**Part 3**

Source :

<https://fr.wikipedia.org/wiki/GitHub>

<https://fr.wikipedia.org/wiki/Git>

[https://git-scm.com](https://git-scm.com/)

<https://www.ionos.fr/digitalguide/sites-internet/developpement-web/alternatives-a-github/>

**What is Github ?**

**Github** est un service web d’hébergement et de gestion de développement de logiciels, utilisant le logiciel de gestion de versions **Git** (wikipedia).

Git : c’est un logiciel de versioning (gestion des versions) décentralisé. Le code informatique développé est stocké non seulement sur l’ordinateur de chaque contributeur du projet, mais il peut également l'être sur un serveur dédié (ex : Github). Il permet de savoir ***Qui Quand Quoi Pourquoi*** on a modifié un fichier.

**When was it created ?**

GitHub a été lancé **le 10 avril 2008.**

Le 19 octobre 2007 **Chris Wanstranth** crée le premier engagement pour ce qui deviendrait GitHub.

**Why ?**

pour permettre aux développeurs de pouvoir communiquer, partager leur code et même récupérer du code d’autres repository.

**By who ?**

Ce site est développé en *Ruby on Rails et Erlang* par **Chris Wanstrath, PJ Hyett et Tom Preston-Werner**

**What similar platforms exist ?**

|  |  |  |
| --- | --- | --- |
| Hébergeurs | Centralisés (SVN) | Décentralisés (git) |
| Bitbucket |  |  |
| Gitlab |  |  |
| Apache allura |  |  |
| RiouxSVN |  |  |

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

D’abord basé sur le logiciel de gestion de versions git qui est décentralisé.

Ensuite la proposition des comptes professionnels payants, ainsi que des comptes gratuits pour les projets de logiciels libres. Il assure le contrôle d'accès et à la collaboration comme le suivi des bugs, la gestion de tâches et un wiki pour chaque projet.

Enfin sa première place parmi les hébergeurs de gestion de versions. Ce qui fait qu’il y à une forte communauté et plein de repository qui pourrait servir tout développeur.

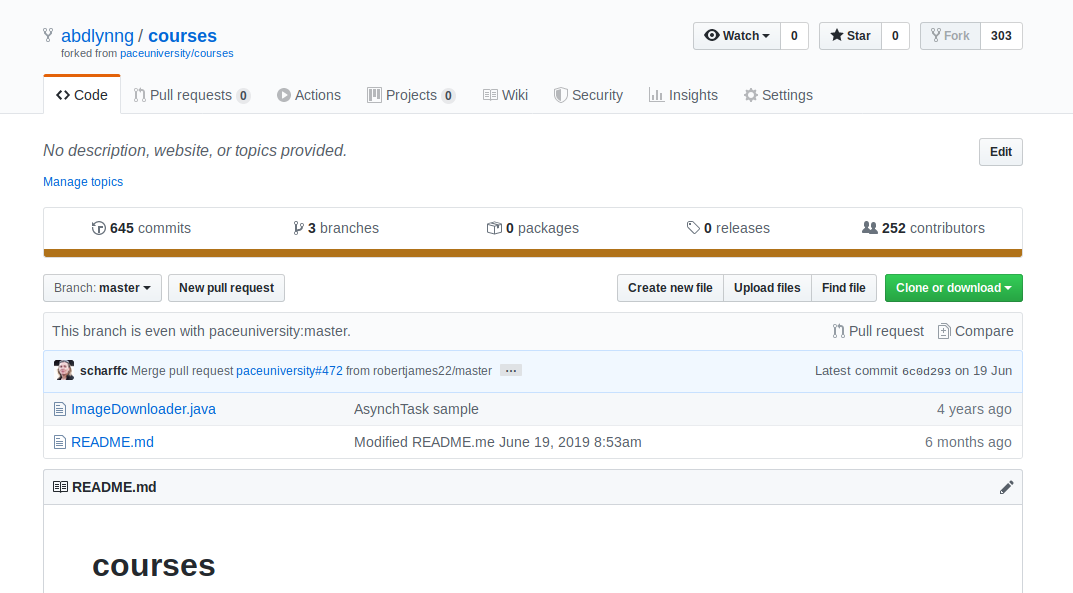
**Part 4**

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

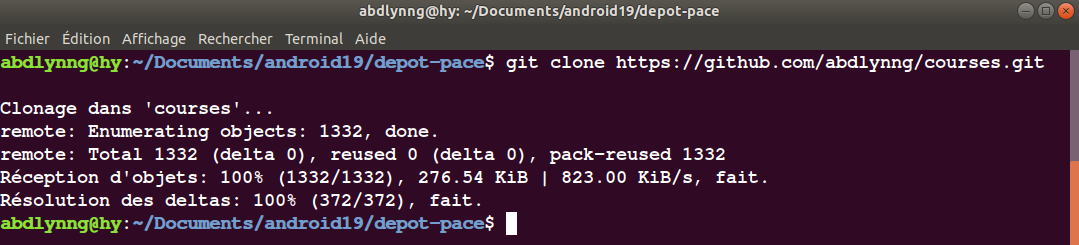
* **Repository**: c’est le répertoire du projet qui va contenir l’ensemble des fichiers et stocke l’historique de révision de chaque fichier.
* **Commit**:la validation cohérente des modifications chronologiques du projet dans l’historique
* **Push**: consiste à envoyer son repository local vers le remote distant (ex : github)
* **Branch**: signifie de diverger de la ligne principale de développement et continuer à travailler sans impacter cette ligne. Le nom de branche par défaut dans Git est master.
* **Fork**: est un nouveau logiciel créé à partir du code source d'un logiciel existant
* **Merge**:récupération fonctionnelle et intégrale d'une branche dans une autre
* **Clone**: Intégration d’une remote distant dans son environnement local et de pouvoir y apporter des modifications à l’avenir
* **Pull**: c’est l’inverse de push c’est a dire importé les modifications distantes en local
* **Pull request**: consiste à demander au détenteur du dépôt original de prendre en compte les modifications apportées.

**Part 6**

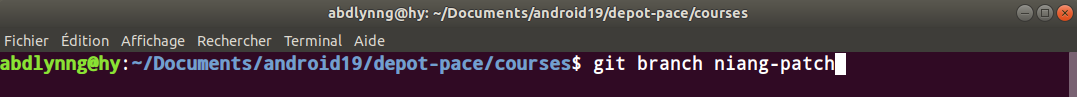
1. fork du projet depuis : <https://github.com/paceuniversity/courses>



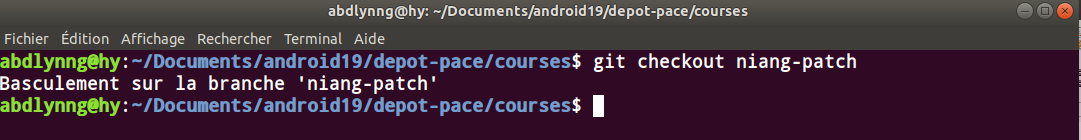
2. clonage du projet dans notre répertoire local



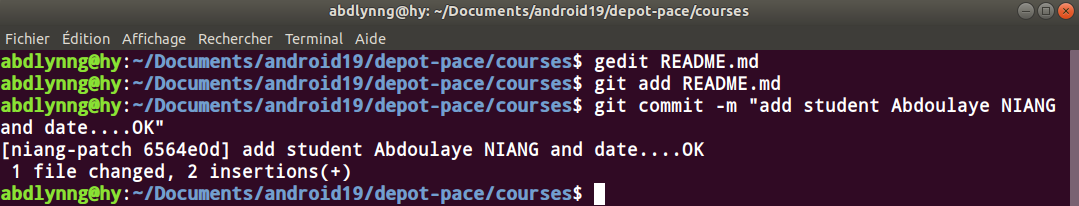
3. création de notre branch de travail



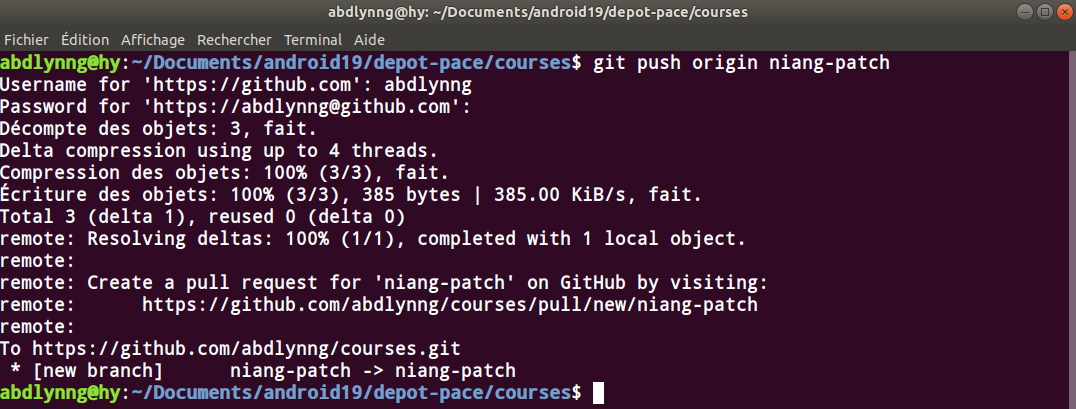
basculement sur la nouvelle branche



on modifie le Readme.md et faire le commit



4. push



5. pull Request

