**GitHub Exercise**

**Part 3:**

Answer the following questions.

What is GitHub? When was it created? Why? By who? What similar platforms exist? Why would you use such a platform? (Answer between 5 and 10 lines)

**Answer:**

GitHub is a code hosting platform for version control and collaboration. It helps you work on projects together from anywhere.

It was created on February 8, 2008.

It lets you and others work together on projects from anywhere.

Created by Linus Torvalds.

Similar Platforms: Bitbucket, CodePlex, StackOverflow, Google Developers, SalesForce.com

I would use such a platform to co-ordinate with my teammates, keep track of work, and manage code hosting. It would be a great in today’s world where people always keep travelling for work. Hence working on a platform where you can collaborate easily even on the go can help working easier for everyone on a team.

Plus some extra technical things from Wikipedia.com:

“It offers all of the distributed version control and source code management (SCM) functionality of Git as well as adding its own features. It provides access control and several collaboration features such as bug tracking, feature requests, task management, and wikis for every project. GitHub offers both plans for private repositories, and free accounts which are commonly used to host open-source software projects. As per Wikipedia, as of April 2016, GitHub reports having more than 14 million users and more than 35 million repositories, making it the largest host of source code in the world.”

More specifics can be found at: https://www.linux.com/blog/10-years-git-interview-git-creator-linus-torvalds

**Part 4:**

**Git Tutorial:**

Press enter to submit commands

> git init

Initialized empty Git repository in /.git/

Success!

$ git status

# On branch master  
#  
# Initial commit  
#  
nothing to commit (create/copy files and use "git add" to track)

Success!

$ git status

# On branch master  
#  
# Initial commit  
#  
# Untracked files:  
# (use "git add <file>..." to include in what will be committed)  
#  
# octocat.txt  
nothing added to commit but untracked files present (use "git add" to track)

Success!

$ git add octocat.txt

Nice job, you've added octocat.txt to the Staging Area

$ git status

# On branch master  
#  
# Initial commit  
#  
# Changes to be committed:  
# (use "git rm --cached <file>..." to unstage)  
#  
# new file: octocat.txt  
#

Success!

$ git commit -m "Add cute octocat story"

[master (root-commit) 20b5ccd] Add cute octocat story  
1 file changed, 1 insertion(+)  
create mode 100644 octocat.txt

Success!

$ git add '\*.txt'

Success!

$ git commit -m 'Add all the octobat txt files'

[master 0bcf5fc] Add all the octobat txt files  
4 files changed, 4 insertions(+)  
create mode 100644 blue\_octocat.txt  
create mode 100644 octofamily/baby\_octocat.txt  
create mode 100644 octofamily/momma\_octocat.txt  
create mode 100644 red\_octocat.txt

Success!

$ git log

commit 3852b4db1634463d0bb4d267edb7b3f9cd02ace1  
Author: Try Git <try\_git@github.com>  
Date: Sat Oct 10 08:30:00 2020 -0500  
  
Add all the octocat txt files  
  
commit b652edfd888cd3d5e7fcb857d0dabc5a0fcb5e28  
Author: Try Git <try\_git@github.com>  
Date: Sat Oct 10 08:30:00 2020 -0500  
  
Added cute octocat story

Success!

$ git remote add origin https://github.com/try-git/try\_git.git

Success!

git push -u origin master

Branch master set up to track remote branch master from origin.

Success!

$ git pull origin master

Updating 3852b4d..3e70b0f  
Fast-forward  
yellow\_octocat.txt | 1 +  
1 file changed, 1 insertion(+)  
create mode 100644 yellow\_octocat.txt

Success!

$ git diff HEAD

**diff --git a/octocat.txt b/octocat.txt  
index 7d8d808..e725ef6 100644  
--- a/octocat.txt  
+++ b/octocat.txt  
@@ -1 +1 @@  
-A Tale of Two Octocats  
+[mA Tale of Two Octocats and an Octodog**

Success!

$ git add octofamily/octodog.txt

Success!

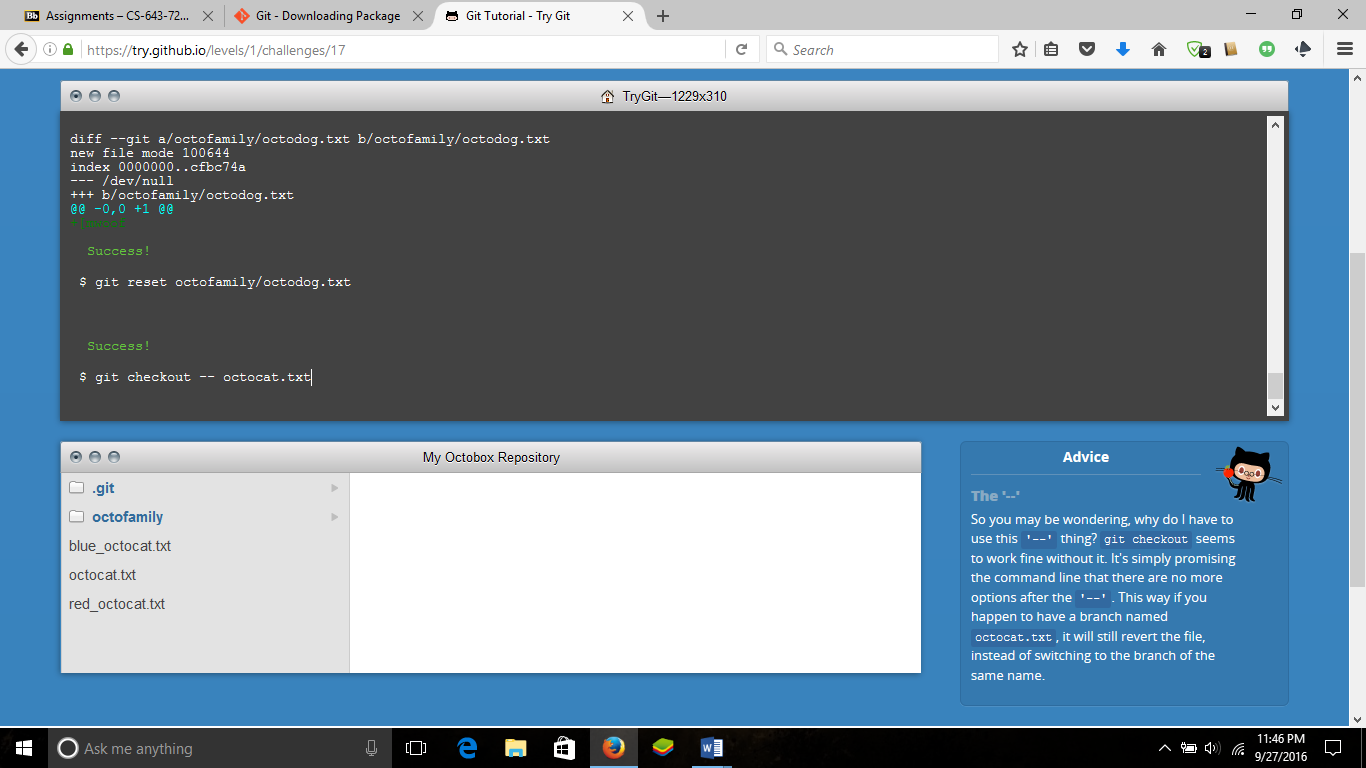
$ git diff --staged

**diff --git a/octofamily/octodog.txt b/octofamily/octodog.txt  
new file mode 100644  
index 0000000..cfbc74a  
--- /dev/null  
+++ b/octofamily/octodog.txt  
@@ -0,0 +1 @@  
+[mwoof**

Success!

$ git reset octofamily/octodog.txt

Success!



$ git checkout -- octocat.txt

Success!

$ git branch clean\_up

Success!

$ git checkout clean\_up

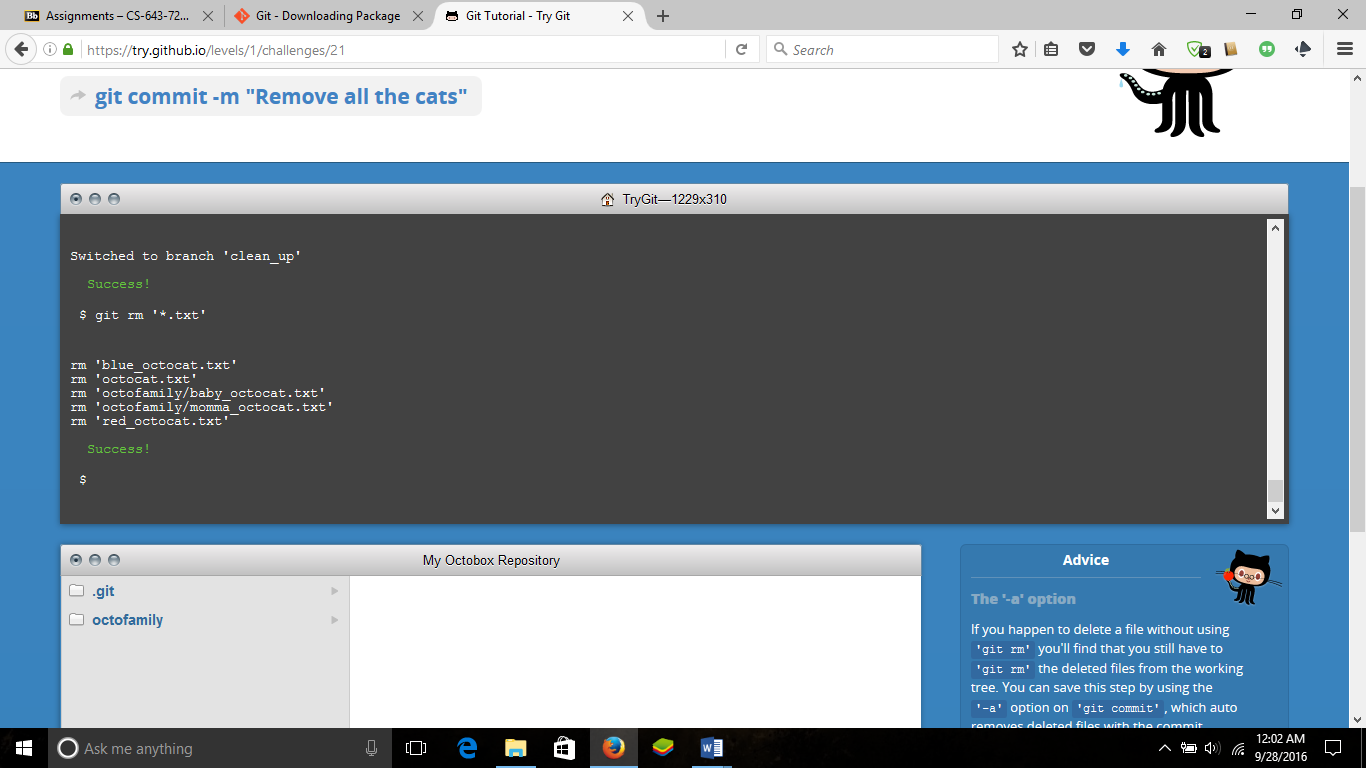
Switched to branch 'clean\_up'

Success!

$ git rm '\*.txt'

rm 'blue\_octocat.txt'  
rm 'octocat.txt'  
rm 'octofamily/baby\_octocat.txt'  
rm 'octofamily/momma\_octocat.txt'  
rm 'red\_octocat.txt'

Success!



$ git commit -m "Remove all the cats"

[clean\_up 63540fe] Remove all the cats  
5 files changed, 5 deletions(-)  
delete mode 100644 blue\_octocat.txt  
delete mode 100644 octocat.txt  
delete mode 100644 octofamily/baby\_octocat.txt  
delete mode 100644 octofamily/momma\_octocat.txt  
delete mode 100644 red\_octocat.txt

Success!

$ git checkout master

Switched to branch 'master'

Success!

$ git merge clean\_up

Updating 3852b4d..ec6888b  
Fast-forward  
blue\_octocat.txt | 1 -  
octocat.txt | 1 -  
octofamily/baby\_octocat.txt | 1 -  
octofamily/momma\_octocat.txt | 1 -  
red\_octocat.txt | 1 -  
5 files changed, 5 deletions(-)  
delete mode 100644 blue\_octocat.txt  
delete mode 100644 octocat.txt  
delete mode 100644 octofamily/baby\_octocat.txt  
delete mode 100644 octofamily/momma\_octocat.txt  
delete mode 100644 red\_octocat.txt

Success!

$ git branch -d clean\_up

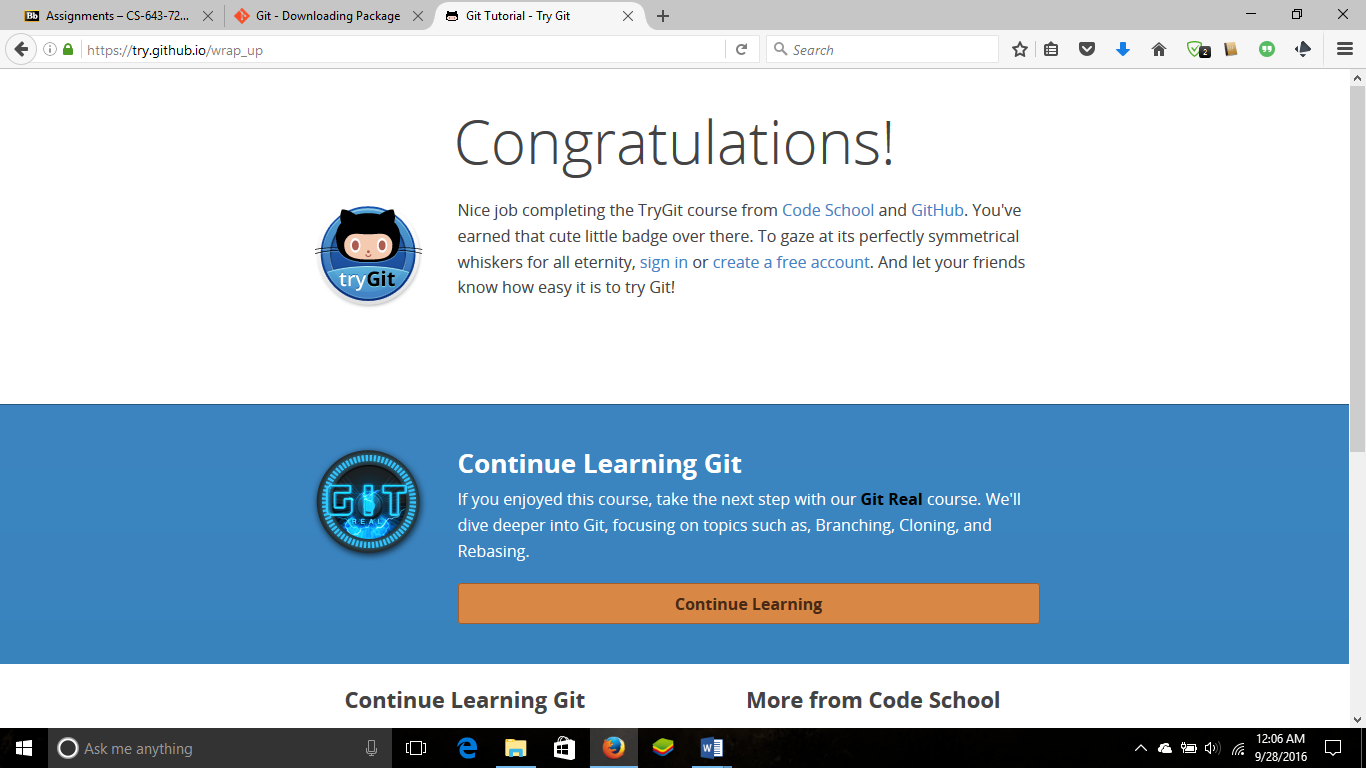
Deleted branch clean\_up (was ec6888b).

Success!

$ git push

To https://github.com/try-git/try\_git.git  
3e70b0f..e0c2dea master -> master

Success!



**Part 5:**

* Repository

A **repository** is something that is used to organize a single project. It can contain files, folders, images, videos, spreadsheets, and data sets, etc. basically everything that a project may need.

* Commit

This records any changes done to the repository.

* Push

Used to update remote references along with associated objects.

* Branch

A branch can exist within a reposirory and can be used to List, Create, or Delete branches. Mainly used for thread development.

* Fork

This isn’t actually a git concept. What fork means on github is, if people aren't happy with the way a project is going, they can take the source code and work on it themselves separate from the original developers.

* Merge

Merge actually Joins two or more development histories together.

* Clone

Clone is used to Clone a repository into a new directory.

* Pull

Pull function is used to Fetch from and integrate with another repository or a local branch.

* Pull request

Pull request generates a summary of changes that are pending.