1. Forking
   Forking creates a copy of your repository that interns can work in. A fork can operate as a full universe distinct from your repository for them to work in, but you can link your repository to theirs to share files to them this way. Forking is not designed for your interns' changes to change your repository.

   (Note: you can get around this and have your interns push from their fork to your repository by adding your interns as collaborators in your repository settings. Then they can push from their fork to your repository by typing **git push upstream master**. This will not merge changes in any elegant way, it will literally just push their stuff onto your repository.)

   a. Creating a fork: click "Fork" on github browser
   b. Cloning fork onto your computer: **git clone <forked URL>**
      When you clone a repository, GitHub automatically creates a remote called "origin" that lets you interact with the online repository and make/download changes
   c. Adding a remote (connecting your local repository (on your computer) to a remote repository (online)): **git remote add <name> <url>**
      For example to have your interns have a remote that points to the original repository (the one they forked from), have them type "**git remote add upstream <URL of your repository>**. This adds a remote called "upstream" that points to your repo.
   d. Have your interns add you as a collaborator to their forks (in settings on browser) so you can see what they do
   e. To see your remotes: **git remote -v**
   f. To add a change to the staging area: **git add <file-name>**
   g. To push those changes to your remote repository: **git push origin master**
   h. To sync your forked repository with any changes made to original: **git pull upstream master** (ie adding any new files from main online repository to your computer)
2. Branching
   -- Branching is a system designed so we can work on different versions of the same files at the same time. Branching is designed so changes will eventually be merged.
   -- Every git repository automatically has a branch called "master." When you're working on different branches, you want them all to merge to master eventually
   -- there is this thing called the "Head pointer" which just keeps track of which branch you're working on.
   -- you will need to add your interns as collaborators on your repository in your repository settings in order to use branching

a. To make a branch: **git checkout -b branch-name**
   This creates a branch called branch-name and switches you to that branch
b. To switch to any branch: **git checkout branch-name**
c. To see what branch you are on: **git branch**
d. To commit a change to the branch you are working on:
   git add <file name>
   git commit -m "message"
e. To merge: switch to master (or whatever branch you want to merge into):
   git checkout master
   git merge <branch you made commited changes on>

   This merges changes *from* <branch you made committed changes on> *to* master, but not the other way around. To merge from master to another branch go to that branch and type git merge master.

   Merge conflicts will happen if there are changes made on the same line(s) of the same files on different branches. Git will tell you there is a merge conflict by saying "`Merge conflict in <filename> Automatic merge failed; fix conflicts and then commit the result`"

f. To fix the conflicts go open the file; git will show you what went wrong. Resolve the changes (manually), git add/commit the file again, and retype git merge <branch you made committed changes on>

How branching could work for your interns:

- You could have your interns clone your repository and connect to your repository as a remote.
- Then you could have each intern make their own branch and work on some code, commit and add their changes
- You could control master and merge their changes, or you could have them each merge their changes to master themselves.

Additional resources:

1. Katy's full github tutorial: https://github.com/katyabbott/BridgeUP-STEM-Oceans-Six/blob/master/git-instructions.md
2. Katy's git cheat sheet: https://github.com/katyabbott/BridgeUP-STEM-Oceans-Six/blob/master/GitHub_steps.pdf
3. Youtube ;)