Notes

* Git allows me to keep track of the code I write and work with it:

**PULL before you PUSH!**

* + Git status > checks on where the code is at
  + Git add > send the file with the code to the staged section.
    - It can be reversed by writing in the terminal: git reset HEAD filename
  + Git commit > prepares the file to be send over to github I can add comments about the changes I am making.
    - Here I can go back in case of a mistake by writing: git reset –soft HEAD~1
  + Git push origin master > sends the committed file into github.
    - If pushed a mistake it can be fixed by typing in the terminal: git revert HEAD
  + Git diff > we see differences between the new html file and the old one
  + Git fetch > is going to search for changes at github and bring them to my local computer
  + Git merge origin/master > this will take those changes from the github and actually apply them in my computer.
  + Git pull > does the same as fetch and merge at the same time
  + Normally rejected push requests occur because someone else has made changes to the files in github. So the good thing to do to avoid that, is to firs pull, take a look at the changes and then push.
  + If when we do a pull request there is a conflict, then we need to resolve the problem on the code editor after that we need to do all the process again: add > commit > push
  + A branch is a copy of the master this avoid us from making changes to the master, so we can experiment.
  + Git branch –a > allows us to see all the branches that we have
  + Git branch branchname > will create a new branch
  + The \* shows us where we are located right now.
  + Git checkout branchname > allows us to switch to a different branch.
  + Git checkout –b branchname > allows us to create a new branch and immediately switch to that one.
  + Git branch –m branchname newname > changes the name of a specific branch
  + Git branch –d branchname > will delete a branch, you need to checkout from it before deleting a branch
  + Git merge > will merge independent branches
  + Git merge branchname > merge this branch into the one I am currently at (fast forward method)
  + Git branch push origin –delete branchName > will delete the remote branch stored in github.
  + Git rebase: moves a branch up to a new base commit, this should limited to local changes and not repositories that are being shared with others.
  + Git log > see the history of the commits
    - Git log –author=username > shows the commits made by an specific team member
    - Git log –graph –decorate –oneline > shows the commit history as a graph
    - Git log –n number > will show the most recent number of commits made to a particular branch
    - Git show merge number > will show the particular lines that have been changed in that commit either adding or substracting
    - Git diff firstBranchName SecondBranchName > will show the differences in the code of both branches
  + Git clone is like copy paste all into a new folder
  + Fork makes a copy of the repository in my account