**Git Tutorial**

**How to git push into somebody else’s GitHub repo (macOS X)**

GH: GitHub **|** T: Terminal | F: Finder

**Permissions (only required the first time)**

1. GH: Master user invite collaborators to the GitHub repo
2. GH: Collaborator user: accept invite

**Creating copy of master into local folder (only required the first time)**

1. T: cd into folder where you want to house a copy of the GitHub master folder
2. T: git clone <link of GitHub>
3. T: git status
   1. Check that there are no files to commit

**Creating new branch in local folder**

1. T: git branch
   1. Check what branch you are in. You should be in the master
2. T: git checkout -b <branchName>
   1. To create new branch in local folder
3. T: git branch
   1. Verify that you are in the newly created, local branch

**Make Changes to local branch folder**

1. F: add new files or change existing files
2. T: git status
   1. Check what changes have been made to the branch folder.
   2. Note: these changes are not committed and are considered as “untracked files present”

**Push local branch folder to GitHub**

1. T: git add .
2. T: git commit -m "<message>"
3. T: git push -u origin <branchName>

**Create pull request in GitHub**

1. GH: login and select the relevant repo
2. GH: create pull request
3. GH: check differences in file
   1. Add or change comments as necessary
4. GH: click on merge branches if accept changes
   1. At this point, the master has changed and everyone should git pull so everyone has the most recent master file copied in their local folder

**Deleting branch in local folder**

1. T: git checkout master
   1. Switches you from branch to master
2. T: git pull
   1. Updates copy of your existing, local master folder so the changes pushed and merged are reflected
3. T: git branch -d <branchName>
4. T: git branch
   1. To confirm that the branch created is deleted

Note: creating a branch in terminal does not translate into a physical folder in Finder. Switching between branches would yield different files. Always check via git branch in order to determine active branch. Use git status to see the changes