Dylan Lawson

Willem Bosman, Spencer Tingley

ILP Training Group

13th November 2019

GitHub Notes

**Adding Existing Project to GitHub:**

* Adding a local project to GitHub with the terminal: <https://help.github.com/en/articles/adding-an-existing-project-to-github-using-the-command-line>
* If “git push origin master” does not work like in the article, use “git push -u origin master” shown in the picture below

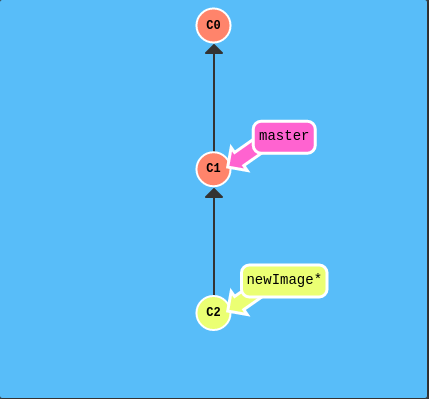


**Grabbing a Project from GitHub:**

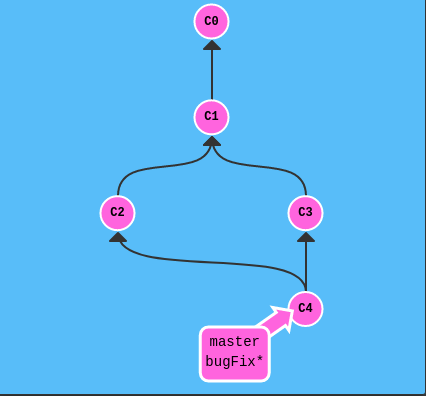
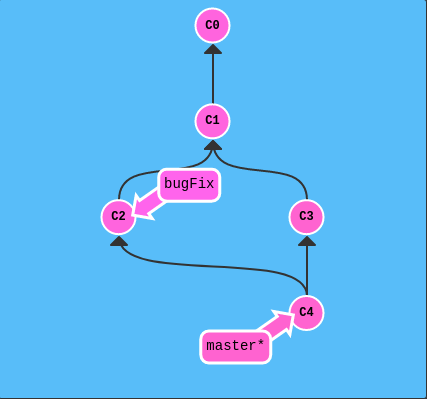
* 1. Go to the Directory for where you want the clone to be and clone the workspace → git clone
  2. (Optional) Open your IDE and make your changes → code .
  3. Once you’re done making changes → git add .
  4. Commit the project and record a message → git commit -m (“Write your comment here”)
  5. Push the new changes → git push
* If someone makes a change to the repo besides you, use this command to update yourself to the newest version
  1. git pull
* If you want to check the status of the repo, use this command:
  1. git status

**Branching off the Master Branch -** <https://learngitbranching.js.org/>

1. At times, you want to create a branch so that you don’t interfere with the master branch → **git branch <newImage>**
2. Once created, you’ll still be on the master branch. To switch over to the newly created branch → **git checkout <newImage>**
3. Now that we’re on newImage, when you commit, the newImage will be committed instead of the master branch → **git commit**

****

1. Once you’re all finished working with the newImage branch, you can merge it back into the master branch and all of the work done in newImage will now be apart of the master branch → **git checkout newImage; (Select the branch first, then merge) → git merge master**



git checkout bugFix; git merge master

**Manually Adding Files to GitHub Repo:**

* + 1. Go to the directory you want the repo to be in (using cd)
    2. Used **git pull** to get the most recent version of the project
    3. Manually add the file you want in the repo folder
    4. Saved the Stitch\_Documentation file as a .doc file
    5. Removed the Stitch\_Documentation.odt file using sudo rm Stitch\_Documentation.odt
    6. **git add .**
    7. **Git commit -m** “Added The Documentation that I worked so hard on”
    8. **git push**

**Miscellaneous/Troubleshooting Tips:**

**Commands:**

* **git pull** <remote> <branch> → **git pull** “<https://github.com/igotmail/Test.git>” “master”
* **git log –** Will show your remote and local branches created by you
* **git branch -av –** Will show you what branches exist in your repo both locally and remotely

**Situations:**

* Local repos will keep a .git file when you use “git init.” If you want to make a fresh repo out of the folders you initialize, you can delete the .git init file and then use “git init” again in the terminal: <https://stackoverflow.com/questions/1213430/how-to-fully-delete-a-git-repository-created-with-init>
* You cannot push an empty folder. The folder will need files in order to be pushed to the repo: <https://dev.to/arisa_dev/git-github-git-initialize-error-error-src-refspec-master-does-not-match-any-2lcj>
* You may need to use **git pull –allow-unrelated-histories** if you run into the “Refusing to merge unrelated histories” error: <https://github.community/t5/How-to-use-Git-and-GitHub/How-to-deal-with-quot-refusing-to-merge-unrelated-histories-quot/m-p/16305#M5000>
* Mark down Cheat Sheet: <https://github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet>
* What to add in your .gitignore file: <https://zellwk.com/blog/gitignore/>
* If all else fails, first clone the repo from GitHub into a new folder: **mkdir [folderName]**. Then type in **git init** to allow that folder to use git commands.
  + Lastly, use **git pull [repoName]** ex. git pull <https://github.com/igotmail/Udemy_MEAN_Course.git>. This folder should now have your remote repo you created in that folder.
  + Second, manually copy what’s inside of the folder you originally wanted to push to GitHub in the cloned repo. Use **git add . -> git commit -m “Type your message” -> git push -u origin master**. This basically is a workaround when GitHub is picky and won’t let you push you work to it