Github Quick Start

Setup

1. Signup for free Github Account
2. Install Git from http://git-scm.com/downloads
3. Setup Git at this website: https://help.github.com/articles/set-up-git/#platform-mac

   NOTE: Select your Operating System at top of page under title, “Set Up Git”

   Skip step 1 if you already installed Git
4. Optional: Setup SSH Keys: https://help.github.com/articles/generating-ssh-keys/

Basic Terminal Commands

1. cd PATH TO DIRECTORY - change directory (e.g. cd ./Documents/Projects/git_tut)

   cd ./ - start from current directory

   cd ../ - go back a directory
2. ls - list content in directory
3. mkdir - make directory
4. rm Filename - delete a file
5. rm -rf Directory - delete a directory as well as it sub-directories

Top Git Commands

1. **git init** - initialize current directory as a git directory. First command to run in a directory
2. **git status** - check the status of the changes you made
3. **git log** - View commit history
4. **git add FILENAME HERE** (note: **git add .** to add all files that were edited)
5. **git commit -m “Put Any Message Here”** - Save changes
6. **git clone REPO URL HERE** - Copy an existing repo from the internet to local computer
7. **git remote add origin REPO URL HERE** - link local directory to a repo online
8. **git remote -v** - Check which online repo this directory is linked to

9. **git remote rm origin REPO URL HERE** - Remove link to online repo
10. **git push origin master** - Push changes to online repo at the master(default) branch
11. **git pull origin master** - Pull from online repo & merge changes to local directory
12. **git branch** - View list of all branches
13. **git branch NAME_OF_NEW_BRANCH** - Create a new branch

10-Step Walkthrough to First Github Project

1. Create a new public Repo (Do not click *Initialize this repository with a README*)

   - Fill out Repository Name - WebMonkeysGit

   - Select Public

   - Click "Create repository" button

2. Open Terminal and navigate to Documents directory

   - **cd Documents**

   - If that didn't work, use the "ls" command to see where you are in your file system, and try to navigate to your Documents or My Documents directory

3. Make a directory called WebMonkeysGit and navigate to it

   - **mkdir WebMonkeysGit**

   - **cd WebMonkeysGit**

4. Initialize this as a a Git repository

   - **git init**

5. Link this directory to your Repo on Github

   - **git remote add origin https://github.com/YOUR_USER_NAME/WebMonkeysGit.git**

   - View the remote to see if it was added

     git remote -v

6. Open a code editor and create a README.md file and another random file

   - Open a new file

   - Add "This is a Web Monkeys Git Tutorial"

   - Save as "README.md"

   - Open a new file

- Add “Hello World”
- Save as index.html or whatever you’d like

7. In Terminal, use Git to see the files you just created and/or modify
   - **git status**
8. Add those changes to prepare to be saved
   - **git add README.md**
   - **git add index.html** (or the name of whatever random file you made)
   - **NOTE:** If you’d rather add all of the file you make or modify, run:

     git add .
9. Commit (Save) new files
   - **git commit -m “I just made two new files”**
10. Push to your repo on github.com and visit(or refresh) your repo webpage to see the new files you made
    - **git push origin master**

Other Important Git Concepts

1. **Git Branches** - Branches are used to separate development tasks, and help manage different versions of your project during development. For instance, if you are working on a web app with a team of 4, each person is working on a different feature or fixing a different bug. In order to maintain integrity of the working version of the app, each person creates a new branch for each task.

   The “master” branch is your main branch. You should always want to make sure the project version on this branch is working because this is the branch that is used for production and everyone collaborates from.

   - To see existing branches, run:

     git branch
   - To create a new branch run:

     git branch NEW_BRANCH_NAME (no spaces allowed)

- To switch between branches(ALWAYS commit before switching:

```
git checkout BRANCH_NAME
```

2. **Merge Conflicts** - When working with your local directory and a repository on github.com, you're trying to manage several copies of one project. This may cause you to run into some problems where the same file has been edited by two different users, and Git fails to allow you to pull and/or merge from a repo online, or merge the same files on your local machine from different branches. The terminal may say "Resolve merge conflicts before committing". Navigate to the file(s) you are trying to merge, and notice special characters were added to show the changes made by one user A and user B. Choose the version you'd like, remove the other user's changes as well as the special characters Git added. Try to commit again.

3. **Pull Requests** - A pull request is when someone submits a branch to be merged to another branch on Github.com. When working on a project with several people, and you are the owner of the repo, you'll need to prevent bad code from getting merge into the master branch, and prevent merge conflicts. You view the changes made by users, and accept or reject a pull request.