Lab 1
Intro to Computer Science (420-121-VA)
Perry R. James, 30 August 2020

# Version Control with Git & GitHub

Git is a distributed version-control system that allows multiple users to easily work together. It also allows for various branches of changes to be maintained and merged to facilitate exploratory development. GitHub is a service that hosts repositories in the cloud.

## Install git command-line tool

If you do not have git on your PC, download and install it from the link below. If have a Mac or Linux, it is probably already installed.

https://git-scm.com/download/win

## Text files

Most of the files directly created by computer programmers are text files. In addition to source code, data files are often used for configuration. A couple of common formats are CSV (comma separated values) and JSON (JavaScript Object Notation). These files can be stored in a git repository to track changes over time and to centralize backups. Binary files that are generated by computers are not usually included in the repository. To facilitate the distributed nature of large software projects, team of developers need a way to be able to work on the same set of files and easily reconcile their changes.

## Documentation

Spend about 20 minutes reading the web page “Git and Github: A Beginner’s Guide for Complete Newbies” https://www.elegantthemes.com/blog/resources/git-and-github-a-beginners-guide-for-complete-newbies

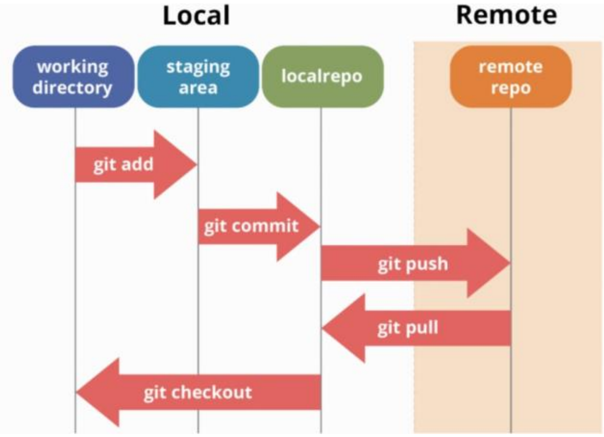


*Fig. 1dev.to/mollynem/git-github--workflow-fundamentals-5496*

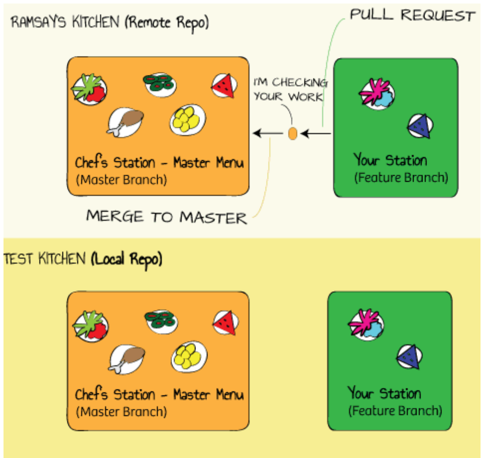


*Fig. 2 bloggytoons.com/posts/2013/10/10/git-kitchen-wchef-ramsay*

## Practice

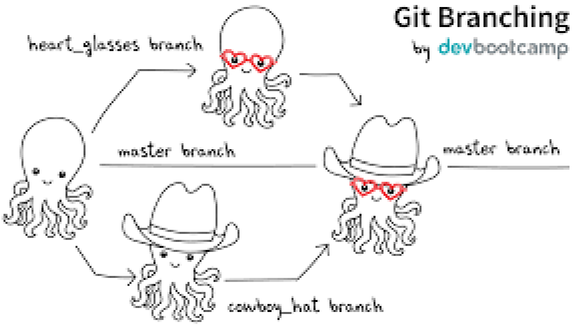
1. Create an account for yourself in a public git repository service (e.g., github.com or gitlab.org)
2. Create a new repository and initialize it with a README file. If you have a favorite license, you can add it too.
3. Using the command line, clone your repository to a local folder. First create a local directory for your git repositories. Then copy your repository's link from the web interface. Finally, clone it to make a local copy with `git clone <your repo's URL>`
4. Create a new text file in your local copy's directory. It can have any contents you like.
5. Add the new file to git's staging area (`git add <filename>`) and commit your changes (`git commit -m "<your commit message here>"`) – or use the shortcut to add (all files) and commit in one step (`git commit -am "<your commit message here>"`)
6. After you push your changes to the remote repository (`git push`), refresh the browser and notice the updates online. Add another file to your project and push it.
7. To simulate working on another computer, go to another directory and clone your repository. Once you have a separate version, make some changes to it, then commit and push them.
8. Because of the possibility of your having made changes elsewhere, or someone else having made changes to the files, it is important to synchronize your local folder with the remote repository to make sure you have the most recent code (`git pull`). Do this before doing any new work.
9. Once you have made a small set of changes locally, you should commit them to the server. Do this by adding files to the staging area, committing your changes, and pushing them to the server often. It is easy to make a change that breaks something, so it is good to be able to quickly go back to a recent working version.
10. Try editing the two copies of the same file, pushing them, and then resolve any conflicts.

THIS IS GIT. IT TRACKS COLLABORATIVE WORK ON PROJECTS THROUGH A BEAUTIFUL DISTRIBUTED GRAPH THEORY TREE MODEL.
COOL. HOW DO WE USE IT?
NO IDEA. JUST MEMORIZE THESE SHELL COMMANDS AND TYPE THEM TO SYNC UP. IF YOU GET ERRORS, SAVE YOUR WORK ELSEWHERE, DELETE THE PROJECT, AND DOWNLOAD A FRESH COPY.

| | COMMENT | DATE |
|---|---|---|
| ○ | CREATED MAIN LOOP & TIMING CONTROL | 14 HOURS AGO |
| ○ | ENABLED CONFIG FILE PARSING | 9 HOURS AGO |
| ○ | MISC BUGFIXES | 5 HOURS AGO |
| ○ | CODE ADDITIONS/EDITS | 4 HOURS AGO |
| ○ | MORE CODE | 4 HOURS AGO |
| ○ | HERE HAVE CODE | 4 HOURS AGO |
| ○ | AAAAAAAA | 3 HOURS AGO |
| ○ | ADKFJSLKDFJSDKLFJ | 3 HOURS AGO |
| ○ | MY HANDS ARE TYPING WORDS | 2 HOURS AGO |
| ○ | HAAAAAAAAANDS | 2 HOURS AGO |

AS A PROJECT DRAGS ON, MY GIT COMMIT
M

Dear git,
Here is the thing I just did.
And this is why I did the thing.
♥ me



Based on a lab by Tassia Camões-Araújo that is licensed under CC BY-NC-SA 4.0 (https://creativecommons.org/licenses/by-nc-sa/4.0)