Setup:
0. create github.com account if you don't have one
1. fork the github repo that you want to work upon in the github interface (this will give your own repo where you can make changes and try things on your own)
2. **git clone *yourgithubrepo*** (you get the *yourgithubrepo* from github interface, this creates a local folder with the files from the repo)
3. **cd** to the new folder created by the previous command
4. **git remote add upstream *upstreamgithubrepo*** (the *uptreamgithubrepo* is the one that you forked from, makes it easier to track changes in the upstream)

Workflow:
1. **git fetch upstream** (to get any changes from upstream before committing)
2. **git merge upstream/master** (to merge the changes)
Alternatively, you can run one command "**git pull upstream master**" instead of fetch and merge, make sure though you don't have any uncommitted changes, this will wipe them.
3. work on the project: modify, add, remove files
4. **git add .** (or "**git add *filename***") to track the changed files
5. **git rm *filename*** (to remove files)
6. **git commit -m "*descriptive commit message*"**
7. **git push origin master** (this pushes master branch to your github repo, you can also push other branches than master)
8. from Github you click on green Compare, Review, create Pull Request and **create a pull request** for the person who manages the upstream repo to pull your changes in their repo.

Useful git commands:
git status (shows you which files were modified which files are untracked)
git diff (shows diffs between your changed files and the last commit)
git log (shows the commits)
git diff HEAD^ (shows the difference from the previous commit)
git checkout -- *filename* (reverts the file to the version from the commit, useful if you messed up and you want the original file)
git checkout -- . (reverts all files, all changes since commit are lost)
git checkout *branchname* (change branch, add -b switch if the branch doesn't exist yet and you want to create it)