Git challendge

Part 3

* Github: est un service web d'hébergement et de gestion de développement de logiciels, utilisant le logiciel de gestion de versions Git.
* Github a ete lancé le 10 avril 2008 mais le développement a commencé le 19 octobre 2007
* GitHub est centré vers l'aspect social du développement. En plus d'offrir l'hébergement de projets avec Git, le site offre de nombreuses fonctionnalités habituellement retrouvées sur les réseaux sociaux comme les flux, la possibilité de suivre des personnes ou des projets ainsi que des graphes de réseaux pour les dépôts (de l'anglais *repository*) . GitHub offre aussi la possibilité de créer un wiki et une page web pour chaque dépôt.
* Il a été écrit par Chris Wanstrath, PJ Hyett et Tom Preston-Werner
* Plateforme simulaire : Gitlab, bitbucket, gogs, Bonobo git server, Gitiles, Gitorious, Gerrit etc...

J'utilise git pour faciliter le travail collaboratif avec mes collegues et gérer les versions des logiciels dévelopés. Aussi grâce aux repository je peux facilement managé mes projets et les déployés sur n'importe quelle machine.

Part 4 :Git tutorial work

* git init
* git status
* git status
* git add octocat.txt
* git status
* git commit -m "Add cute octocat story"
* git add '\*.txt'
* git commit -m "add all the octocat txt files"
* git log
* git remote add origin <https://github.com/try-git/try_git.git>
* git push -u origin master
* git pull origin master
* git diff HEAD
* git add octofamily/octodog.txt
* git diff --staged
* git reset octofamily/octodog.txt
* git checkout -- octocat.txt
* git branch clean\_up
* git checkout clean\_up
* git rm '\*.txt'
* git commit -m 'remove all the cats'
* git checkout master
* git merge clean\_up
* git branch -d clean\_up
* git push

Part 5 :

* Repository : emplacement utilisé par git pour garder l'historique du travail
* Commit : permet de valider les modifications apportées grace à la commande git add
* Push: pousser les modifications sur le serveur
* branch : permet de creer une nouvelle branche
* fork : consiste à récuperer le projet a un certain point afin d'y appliquer ses propres modifications.
* Merge: permet de reconcilier 2 ou plusieurs branche
* Clone : permet d'obtenir une copie d'un dépôt git existant
* Pull : récupère généralement les données depuis le serveur qui a été initialement cloné et essaie de les fusionner dans votre branche de travail actuel
* Pull request : permet de notifier les autres utilisateurs sur les changements effectués après un push sur le dépot.

Part 6: