Using Git and Collaborating with others on GitHub

Author: Alejandro Martinez

Introduction

       This guide is meant for beginners and will help you learn the basics of how to use git and how to work with and add to a GitHub repository. Feel free to look over the Git Commands and Git Vocabulary files located in <u>How to Use Git</u> repository to get a better understanding of the commands we are going to use here.

       To use this guide follow the numbered instructions. Following the guide will lead you to making your first contribution to a GitHub repository. While using this guide it will be best to type out all the commands instead of just copy and pasting so you get as much hands on typing as possible. Any instructions that are preceded by a ** are extra credit but will be very beneficial.

       If the commands are confusing please refer to appendix A to see how to read them.

Instructions

1. ## Create a Github Account

   -Go to https://GitHub.com and create your free GitHub account

2. ## Install Git on Your Computer

   -Windows/Linux/MacOS go to https://git-scm.com/downloads
   -If the link above does not work, google git then click on the link with the url
   http://git-scm.com
   -For Apple computers it will be significantly more complicated but i suggest
   following the instructions that include installing homebrew.

3. ## Create a Folder/Directory

   -Create a Folder/Directory in your user directory called GitRepos.

   **Create the Folder/Directory using the Command Line/Terminal

Command to Enter	Description on What Command Does
cd	:Using this command by itself changes current directory to your default user directory
pwd	:Prints the directory you are working in
ls -l	:Shows you all the items in your current directory the -l option list them neatly
mkdir GitRepos	:Makes the GitReops folder/directory in that directory
ls -l	:Compare this list to the previous one your GitRepos folder/directory should be here now
cd ./GitRepos	:moves you into the folder/directory you just created
pwd	:Prints out what directory your in, it should have a path and end with /GitRepos

## 4. Fork - Creating a Copy of a GitHub Repository onto your GitHub

-On the Github page <u>How-to-Use-Git</u> you go to the top right of the page and select *Fork*, this creates a copy of the Repository onto your account. This is done if you want to contribute to a project and don't have permission to edit.
-Reference the photo below to see where the fork button is after you click on the link.
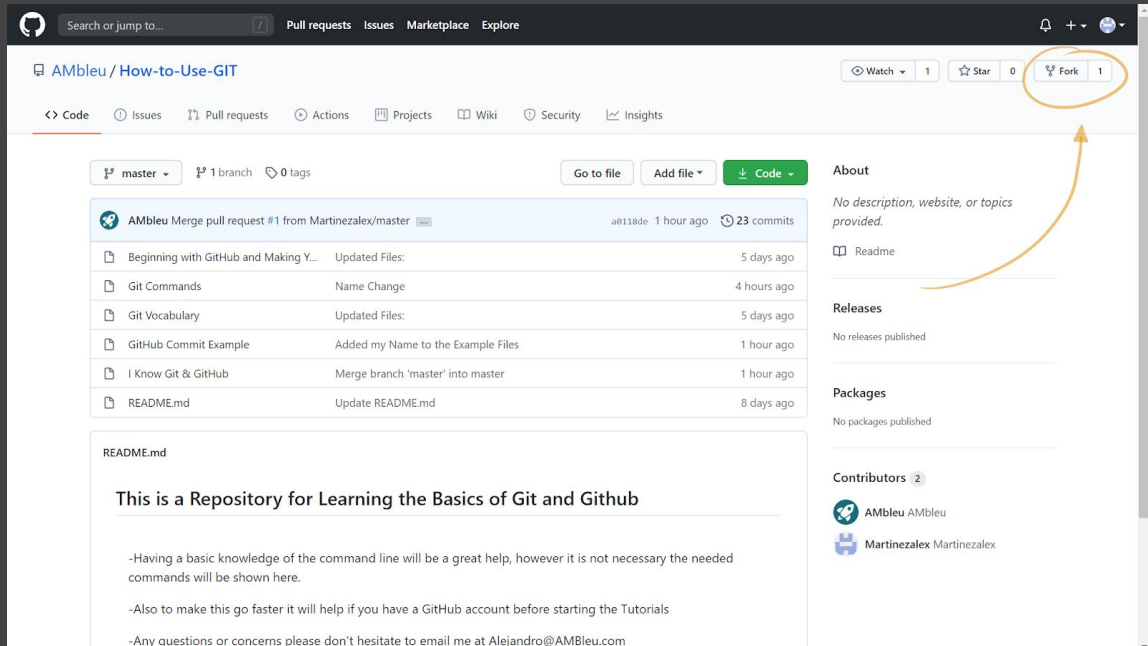
5. Clone - Creating a Local Copy of your GitHub Repository

-Go to your github profile and click on the "How-to-Use-Git" Repository click on the green code button and copy the link shown.
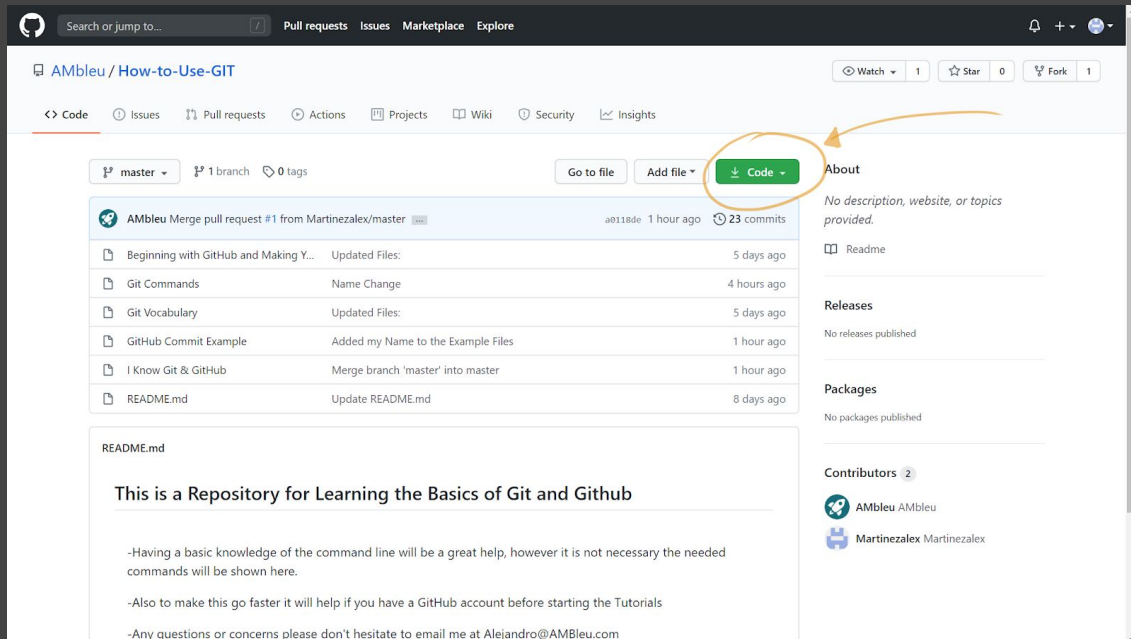
-Open Terminal, if not already in it, make sure you are in your GitRepos directory using the pwd command. If not, use the cd command to navigate to the GitRepos directory. Now use the copied link in the following command.

Command to Enter	Description on What Command Does
git clone -Link copied from Github-	:Paste the link you copied from GitHub into this command
cd How-to-Use-Git	:This puts you in the repository you just cloned
ls -l	:Allows you to see all the files previously on GitHub now on your local machine

6. Setting Up Identity to Log your Changes Locally

-In order to be able to use Git and store changes you have to set your name and email so Git can track who is making changes.

-To set these you use the followings commands:

Command to Enter	Description on What Command Does
git config user.name "Your Name"	:Enter the name you want to log your changes with.
git config user.email "Your Email"	:Enter the email you used for your Github account.
git config -l	:Shows you the a list of global fields

7. Creating a Branch and Switching to It

-Now that you have a local copy of the repository and set up your identity we can start editing the files. To do this we will make a branch and edit the branch.
-Even though edits to the master branch made in your copy will not edit the master branch in the original repository, when we request to merge into it later it will so for now we will make our edits in a new branch and later merge that branch into the original repository.

Command to Enter	Description of What Command Does
git branch	:Shows you all the branches. The asterisk shows what branch you are currently in.
git branch Your First and Last Name	:This creates a branch with your name as the branch name
git branch	:If the list of branches is short enough you should see your branch added here.
git checkout Your First and Last Name	:This lets you switch from working in the master branch to your unique branch
git branch	:Now you should see the list again with the asterisk next to the branch with your name on it.
git push --set-upstream origin Your First and Last Name	:Since we created the branch locally we have to use this to create the branch in GitHub

8. Editing a File in Your Local Repository

-Now we are going to start editing the files
-Navigate to the file "GitHub Commit Example" in your file explorer and edit the file save and exit.
**Edit the file called "GitHub Commit Example" on the command line using the following command:

Command to Enter	Description on What Command Does
nano 'GitHub Commit Example'	:This opens up a command line text editor, now put your name in the file like instructed.
(press Crtl + x)	:This exits the command and asks if you want to save your changes.
(press y)	:The edits have been made to the file "GitHub Commit Example" it is time to get ready to commit the changes. Committing makes a record that changes have been made so alter in the future you can come back to those changes, it's like a checkpoint.ssing y tells the text editor that yes you do want to save those changes

9. Add - Adding Files to Your Staging Area

-The edits have been made to the file "GitHub Commit Example" it is time to stage your changes to get ready to commit. Use the following command to add files to the staging area:

Command to Enter	Description on What Command Does
git add 'GitHub Commit Example'	:Moves the file into the staging area
git status	:Shows you what files are in your staging area ready to be committed

10. Commit - Committing Changes to Your Local System

-Now that the file you want to change has been moved into the staging area it is ready to be committed. Committing makes a "checkpoint" of your changes, which can be accessed later. To commit use:

Command to Enter	Description on What Command Does
git commit -m "Enter Your Message Here"	:This commits the change and makes a log of the change, putting in a substantial message is important here because the message will be used to find the changes at a future date.
git status	:Shows that there is nothing to be committed
git log	:Shows all the commits you have made.

11. Push - Pushing Files to GitHub

-Now you are ready to push files onto your repository on GitHub. This moves all the changes you made locally onto Github. To do this do the following:

Command to Enter	Description on What Command Does
git push	:This pushes all your changes onto GitHub. If this is your first time doing this it will ask you for your GitHub username and Password.

-After you push these changes to GitHub go onto your GitHub page to check if the files have been edited

12. Extra Practice

-To get extra practice before finishing the tutorial redo the steps 7 - 10 on the file called "I Know Git and GitHub". Goodluck and if you get stuck for whatever reason just start over or go to the common errors file and check if that helps, if not contact me at alejandro@ambleu.com

13.   Making a Pull Request to Original Repository

-Now you have uploaded your changes to GitHub. However where you uploaded them to was a copy of the original How-to-Use-Git repository. In order to contribute to projects you must now make a pull request which asks the owner of the repository to look at the changes you made and add them to the original. They have the option to either accept or reject them but the whole point of Git is contributing so whether your change gets accepted or rejected is ok as long as you contribute.
-To make this pull request follow the pictures and instructions below.
-Once a few changes have been made and committed to GitHub, the popup will appear and you can click on Pull Request to begin the process of contributing to the original repository.
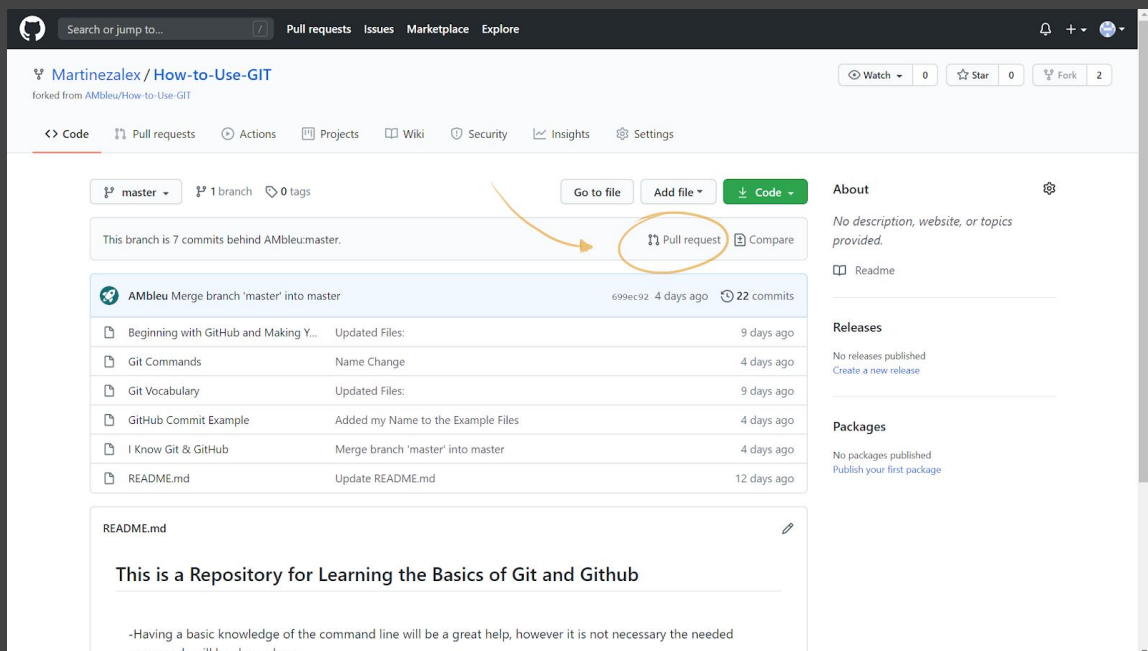

-For the following picture I numbered 3 main items to focus on.
1) This is where it is going to be merged to, it should say AMbleu/How-to-Use-Git.
2) This is what is going to be merged, this should be <Your GitHub Username>/How-to-Use-Git
3) Once you have confirmed the information above you can press the green "Create pull request" button.
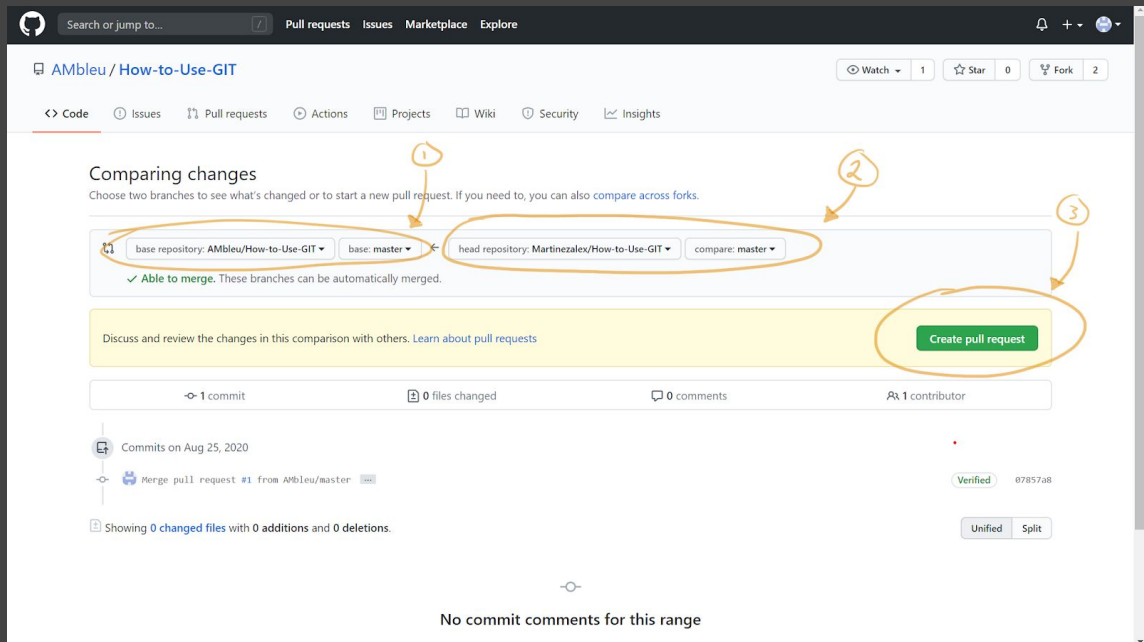
-Similar to a commit you must create a message for your pull request. As before there are numbered items to focus on.

4) That is the title of the pull request, you can title this First Github Pull Request, Finished GitHub Startup Guide, or Simply your name
5) A message explaining what is in your pull request, how it will benefit the code, and/or a quick explanation why you made the pull request.
6) Once those fields have been filled out then once again click on the green "Create pull request" button and then your job is done all that is left to do is wait until you get a response to the pull request.
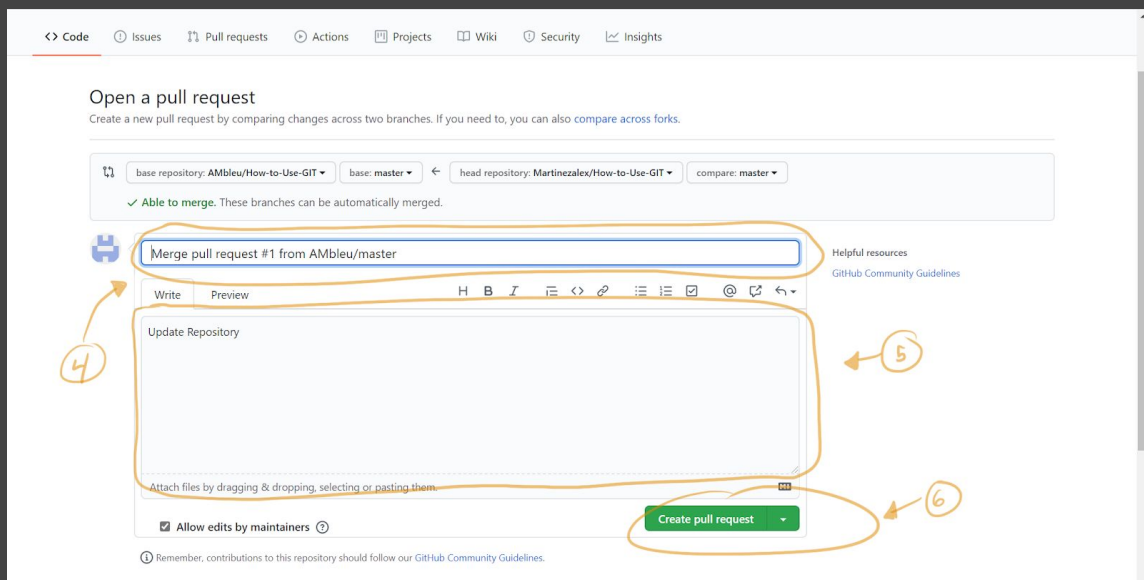
Appendix A

Text in green should be entered exactly as it is typed, text in white is a command to the user to enter that type of information

Ex.

-git config user.name "Your Name"

-In this case the git config user.name" is typed in the terminal and then the white text is replaced with your name and then the last " is added. If your name was John Doe then the command in the terminal would look like this: git config user.name "John Doe"