**How to use Git and the Terminal**

*Ebonie Gadson*

*12 August 2018*

**Table of Contents**

1. Getting Started 1.1. Github

1.1.1. Creating a Github account 1.1.2. Request a discount 1.1.3. Creating a Project in Github 2. How to begin using Git

2.1. Git Desktop

2.1.1. What is Git Desktop? 2.1.2. How to download and install Git Desktop 2.1.3. How to clone a repository into Git Desktop 2.1.4. Comitting our changes 2.1.5. Pushing our changes 2.1.6. Pulling from the repository 2.2. Git Bash

2.2.1. What is Git Bash? 2.2.2. Download 2.2.3. Installation 2.3. Terminal

2.3.1. macOS 2.3.2. Linux 3. On Git commands in general: 4. Git Status

4.1. What is it for? 4.2. Usage 4.3. Git status - Add 4.4. Git status - Remove 5. Git Commit

5.1. What is it for? 6. Git Push

6.1. What is it for? 6.2. Git Pull 6.3. What is it for?

1

**1. Getting Started**

***1.1. Github***

**1.1.1. Creating a Github account**

*Figure 1.1.1: In order to have access to an unlimited amount of private repositories, sign up without your MCs student email address (MYMCID@montgomerycollege.edu).*

2

**1.1.2. Request a discount**

*Figure 1.1.2: After signing up for github go to https://education.github.com to request a free 2 year discount. Select “GitHub Student Developer Pack”*

*Figure 1.1.3: Choose: "Get your pack"*

3

*Figure 1.1.4: Select: "Yes, I'm a student"*

4

*Figure 1.1.5: Fill out the information above and upload a photo of your Montgomery College ID. Then, submit your request.*

5

*Figure 1.1.6: Check college email to see when you have leveled up to student pack. You should receive a similar email to the one above typically in minutes or 3 to 5 business days. After submitting your request go to the billings tab on github and you should see your coupon.*

6

*Figure 1.1.8: Next, Select the "Billing" option*

*Figure 1.1.7: You can view your coupon by clicking on the down arrow next to your profile pic and selecting the “Settings” option.*

7

*Figure 1.1.9: After, check to see if your plan includes unlimited private repositories and that you have an active coupon for $7.00.*

**1.1.3. Creating a Project in Github**

*Figure 1.1.10: When we are logged into GitHub.com, we can store the project by first clicking on “Start a project.”*

1. Creating a repository

8

*Figure 1.1.11: Next, we name the repository, select either public or private, and check the initialize README box. Then, click “Create repository.”*

9

*Figure 1.1.12: We can click on “Clone or download” to open our repository locally.*

*Figure 1.1.13: This will give us an HTTPS link we can use to clone our repository anywhere.*

**2. How to begin using Git**

***2.1. Git Desktop***

**2.1.1. What is Git Desktop?**

Git Desktop is a graphical environment for using the git version-control software. It allows us to do all operations that can be done from Git Bash or the Terminal through a graphical interface.

10

**2.1.2. How to download and install Git Desktop**

*Figure 2.1.1: We can download GitHub desktop to manage our workflow for Windows and macOS. https://desktop.github.com/ Choose “Download for Windows (64bit)” or “Download for macOS.”*

1. Installation

11

*Figure 2.1.2: (WINDOWS) Then, click run in order to start the installation of the program.*

*Figure 2.1.3: (macOS) Click “Open” when a notification pops up asking to download GitHub Desktop.*

*Figure 2.1.4: (macOS) Finally, open the application where it has been downloaded (in this case in the Desktop) in order to start the installation process.*

12

*Figure 2.1.5: Once we run GitHub desktop, we can login to our account by choosing “Sign into GitHub.com.*

*Figure 2.1.6: Once we successfully sign in using our credentials, we click "Continue" and then "Finish".*

13

**2.1.3. How to clone a repository into Git Desktop**

*Figure 2.1.7: To download our repository in GitHub desktop, we select the “Add a local repository” option. Note: If we have not created a repository on GitHub.com, we should select “Create new repository” instead.*

*Figure 2.1.8: Next, we select a repository from GitHub.com, choose where to store it locally, and click “Clone.”*

14

**2.1.4. Comitting our changes**

*Figure 2.1.9: Add a summary of what was changed (a message). You can add a description but it is not necessary. Click “Commit to [branch]”.*

**2.1.5. Pushing our changes**

15

*Figure 2.1.10: Select “Push Origin” to push changes to GitHub.com.*

**2.1.6. Pulling from the repository**

*Figure 2.1.11:*

***2.2. Git Bash***

**2.2.1. What is Git Bash?**

Git bash is a software that allows us to use the *git* software from a terminal emulator using the Windows OS. Most developers usually interact with the git version control system this way, either by using Git Bash or another terminal emulator (in macOS or Linux).

**2.2.2. Download**

You can download Git Bash here: https://git-scm.com/download/win

**2.2.3. Installation**

Instructions for installing Git Bash:

16

*Figure 2.2.1: Make sure all the boxes for “Windows Explorer integration”, “Git LFS”, “Associate .git configuration”, and “Associate .s files are checked.” Click “Next”.*

17

*Figure 2.2.2: After choose a local path where Git will be installed. Click “Next”.*

*Figure 2.2.3: Save the name for the program shortcut in the start menu as “Git Bash” and click “Next”.*

18

*Figure 2.2.5: After reading the “GNU General Public License”, click “Next”.*

*Figure 2.2.4: Choose Git’s default editor as “Use Vim (the ubiquitous text editor) as Git’s default editor” and click “Next”.*

19

*Figure 2.2.6: Make sure the “Use Git from Windows Command Prompt” before clicking “Next”.*

*Figure 2.2.7: Choose “Use Open SSL library” and click “Next”.*

20

*Figure 2.2.8: Make sure “Checkout Window-style, commit Unix-style line” is the selected option before clicking “Next”.*

*Figure 2.2.9: Select “Use MinTTY (default terminal of MSYS2)” as the GitBash terminal, then click “Next”.*

21

*Figure 2.2.10: Make sure all but the “Enable symbolic links” box are checked before clicking “Install”.*

***2.3. Terminal***

**2.3.1. macOS**

1. Download

Download *git* from the official website here: https://git-scm.com/download/mac

2. Installation

Open the downloaded files and open the *package* (.pkg) file. Then, click on "Install".

**2.3.2. Linux**

Installing *git* on different Linux-based distributions

1. Ubuntu

Open a terminal and type in the following command:

sudo apt install git-all

2. Fedora

sudo dnf install git-all

22

**3. On Git commands in general:**

The instructions shown below are the same for *all* methods shown (except for *Git Desktop*). This means, that whether you are using *Git Bash* or a terminal in either *macOS* or *Linux*, the commands will be the same.

**4. Git Status**

***4.1. What is it for?***

The *git status* shows us the status of our files within our working project. This means that it will show whether changes have been made from the previous version of it.

***4.2. Usage***

In order to use it, we execute the following command:

git status

This will show something similar to the following:

Untracked files:

(use git add <file>... to include in what will be committed)

file\_name.txt

***4.3. Git status - Add***

The add command will allows us to *add* files in order to later have them change the latest version of our project (essentially, adding our new changes). This can be done with the following command:

git add <file>

Once doing so, the file will be added. The status will know look like this:

Changes to be commited:

(use git reset HEAD <file>..." to unstage)

new file: hello\_there.txt

***4.4. Git status - Remove***

The remove command has the opposite effect to the add command. This will revoke changes (not add them) once these want to be made.

git remove <file>

**5. Git Commit**

***5.1. What is it for?***

The *git commit* command allows us to tell the git software that we want the added changes to be later added into the latest version of our project. The way we do this is by *comitting* our changes, and

23

adding a message explaining what changes we have made.

git commit -m "Our message goes here. We briefly explain what it is we have changed."

Example case:

$ git commit -m "Added a welcome message to the website's homepage" [master 74a4776] made changes

**6. Git Push**

***6.1. What is it for?***

The *git push* command is the final step towards making an effective change in our project. It tells git to grab all the different changes we have made and update the repository stored in github.

Usage:

git push

This will prompt us for our username and password for github. This is a security measure, in order to make sure only a valid user can edit the project.

***6.2. Git Pull***

***6.3. What is it for?***

The *git pull* command does the opposite of the *git push* command. It gets all the changes from our project stored in github in order to have the latest changes to the project.

This means that we should either get a confirmation message stating that the local (our) version of the project has been updated, or that it already was upto date.

24