**Tutorial on GitHub, Git and Waffle.io**

**Part 3**

* **What is GitHub?**

GitHub is a web-based git repository hosting service. It also permits to document a project and manage tasks and bug reports from anywhere. It is known as the social network of developers.

* **When was it created?**

Development of the GitHub platform began on 1 October 2007.But the site was finally launched on 10 April 2008

* **Why?**

I think GitHub was develop to connect a team working on a project and also others developers and to allow them to interact from anywhere.

* **By who?**

GitHub was created by Tom Preston-Werner, Chris Wanstrath and PJ Hyett.

* **What similar platforms exist?**

A similar platform to GitHub is Bitbucket.

* **Why would you use such a platform? (Answer between 5 and 10 lines)**

I would you use GitHub because it lowers the barriers to collaboration. It’s easy to offer suggested changes to others’ code through github. And in a team project You can look at what others are working on and easily peruse their code and make suggestions or changes.

**Part4 : Tutoriel**

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 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 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 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..401a4d4 master -> master

Success!

**Part 5**

Define the following terms in the context of Git (2 lines maximum):

* **Repository :**

A repository is simply a place where the history of the work is stored. It often live a copy of the most recent state of the files you're working on.

* **Commit :**

To commit is to store staged changes.

* **Push :**

The push command tells Git where to put our commits.

* **Branch :**

Branchs are forks within your own repository. It will have an ancestor commit in your repository, and will diverge from that commit with your changes. Branches let you work on multiple disparate features at once.

* **Fork :**

To fork a project is to take the source from someone's repository at certain point in time, and apply your own diverging changes to it, you would clone the remote repository to create a copy of it, then do your own work in your local repository and commit changes.

* **Merge :**

Merge is used to combine one or more branches into the branch you have checked out. It will then advance the current branch to the result of the merge.

* **Clone :**

Serve to creates remote-tracking branches for each branch in the cloned repository, and creates and checks out an initial branch that is forked from the cloned repository’s currently active branch.

* **Pull :**

After using the push command, We can check for changes on our GitHub repository and pull down any new changes.

* **Pull request :**

Is simply used to generate an example message body to email to someone.