



Task 2 : GitHub : Princip, Concept and Functionality

réalisé par :

DOUKH mohamed malek

Hammouda Abdelillah

specialité :

M1 ELNSE

Table des matières

1	Introduction	2
2	Différence entre Git et GitHub	2
3	Principe de fonctionnement	3
4	Le concept de GitHub	3
5	Les fonctionnalités de GitHub	3
6	Création d'un compte de GitHub	4
7	Conclusion	7

1 Introduction

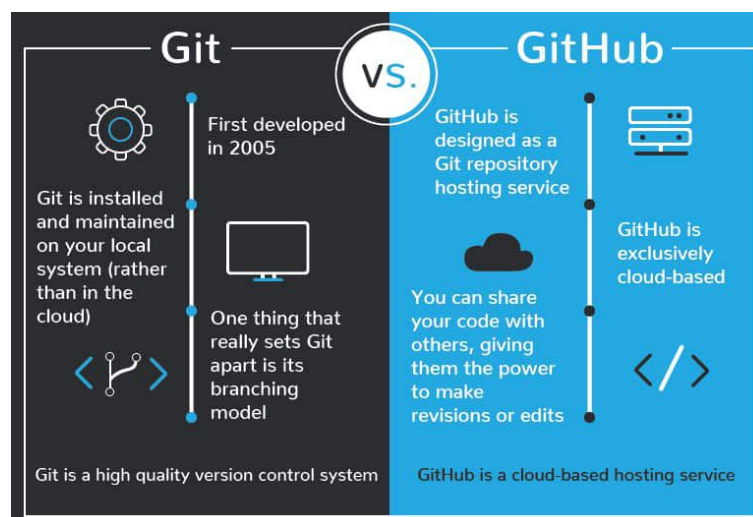
GitHub est une plateforme incontournable pour les développeurs souhaitant collaborer efficacement sur des projets logiciels. En se basant sur le système de contrôle de version Git, GitHub offre un ensemble complet d'outils permettant de héberger, gérer et collaborer autour du code source. Avec GitHub, les développeurs peuvent facilement partager leurs dépôts, proposer des modifications via des demandes de tirage, et suivre les problèmes liés à leurs projets. En plus de faciliter la collaboration, GitHub favorise également la transparence et la traçabilité des modifications apportées au code grâce à des fonctionnalités telles que la documentation intégrée, les wikis, et les outils d'intégration continue. En résumé, GitHub joue un rôle central dans le processus de développement logiciel moderne, offrant un environnement propice à la collaboration et à la qualité du code.

2 Différence entre Git et GitHub

1. Get : Cette orthographe pourrait être une faute de frappe, mais si vous voulez parler de "Git", c'est un système de contrôle de version distribué largement utilisé pour le suivi des modifications dans le code source lors du développement de logiciels. Git permet aux développeurs de travailler efficacement sur des projets en équipe, de suivre l'historique des modifications et de fusionner les différentes branches de code. C'est la technologie sous-jacente sur laquelle GitHub est construit.

2. GitHub : GitHub est une plateforme web qui utilise Git pour fournir des fonctionnalités supplémentaires de gestion de projet, de collaboration et de partage de code. Il offre un hébergement de dépôt Git, des outils de suivi des problèmes, des fonctionnalités de demandes de tirage pour proposer des modifications, ainsi que des fonctionnalités sociales permettant aux développeurs de discuter et de contribuer à des projets open source.

En bref, Git est le système de contrôle de version, tandis que GitHub est une plateforme construite autour de Git, offrant des fonctionnalités supplémentaires pour faciliter la collaboration et la gestion de projet pour les développeurs.



3 Principe de fonctionnement

GitHub fonctionne comme une plateforme collaborative incontournable pour les développeurs de logiciels. Au cœur de son fonctionnement se trouve Git, un système de contrôle de version distribué. GitHub offre un espace d'hébergement pour les dépôts Git, permettant aux développeurs de stocker leur code source et de suivre son évolution au fil du temps. Les développeurs peuvent travailler sur des branches distinctes, puis fusionner leurs modifications via des demandes de tirage. Cette approche facilite la collaboration, tout en maintenant un historique clair et complet des modifications. De plus, GitHub propose des outils de gestion de projet, comme le suivi des problèmes et les wikis, ainsi que des fonctionnalités sociales telles que les étoiles et les abonnements aux dépôts, encourageant ainsi la collaboration ouverte et le partage de connaissances au sein de la communauté des développeurs. En intégrant également des fonctionnalités d'intégration continue et de déploiement automatique, GitHub permet aux équipes de développer, tester et déployer leur code de manière efficace et transparente. En résumé, GitHub révolutionne la façon dont les développeurs collaborent, partagent et développent des logiciels, en créant un écosystème dynamique et ouvert propice à l'innovation et à la création.

4 Le concept de GitHub

Le concept de GitHub englobe les idées et les principes fondamentaux qui définissent la raison d'être et les objectifs de la plateforme. Cela inclut des éléments tels que la collaboration ouverte, la transparence, la gestion efficace du code source, la promotion de la qualité du code, la création d'une communauté de développeurs, et bien d'autres encore. Ces principes sont essentiels pour comprendre la philosophie qui guide le développement et l'utilisation de GitHub en tant que plateforme de développement de logiciels. En mettant l'accent sur ces concepts, GitHub vise à fournir un environnement propice à la collaboration, à l'innovation et à la création de logiciels de qualité supérieure.[1]

5 Les fonctionnalités de GitHub

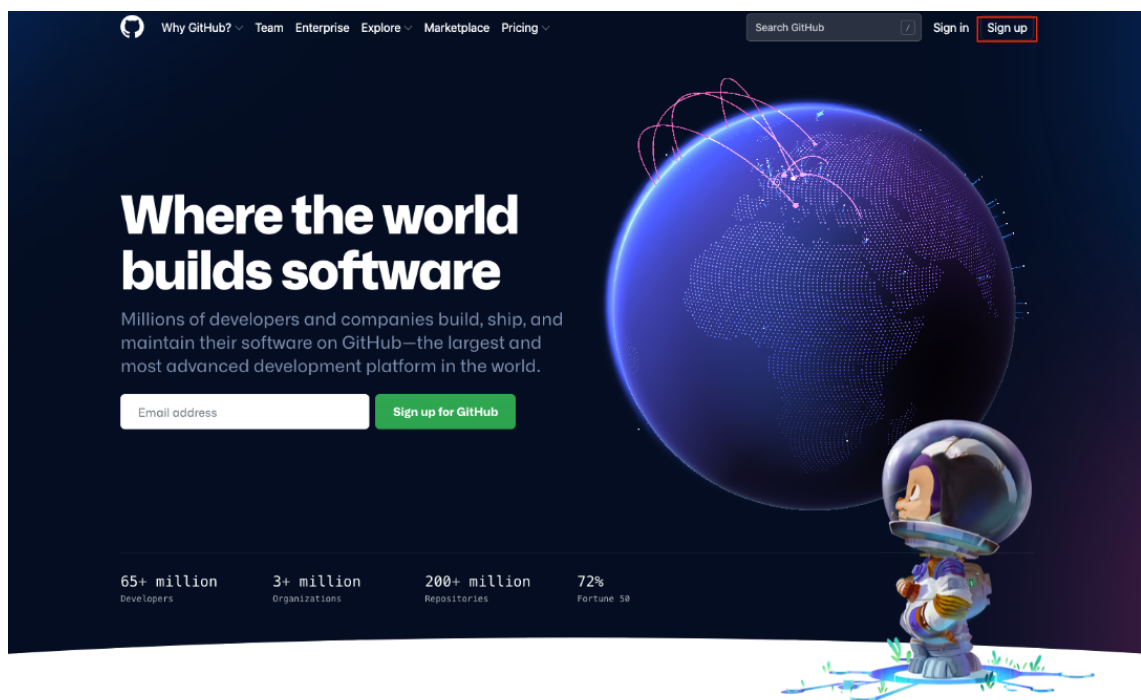
1. **Hébergement de dépôts Git** : GitHub permet aux développeurs de stocker leurs dépôts Git sur la plateforme, offrant ainsi un espace centralisé pour le code source de leurs projets.
2. **Demandes de tirage (Pull Requests)** : Les développeurs peuvent proposer des modifications à un dépôt en créant une demande de tirage. Cela permet aux autres contributeurs de réviser les modifications, de discuter des changements et de les fusionner dans le dépôt principal.
3. **Suivi des problèmes (Issue Tracking)** : GitHub propose un système intégré de suivi des problèmes permettant de signaler, suivre et résoudre les bogues, les demandes de fonctionnalités et d'autres tâches liées au développement du logiciel.

4. **Wikis et documentation** : Les utilisateurs peuvent créer et héberger des wikis pour documenter leurs projets, ce qui facilite la collaboration et le partage des connaissances au sein de l'équipe de développement.
5. **Gestion des branches** : GitHub facilite la création, la fusion et la gestion des branches de code, ce qui permet aux développeurs de travailler sur des fonctionnalités ou des correctifs de manière isolée avant de les fusionner dans la branche principale.
6. **Intégration continue (CI) et déploiement** : GitHub offre des outils d'intégration continue qui permettent d'automatiser les tests et les déploiements à chaque modification du code source, garantissant ainsi la qualité du code et facilitant les processus de déploiement.
7. **Contributions open source** : GitHub favorise la collaboration ouverte en permettant aux développeurs de contribuer à des projets open source, de signaler des problèmes et de proposer des améliorations, ce qui contribue à l'essor de la communauté des logiciels libres.
8. **Réseautage social** : GitHub permet aux utilisateurs de suivre d'autres développeurs, de "starrer" des projets qui les intéressent, et de participer à des discussions sur le code source via des commentaires et des discussions.

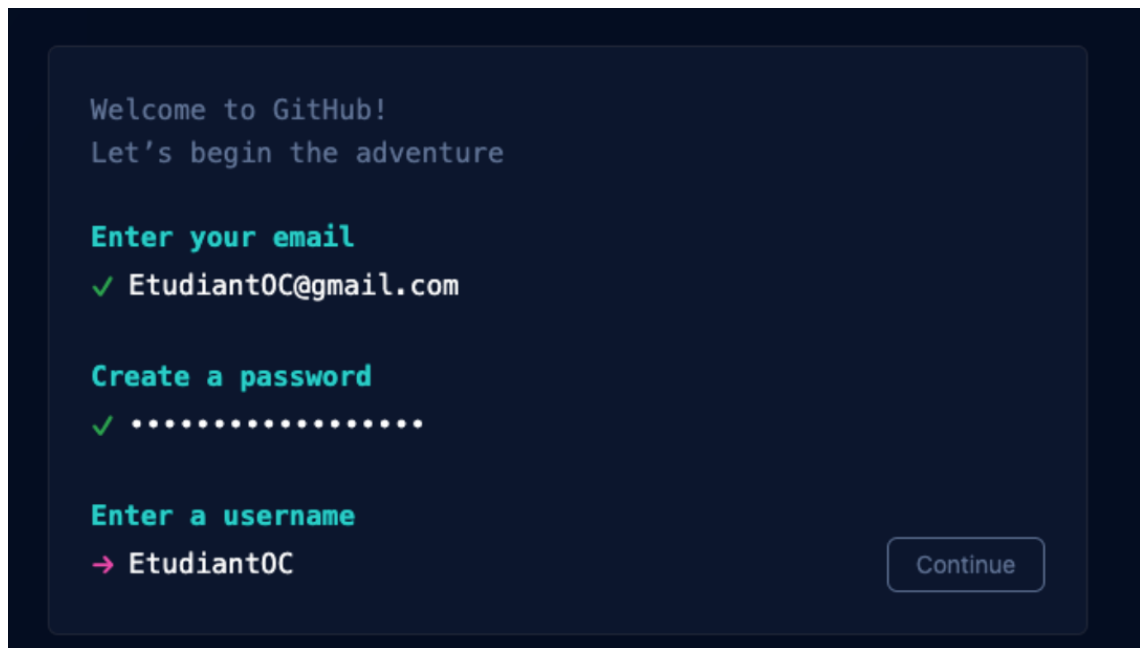
[2]

6 Création d'un compte de GitHub

Comme nous l'avons vu déjà, GitHub est un service en ligne qui va vous permettre d'héberger vos dépôts distants. Pour créer votre compte GitHub, rendez-vous sur la page d'accueil et cliquez sur Sign up



On vous demandera alors de renseigner un e-mail, un mot de passe et un nom d'utilisateur.

A screenshot of the GitHub registration interface. It features a dark blue background with a central white rounded rectangle containing the registration steps. The steps are: 'Welcome to GitHub! Let's begin the adventure', 'Enter your email' with a green checkmark and the email 'Etudiant0C@gmail.com', 'Create a password' with a green checkmark and a masked password '.....', and 'Enter a username' with a pink arrow and the username 'Etudiant0C'. A 'Continue' button is located at the bottom right of the form.

Welcome to GitHub!
Let's begin the adventure

Enter your email
✓ Etudiant0C@gmail.com

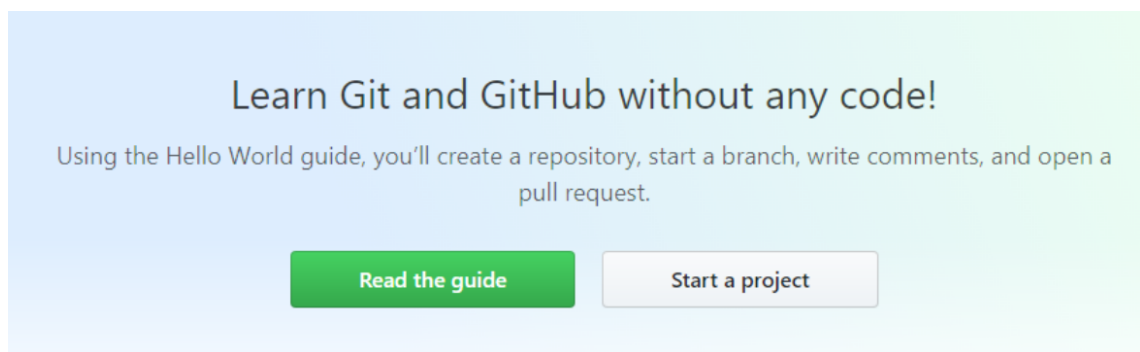
Create a password
✓

Enter a username
→ Etudiant0C

Continue

Et voilà, vous êtes à présent inscrit sur GitHub ! Par défaut, GitHub est gratuit. Mais sachez qu'il existe également des offres payantes si vous décidez de passer à la vitesse supérieure.

L'interface Repositories est l'emplacement où vous pourrez créer et retrouver vos dépôts existants. Pour créer un projet, il suffit de cliquer sur "Start a project".

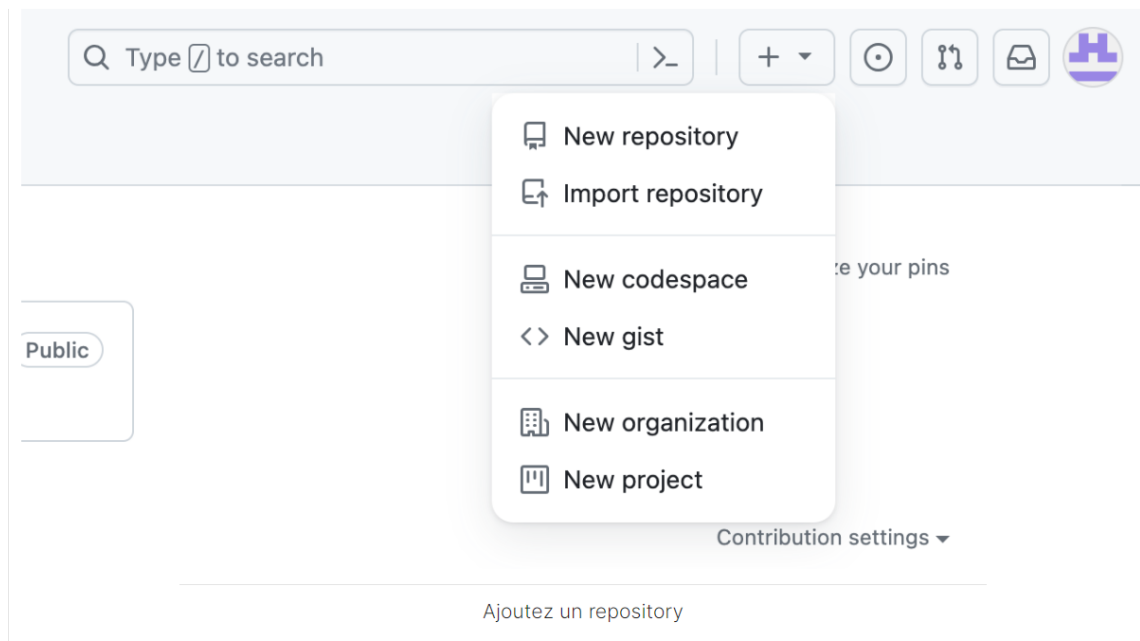
A screenshot of a GitHub banner with a light blue and green gradient background. The text reads: 'Learn Git and GitHub without any code!', 'Using the Hello World guide, you'll create a repository, start a branch, write comments, and open a pull request.', and two buttons: 'Read the guide' (green) and 'Start a project' (light blue).

Learn Git and GitHub without any code!

Using the Hello World guide, you'll create a repository, start a branch, write comments, and open a pull request.

Read the guide Start a project



Cliquez sur le + dans le coin supérieur droit, pour faire apparaître l'option New repository.



Choisissez un nom simple pour votre dépôt. Dans ce cours, nous utiliserons “OpenclassroomsProject”.


Puis, choisissez si vous souhaitez créer un dépôt public ou privé.


Owner * **Repository name ***

 EtudiantOC / 

Great repository names are short and memorable. Need inspiration? How about [sturdy-octo-parakeet?](#)

Description (optional)

☒  **Public**
Anyone on the internet can see this repository. You choose who can commit.

☐  **Private**
You choose who can see and commit to this repository.

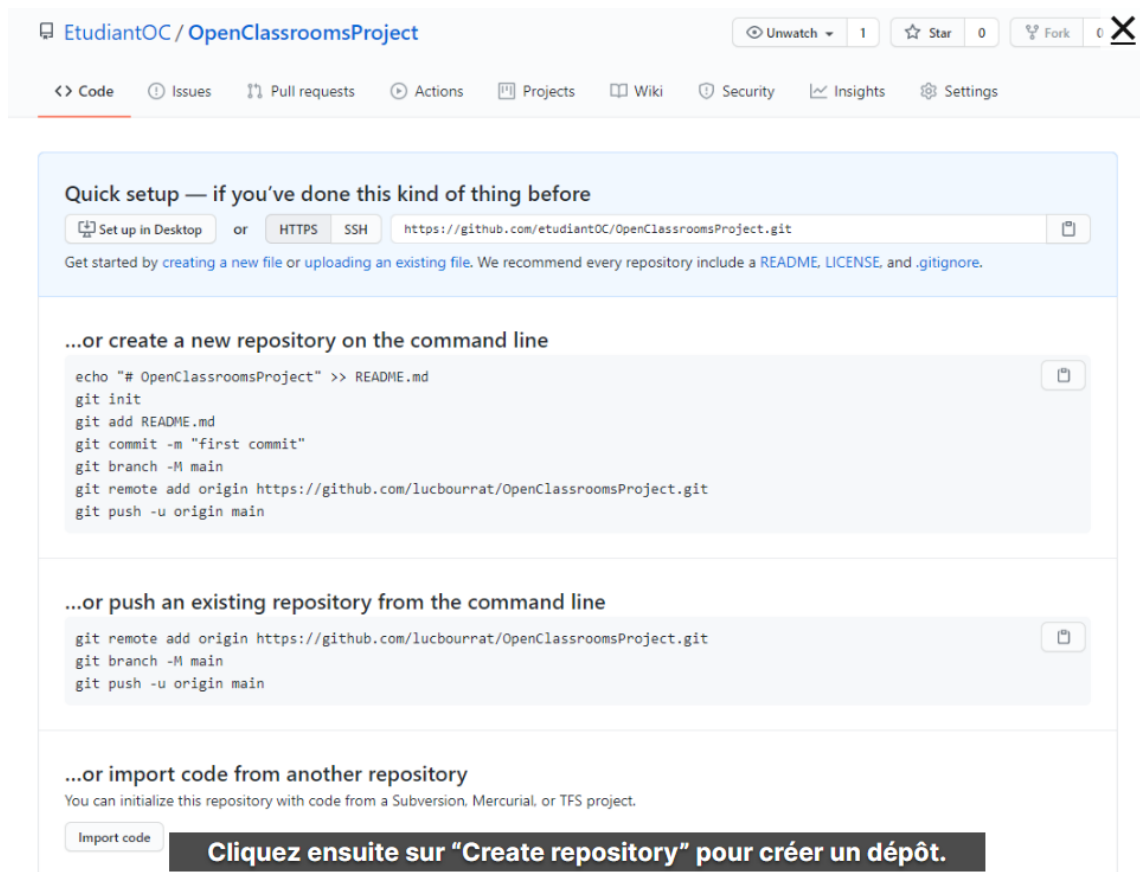
Initialize this repository with:
Skip this step if you're importing an existing repository.

☐ **Add a README file**
This is where you can write a long description for your project. [Learn more.](#)

☐ **Add .gitignore**
Choose which files not to track from a list of templates. [Learn more.](#)

remarque : README est un fichier qui indique les informations clés de votre projet : description, environnement à utiliser, dépendances possibles et droits d'auteurs. C'est un peu comme le mode d'emploi de votre projet.

gitignore est un fichier qui permet d'ignorer certains fichiers de votre projet Git. Nous reviendrons là-dessus plus tard.



7 Conclusion

GitHub est une plateforme incontournable dans le domaine du développement logiciel, offrant des fonctionnalités de contrôle de version, de gestion de projets et d'hébergement de code. Avec son interface conviviale et sa communauté dynamique, GitHub facilite la collaboration entre les développeurs de tous niveaux. Grâce à son système de contrôle de version basé sur Git, les équipes peuvent suivre les modifications, expérimenter en toute confiance et maintenir l'intégrité de leurs projets. Les nombreuses intégrations avec des outils de développement tels que les pipelines CI/CD et les gestionnaires de tâches renforcent la productivité des équipes. Enfin, la communauté active de GitHub favorise l'innovation et le partage des connaissances, faisant de la plateforme un pilier essentiel de l'écosystème du développement logiciel.

Références

- [1] Valerio COSENTINO, Javier LUIS et Jordi CABOT. “Findings from GitHub : methods, datasets and limitations”. In : *Proceedings of the 13th International Conference on Mining Software Repositories*. 2016, p. 137-141.
- [2] Eirini KALLIAMVAKOU et al. “The promises and perils of mining github”. In : *Proceedings of the 11th working conference on mining software repositories*. 2014, p. 92-101.