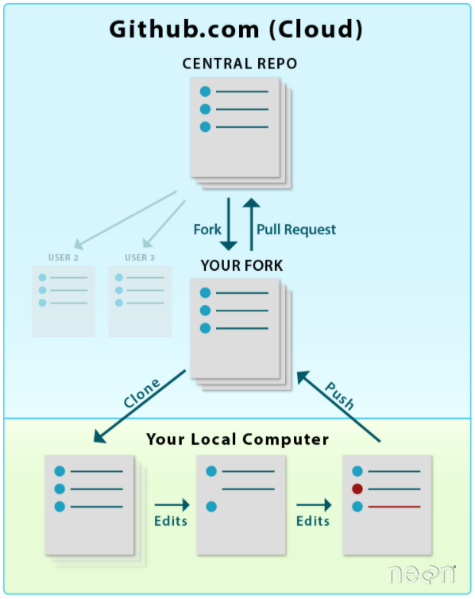
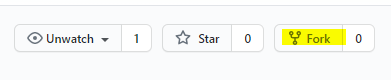
Forking:

A [GitHub fork](https://help.github.com/articles/fork-a-repo/) is a copy of a repository (repo) that sits in your account rather than the account from which you forked the data from. Once you have forked a repo, you own your forked copy. You can edit the contents of your forked repository without impacting the parent repo.



Procedure for Forking:

**#Step 1- Fork repo by clicking the fork button in the upper right-hand corner of a repo page.**



**# Step 2- Clone your Fork onto your local computer. You now have a creative space. *https://youtu.be/O72FWNeO-xY***

git clone *https://github.com:youruserID/carsales.git*

**# Step 3- Add upstream connection to team repo.**

git remote add upstream [*https://github.com:danawoodruff/carsales.git*](file:///C:\Users\plqua\Downloads\%20https:\github.com:danawoodruff\carsales.git)

**# Step 3a-**  git remote -v

Procedure to update your Fork and push your work upstream:

***CRITICAL STEPS, DO NOT SKIP ANY!!!***

***From your cloned file on your local computer open GitBash (Windows PC):***

**#Step 1-** **Create your branch**

git checkout -b *Branchname*

**#Step 2- Add your files to queue**

git add -A

**#Step 3- Commit files in queue**

git commit -m " I have completed some work”

**#Step 4- Switch to the main branch**

git checkout main

**#Step 5- Fetch from UPSTREAM (from team captain)**   
git fetch upstream

**#Step 6- Merge those new files to your main branch (your own repo)**  
git merge upstream/main

git push origin main

***#OPTIONAL: You can perform a git log to see latest commit changes from UPSTREAM***  
git log –oneline

**#Step 7- Switch to your branch**

git checkout *Branchname*

**#Step 8- Merge all the latest changes onto your branch.**

git merge main

**#Step 9- Push all changes to your forked repo after you’ve done work.**

git push origin *branchname*

COMPARE/Pull Request

**#Step 10- Go to GitHub repo, and a click the button to open a PR: (Pull request).**