What is GitHub? When was it created? Why? By who? What similar platforms exist? Why would you use such a platform?

**GitHub** is a web-based Git or Version Control Repository or Source Code Management (SCM) system which used for code sharing and collaboration. It provides access control and several collaboration features such as bug tracking, feature requests, task management, and wikis for every project.

GitHub was launched in April 2008 by Tom Preston-Werner, Chris Wanstrath, and PJ Hyett.

Similar Platforms like GitHub are listed below:

* Bitbucket
* Kiln/DevHub
* Beanstalk
* SourceForge
* Apache Allura
* AWS CodeCommit
* GitKraken
* GitLab
* Codeplane
* CodePlex

We might use these alternatives instead of github in the following scenarios:

* When team needs the security and privacy of an in-house system
* When you have a large team, and GitHub’s pricing doesn’t fit
* When you work in a version control language other than Git

Part 4:

To initialize a Git repository:

> **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 this message"**

[master (root-commit) 9a0422d] Add this message

1 file changed, 1 insertion(+)

create mode 100644 octocat.txt

Success!

**$ git add '\*.txt'**

Success!

**$ git status**

# On branch master

# Changes to be committed:

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

#

# new file: blue\_octocat.txt

# new file: octofamily/baby\_octocat.txt

# new file: octofamily/momma\_octocat.txt

# new file: red\_octocat.txt

#

Did not use git commit

**$ git commit -m 'Add all the octocat txt files'**

[master 3852b4d] Add all the octocat 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 branch**

clean\_up

\* master

Use 'git checkout' to switch to the 'clean\_up' branch

**$ 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 status**

# On branch clean\_up

# Changes to be committed:

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

#

# deleted: blue\_octocat.txt

# deleted: octocat.txt

# deleted: octofamily/baby\_octocat.txt

# deleted: octofamily/momma\_octocat.txt

# deleted: red\_octocat.txt

#

# Untracked files:

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

#

# octofamily/

Did not use git commit

**$ 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..63ce92b master -> master

Success!

**Define the Following terms in terms of Git**

**Repository**: A collection of refs together with an object database containing all objects which are reachable from the refs, possibly accompanied by meta data from one or more porcelains. A repository can share an object database with other repositories via alternates mechanism.

Or in Simple words:

A repository is a folder inside which you are going to store every piece of your code.

**Commit**: The "commit" operation enables you to record changes that were made to a file or directory in the Git history.

**Push:**  To start sharing our file changes with others, we have to push them to a remote repository using the "push" command. This will cause the remote repository to update and synchronize with our local repository.

**Pull:** Wedo a "pull" operation to pull the changes from the remote repository onto your local repository. When pull is executed, the latest revision history is downloaded from the remote repository and imported to your local repository.

**Branch:** Branch is essentially an independent line of development. We can take advantage of branch when working on new features or bug fixes as it helps to isolate your work from that of other team members.

**Clone:** Use the "clone" command to copy a remote repository. By default, the "clone" command would automatically set up a local master branch that tracks the remote master branch that was cloned from.

**Merge:** Join two or more development histories together.

**Fork:** Fork is a copy of a repository. Forking a repository allows you to freely experiment with changes without affecting the original project.

**Pull Request**: Pull requests let you tell others about changes you've pushed to a GitHub repository. Once a pull request is sent, interested parties can review the set of changes, discuss potential modifications, and even push follow-up commits if necessary.