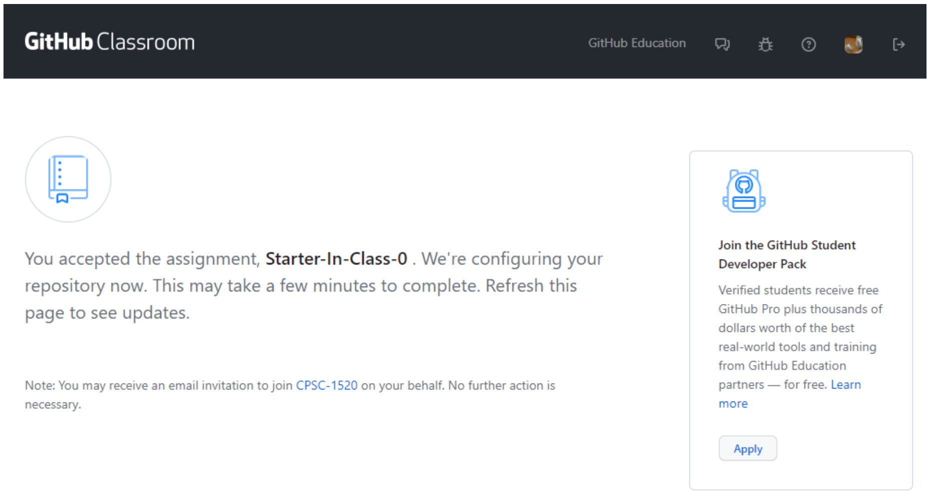


CPSC-1520-classroom-automated-testing-temporary

Accept the assignment — Starter-In-Class-0

Once you accept this assignment, you will be granted access to the `starter-in-class-0-dgmouris` repository in the CPSC-1520 organization on GitHub.

You'll see a page like this.



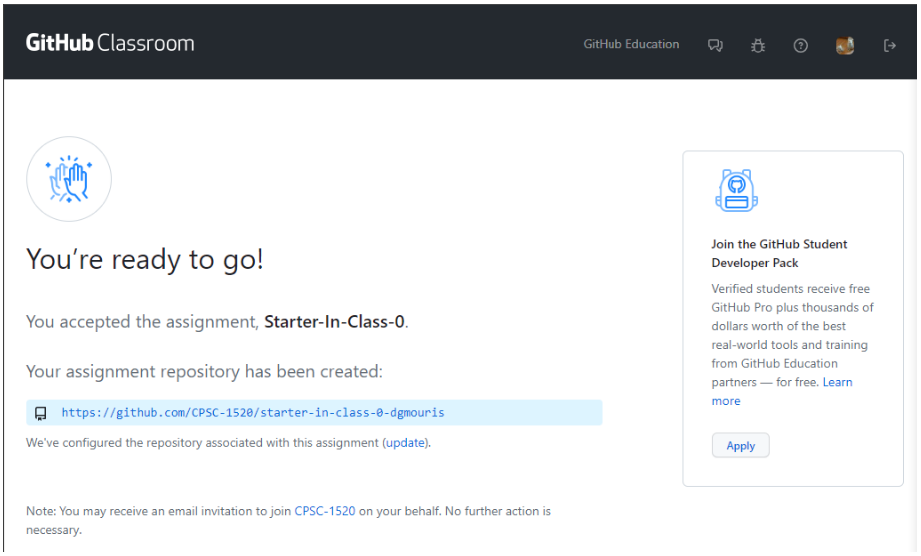
You accepted the assignment, **Starter-In-Class-0** . We're configuring your repository now. This may take a few minutes to complete. Refresh this page to see updates.

Note: You may receive an email invitation to join CPSC-1520 on your behalf. No further action is necessary.

Join the GitHub Student Developer Pack

Verified students receive free GitHub Pro plus thousands of dollars worth of the best real-world tools and training from GitHub Education partners — for free. Learn more

One you're repo is ready the page should look like the following.



You're ready to go!

You accepted the assignment, **Starter-In-Class-0**.

Your assignment repository has been created:

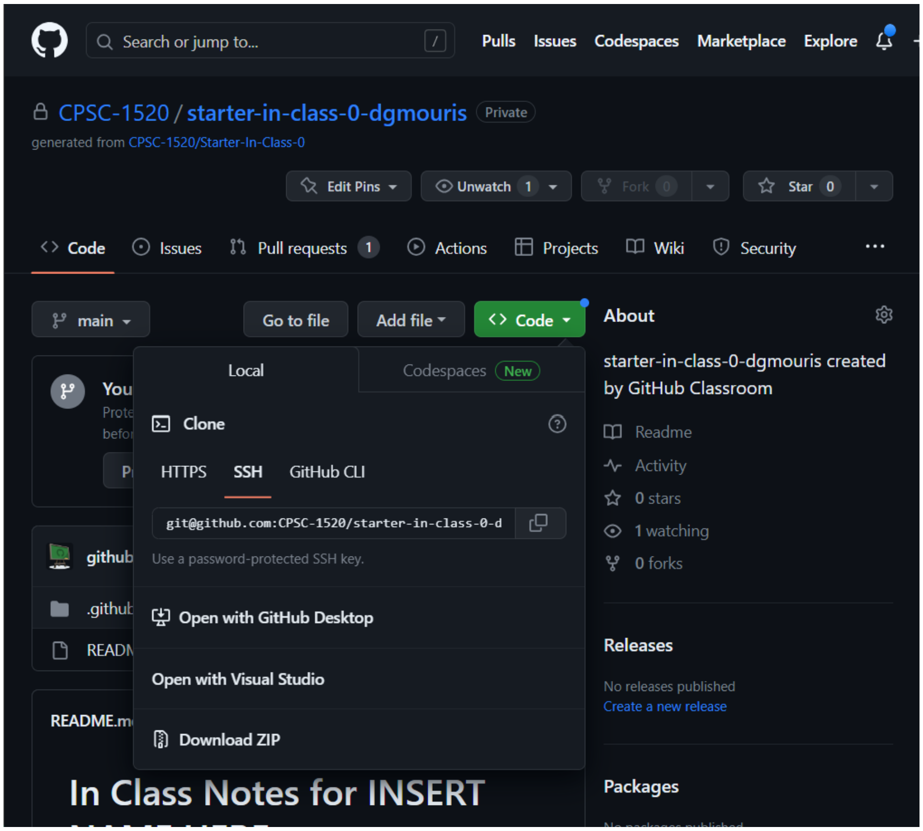
https://github.com/CPSC-1520/starter-in-class-0-dgmouris

We've configured the repository associated with this assignment (update).

Verified students receive free GitHub Pro plus thousands of dollars worth of the best real-world tools and training from GitHub Education partners — for free. Learn more

Note: You may receive an email invitation to join CPSC-1520 on your behalf. No further action is necessary.

2. You should see the page below once you click on the link highlighted in blue. Click the button that says "Code." You'll need to select "HTTPS" unless you've set up "SSH" (you can also set up GitHub CLI". Click on the copy icon once you've selected the proper icon.

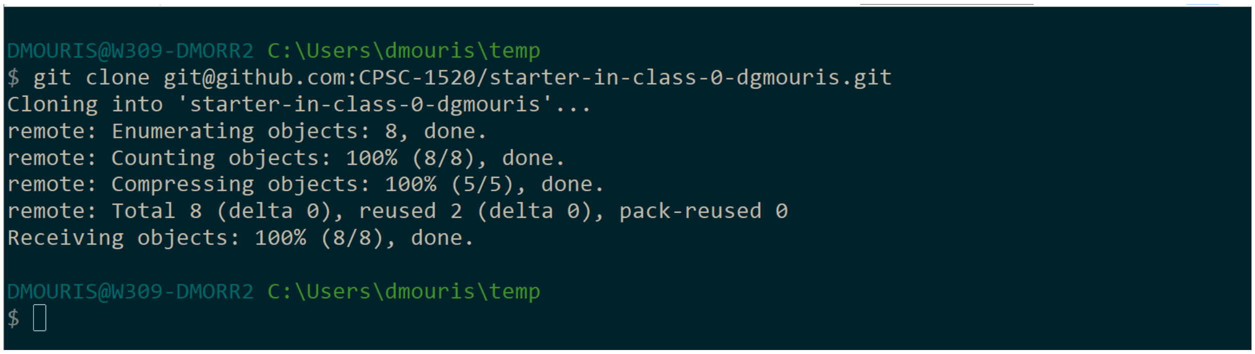


git@github.com:CPSC-1520/starter-in-class-0-d

Use a password-protected SSH key.

starter-in-class-0-dgmouris created by GitHub Classroom

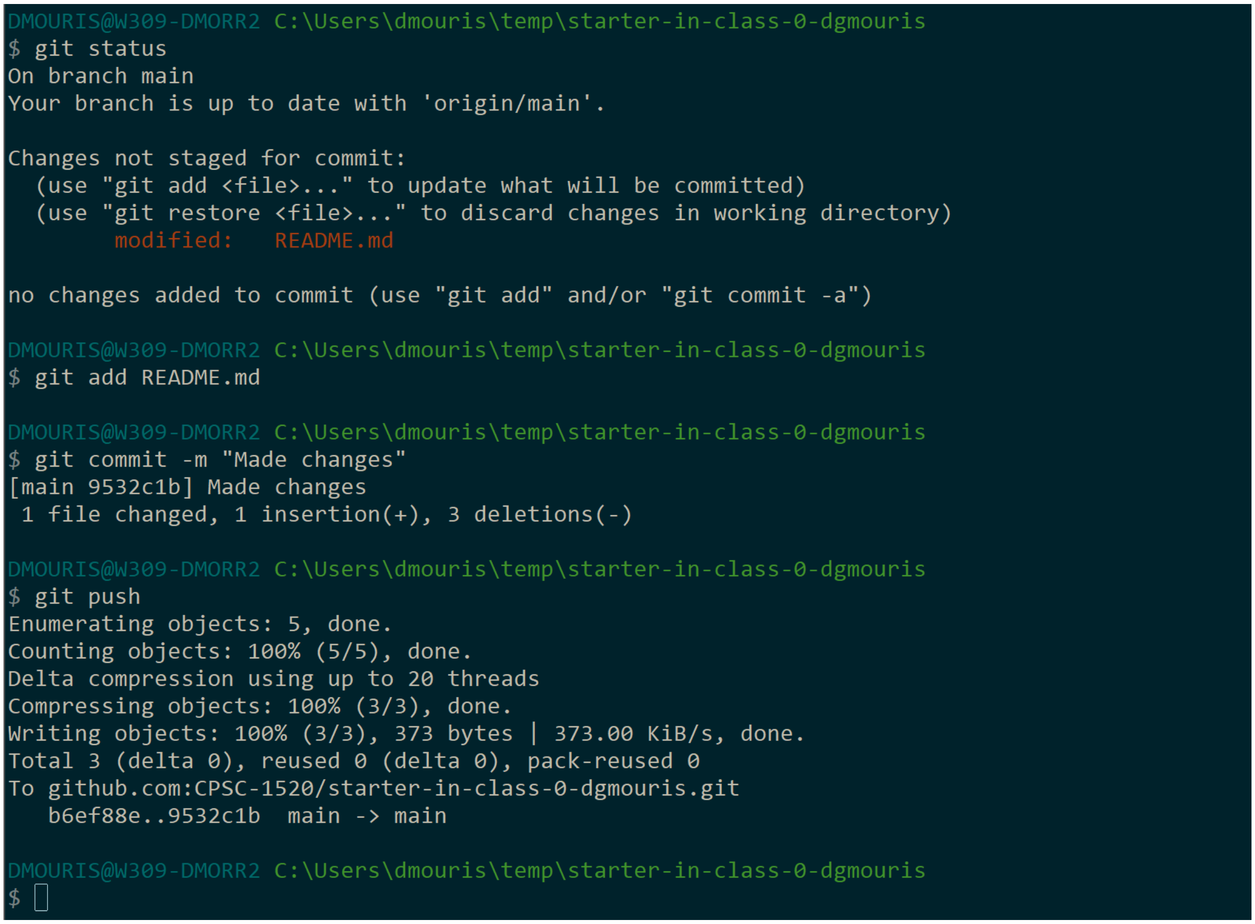
3. Clone the repository in your console (or if you're using GitHub Desktop) using the "git clone REPO_URL" command.



```
DMOURIS@W309-DMORR2 C:\Users\dmouris\temp
$ git clone git@github.com:CPSC-1520/starter-in-class-0-dgmouris.git
Cloning into 'starter-in-class-0-dgmouris'...
remote: Enumerating objects: 8, done.
remote: Counting objects: 100% (8/8), done.
remote: Compressing objects: 100% (5/5), done.
remote: Total 8 (delta 0), reused 2 (delta 0), pack-reused 0
Receiving objects: 100% (8/8), done.

DMOURIS@W309-DMORR2 C:\Users\dmouris\temp
$ 
```

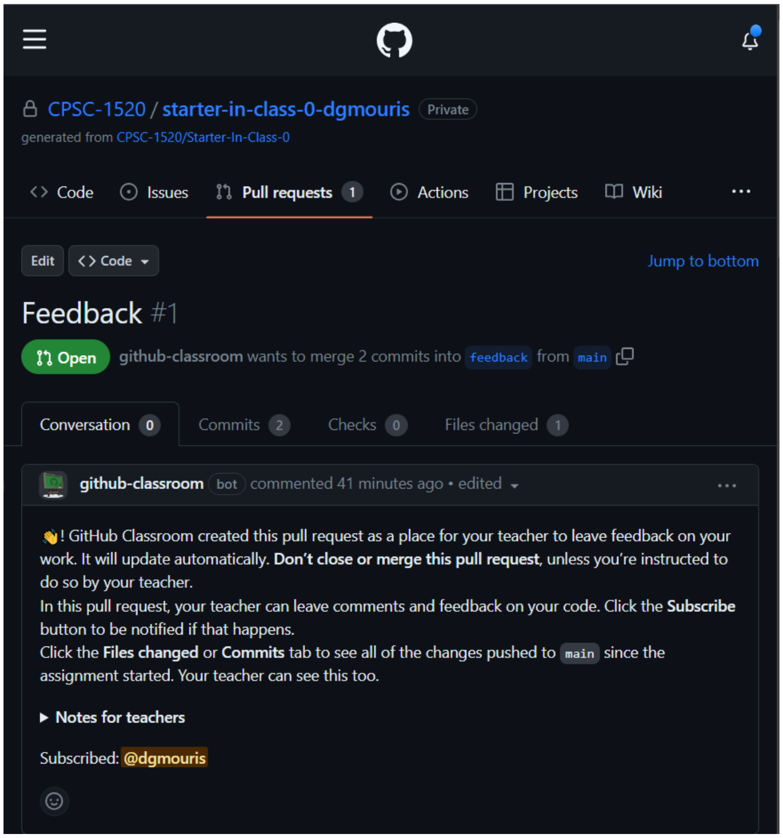
And go into this folder.

4. Make your changes, then add them to staging (using "git add .") and commit them (using "git commit -m "CHANGE THIS MESSAGE"). Once committed, push them up to GitHup (using "git push") it should look like below.



```
DMOURIS@W309-DMORR2 C:\Users\dmouris\temp\starter-in-class-0-dgmouris
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   README.md

no changes added to commit (use "git add" and/or "git commit -a")

DMOURIS@W309-DMORR2 C:\Users\dmouris\temp\starter-in-class-0-dgmouris
$ git add README.md

DMOURIS@W309-DMORR2 C:\Users\dmouris\temp\starter-in-class-0-dgmouris
$ git commit -m "Made changes"
[main 9532c1b] Made changes
 1 file changed, 1 insertion(+), 3 deletions(-)

DMOURIS@W309-DMORR2 C:\Users\dmouris\temp\starter-in-class-0-dgmouris
$ git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 20 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 373 bytes | 373.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:CPSC-1520/starter-in-class-0-dgmouris.git
   b6ef88e..9532c1b  main -> main

DMOURIS@W309-DMORR2 C:\Users\dmouris\temp\starter-in-class-0-dgmouris
$ 
```

5. If you click "Pull Requests" and then the first item called "Feedback" you should see your commit (seen at the bottom).



Feedback #1

Open github-classroom wants to merge 2 commits into `feedback` from `main`

👋! GitHub Classroom created this pull request as a place for your teacher to leave feedback on your work. It will update automatically. **Don't close or merge this pull request**, unless you're instructed to do so by your teacher.
In this pull request, your teacher can leave comments and feedback on your code. Click the **Subscribe** button to be notified if that happens.
Click the **Files changed** or **Commits** tab to see all of the changes pushed to `main` since the assignment started. Your teacher can see this too.

▸ **Notes for teachers**

Subscribed: @dgmouris

6. Upload the link of your repository to Moodle.

Grading

I'll give full marks if:

- Your code is pushed up to github correctly.
- You didn't change the html.
- List element is selected "event.target" is used, other janky ways will not be accepted.
- The one single list item is bolded when I mouse over it.
- When I click one of the items, that specific item is italicized
- When I open your project it should behave exactly as described above.

If you don't follow the instructions you'll get a zero. There are no marks in between.