

Branching Exercise:

Mahad Osman

Oct 1, 2022

Part I

Answer the following questions:

- What **git** command creates a branch?

```
git branch

//To name move out to the branch at the same time
git checkout -b new-branch
```

- What is the difference between a fast-forward and recursive merge?

▼ A fast-forward merges commits what happens with one set of commits that will stack chronically, while a recursive merge will continue to have two parents even after the merge occurs.

▼ reference: <https://www.geeksforgeeks.org/merge-strategies-in-git/#:~:text=The merge commit continues to have two parents.&text=Note%3A There is nothing right,it is of multiple lines>.

- What **git** command changes to another branch?

```
git checkout
```

- What **git** command deletes a branch?

```
#if the branch has never been used or already merged
git branch -d

#if the branch has not yet been merged but has had changes
git branch -D
```

- How do merge conflicts happen?

▼ Occurs when changes are committed to the same file from two different branches
ex. Changes have been made on say the master and another file on the same lines of code then a merge is initiated.

Part II

Practice with fast forward and recursive merges! Make a branch and add and commit onto it and merge it back into master.

```
#Made an initial Commit with a simple html and js file with an h1
mahadosman@Mahads-MacBook-Pro git_merge_conflicts % git checkout master
Switched to branch 'master'
mahadosman@Mahads-MacBook-Pro git_merge_conflicts % ls
app.js    index.html

#created a branch with to add a css file to style the h1 and background
mahadosman@Mahads-MacBook-Pro git_merge_conflicts % git checkout add-colors
Switched to branch 'add-colors'
mahadosman@Mahads-MacBook-Pro git_merge_conflicts % git status
On branch add-colors
nothing to commit, working tree clean
mahadosman@Mahads-MacBook-Pro git_merge_conflicts % ls
app.css  app.js    index.html

#The changes that exist inside of app.css
mahadosman@Mahads-MacBook-Pro git_merge_conflicts % cat app.css
h1{
    display: flex;
    text-align: center;
    color: gold;
    font-size: 10rem;
}

body{
    background-color: black;
}%

#A fastforward merge to add in the css file
mahadosman@Mahads-MacBook-Pro git_merge_conflicts % git merge add-colors
Updating 2c30a5f..fdc7878
Fast-forward
 app.css      | 10 ++++++++
 index.html  |  1 +
 2 files changed, 11 insertions(+)
```

```
create mode 100644 app.css

#added another branch to add an h2 and showcasea reculsive merge
mahadosman@Mahads-MacBook-Pro git_merge_conflicts % git status
On branch master
All conflicts fixed but you are still merging.
    (use "git commit" to conclude merge)

Changes to be committed:
  modified:   app.css
  modified:   index.html

mahadosman@Mahads-MacBook-Pro git_merge_conflicts % git commit -m "adding in the h2"
[master b2aede] adding in the h2
mahadosman@Mahads-MacBook-Pro git_merge_conflicts % git status
On branch master
nothing to commit, working tree clean
```

Try to create your own merge conflict by modifying the same file on two separate commits on two separate branches.

- ▼ See above for a merge conflict. I tried forcing a reculsive merge before commit the changes that were in the staging area hence the merge conflict issue that occurred.
- ▼ Another merge conflict example that could be caused was modifying my background-Color in app.css in both my master branch and add-h2 branch than try to force a merge.