



Managing Merge Conflicts

A quick reference guide to complete the steps in the course.

<https://lab.github.com/courses/managing-merge-conflicts/>

1. Create a pull request

- 1.1. On the **Code** tab, click the `_data/skills.yml` file
- 1.2. Click the pencil (✎) icon to enter edit mode
- 1.3. In the file, add a skill or two that you're proficient in
- 1.4. Scroll to the bottom and select the "Create a new branch for this commit and start a pull request" option
- 1.5. Replace the default patch branch name with `change-skills`
- 1.6. Select **Propose file change**
- 1.7. Write some text in the **Leave a comment** field of the pull request
- 1.8. Click **Create pull request**
- 1.9. ⚠️ Don't merge yet! ⚠️ Refresh the pull request to receive the next comment/instructions.

2. Merge the pull request

- 2.1. Click **Merge** pull request below
- 2.2. Click **Confirm merge**
- 2.3. Click **Delete branch**.


3. Resolve a simple conflict

- 3.1. In the "This branch has conflicts that must be resolved" section of the Pull Request (below, keep scrolling), click **Resolve conflicts**
- 3.2. Look for the highlighted sections that begins with `<<<<<< update-config` and ends with `>>>>>> master`. These markers are added by Git to show you the content that is in conflict
- 3.3. Remove the changes made on the `master` branch by deleting all of the content below the `=====` and above `>>>>>> master`
- 3.4. Next, remove the merge conflict markers by deleting the following lines:
`<<<<<< update-config`
`=====`
`>>>>>> master`
- 3.5. **Optional:** If you're thinking about using this resume, you can edit the `_config.yml` file with your own information now. Change any of the lines within the file, even outside of where the markers were. More about this below
- 3.6. With the merge conflict markers removed, click **Mark as resolved**
- 3.7. Click **Commit merge**.

4. Merge the first resolved pull request

- 4.1. Click **Merge pull request** now
 - 4.2. Click **Confirm merge**
 - 4.3. Click **Delete branch** and get ready for your next activity.
-

5. A: Create your own conflict

- 5.1. Click on the **Files changed** tab in this pull request
- 5.2. Click the pencil () icon found in the top right-hand corner of the `_data/education.yml` file that had been previously modified
- 5.3. Modify the content in the `degree:`, `uni:`, `year:`, and `summary:` lines
- 5.4. Scroll to the bottom of the page and enter a commit message for your change
- 5.5. Click the **Commit changes** button, making sure the “Commit directly to the add-education branch” option is selected.

B: Merging similar changes

- 5.6. Click on your second pull request
 - 5.7. Follow the instructions in that pull request to approve it
-

6. Conflicting change

- 6.1. In the pull request status window below, click **Add your review**
 - 6.2. When the pull request window displays, select **Approve** and click **Submit review**
-

7. Resolve conflicts you created

- 7.1. In the “This branch has conflicts that must be resolved” section of the pull request, click **Resolve conflicts**
-

8. Merge the third pull request

- 8.1. Click **Merge pull request** below
 - 8.2. Click **Confirm Merge**
 - 8.3. Click **Delete branch**
-



9. Resolve conflicts in the Advanced Conflicts pull request

- 9.1. Click **Resolve conflicts**
- 9.2. On the left, you will notice two files listed: `_data/experience.yml` and `_data/interests.yml`. Let's start with `experience.yml`
- 9.3. Notice there are two distinct sets of conflict markers. This is because multiple sections of the file were modified on both branches, so Git identified the two changes within the file as two separate conflicts
- 9.4. **Optional:** Again, if you're trying to use this course to build a resume, you can resolve the conflicting files with your own information
- 9.5. With the merge conflicts resolved and the markers removed in the `experience.yml` file, click **Mark as resolved**
- 9.6. GitHub will present the next file with conflicts, `interests.yml`
- 9.7. This file has some extra merge commit markers. Simply select the interests you would like to list and remove the others (and all those extra conflict markers)
- 9.8. When you are finished, click **Mark as resolved**
- 9.9. Click **Commit merge**.

10. Merge fourth pull request

- 10.1. Click **Merge pull request** below
- 10.2. Click **Confirm Merge**
- 10.3. Click **Delete branch**

11. Enable GitHub Pages

- 11.1. Click on the **Settings** tab
- 11.2. Scroll to the "GitHub Pages" section
- 11.3. In the **Source** drop-down, select `master branch`
- 11.4. Click **Save**
- 11.5.  Warning!  Make sure you don't see any errors after you select save. If you do, your page won't build correctly and this step will be incomplete