**Part 3 :**

**What is GitHub ?**

GitHub is a web hosting service and software development management, using the Git version control software. It is considered to be the social network of developers.

**When was it created?**

GitHub was launched on 10 April 2008. The development began on 19 October 2007.

**Why?**

Github allows easy viewing of files, which allows you to see all changes made on this one chronologically and allow backtracking when necessary.

**By who?**

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

**What similar platforms exist?**

GitLab, *Gogs* (Go Git Service) …

**Why would you use such a platform?**

* Facilitate management of versioning my source codes.
* Participate in open source projects developments.
* Allow other members to bring their contributions on my projects.
* Follow good practices in terms of developments.
* Chat with other enthusiasts.

**Part 4 : Tutorial**

Press enter to submit commands

> git init

$ git status

$ git status

$ git add octocat.txt

$ git status

$ git commit -m "Addcute octocat story"

$ git add '\*.txt'

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

$ 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 commit -m "Remove all the cats"

$ git checkout master

$ git merge clean\_up

$ git branch -d clean up

$ git branch -d clean\_up

$ git push

>

**Part 5:**

**Repository:** This is the place that contains the files of the project, with all of its history. Each project developer has its local repository. There is also a repository named origin acting as a server.

**Commit:** a commit is a set of changes made and validated.

**Push:** Send commits to a Remote Repository.

**Branch:** A branch allows you to make changes in parallel. A div X works on the branch A, Y dev working on branch B. No one sees the work of the other.

**Fork:** A fork is a personal copy of another user's repository that lives on your account. Forks allow you to freely make changes to a project without affecting the original.

**Merge:** Merging takes the changes from one branch (in the same repository or from a fork), and applies them into another.

**Clone:** A clone is a copy of a repository that lives on your computer instead of on a website's server somewhere, or the act of making that copy. With your clone you can edit the files in your preferred editor and use Git to keep track of your changes without having to be online.

**Pull:** Recover commits other team members from the Repository.

**Pull request:** Pull requests are proposed changes to a repository submitted by a user and accepted or rejected by a repository's collaborators.