**NGONO NGUIETSI**

**VANINA SYLLA**

**Petite synthèse sur le GIT et le GITHUB**

**Qu'est-ce que Git ?**

GIT est un système de gestion de versions de code source distribué, utilisé pour suivre les modifications apportées au code source d'un projet logiciel. Il permet aux développeurs de travailler sur le même code source simultanément, de fusionner les modifications et de suivre l'historique des versions une plateforme appelé Github.

**Qu'est-ce que Github?**

GitHub est une plateforme de développement collaboratif basée sur Git, un système de contrôle de version distribué. Elle permet aux développeurs de travailler ensemble sur des projets, de partager du code source et de gérer les modifications apportées à ce code.

**Différence entre Git et Github**.

Disons que Git est tout simplement un outil qui vas nous permettre de gérer nos différents projets c'est à dire manipuler, envoyer nos différents projets sur le Web et les récupérer et Github est donc la plateforme qui doit héberger ces différents projets Git.

Pour résumer, Git n'est rien sans Github et inversement.

**Différence entre le push et le pull**

Le push et le pull sont des commandes utilisées dans Git pour synchroniser les modifications entre un dépôt local et un dépôt distant.

Le **push** est utilisé pour envoyer les modifications locales vers le dépôt distant. Cela signifie que toutes les modifications que vous avez effectuées localement seront poussées vers le dépôt distant, mettant ainsi à jour son contenu avec vos changements.

**La** **commande** **push** est généralement utilisée après avoir effectué des commits locaux pour partager vos modifications avec d'autres collaborateurs travaillant sur le même projet.

Par contre, le **pull** est utilisé pour récupérer les modifications du dépôt distant vers votre dépôt local. Cela signifie que toutes les modifications qui ont été effectuées par d'autres collaborateurs sur le projet seront récupérées et fusionnées avec votre code local.

**La** **commande pull** est généralement utilisée lorsque vous souhaitez mettre à jour votre code local avec les dernières modifications apportées par d'autres collaborateurs.

En résumé, le push envoie vos modifications locales vers le dépôt distant, tandis que le pull récupère les modifications du dépôt distant vers votre dépôt local.

**C’est quoi la collaboration et le versioning sur Github**

**La collaboration** sur Github est un processus qui permet à plusieurs personnes de travailler ensemble sur un projet logiciel. Cela implique généralement la création d'un référentiel (repository) sur Github, où les développeurs peuvent télécharger (pull) le code source, apporter des modifications et proposer des changements (pull requests). Les autres membres de l'équipe peuvent alors examiner les modifications, commenter et approuver les demandes de tirage avant de fusionner les modifications dans le référentiel principal. Cela facilite la collaboration entre les développeurs, en leur permettant de travailler simultanément sur différentes parties du projet et en gardant une trace des modifications apportées.

**Le versioning** sur Github fait référence à la gestion des différentes versions d'un projet logiciel. Il permet de suivre les modifications apportées au code source au fil du temps, de manière à pouvoir revenir à une version antérieure si nécessaire.

Sur Github, le **versioning** est principalement réalisé à l'aide de Git, un système de contrôle de version distribué. Git permet aux développeurs de travailler simultanément sur un même projet, en gardant une trace des modifications apportées à chaque fichier. Chaque modification est enregistrée sous la forme d'un "commit", qui représente une unité logique de changement. Chaque commit est accompagné d'un message décrivant les modifications apportées.

Grâce au **versioning** sur Github, il est possible de visualiser l'historique complet des modifications d'un projet, de comparer les différentes versions et même de fusionner des branches parallèles pour intégrer les changements dans une version principale du projet.

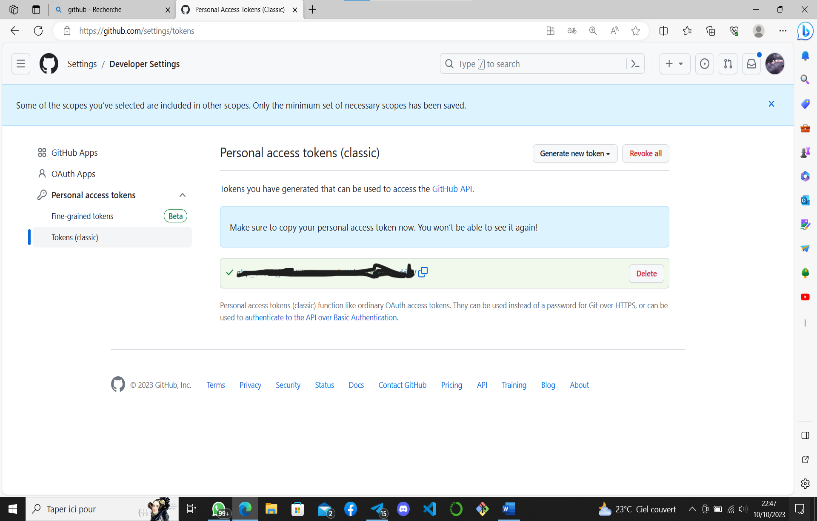
En résumé, le versioning sur Github permet aux développeurs de collaborer efficacement sur un projet tout en gardant un historique détaillé des modifications effectuées.

**C’est quoi un Token ?**

Un token est une chaîne de caractères générée par la plateforme pour authentifier et autoriser l'accès à certaines fonctionnalités ou ressources. Les tokens sont souvent utilisés pour permettre aux applications tierces d'interagir avec les dépôts GitHub, par exemple pour effectuer des opérations de lecture ou d'écriture via l'API GitHub. On distingue 2 types de Token (jetons à granularité fine et jetons (classique)

Un jeton classique est une unité indivisible qui représente une valeur unique. Par exemple, dans un langage de programmation, un jeton classique peut représenter un mot-clé, un opérateur ou une constante.

En revanche, un jeton à granularité fine est une unité plus petite et divisible qui représente une partie d'une valeur ou d'un symbole. Par exemple, dans un langage de programmation, un jeton à granularité fine peut représenter un caractère individuel d'une chaîne de caractères ou un élément d'une liste.



La principale différence réside donc dans la taille et la divisibilité des unités représentées par les jetons. Les jetons classiques représentent des valeurs complètes et indivisibles, tandis que les jetons à granularité fine représentent des parties plus petites et divisibles de ces valeurs.

**C’est quoi un READMe ?**

Un README est un fichier texte qui se trouve généralement à la racine d'un dépôt GitHub et qui sert à fournir des informations sur le projet. Il contient souvent une description du projet, des instructions d'installation et d'utilisation, ainsi que toute autre information pertinente pour les contributeurs ou les utilisateurs du projet.

**C’est quoi une Licence ?**

Une licence sur GitHub est un document juridique qui définit les conditions d'utilisation et de distribution du code source d'un projet. Lorsqu'un développeur publie son code sur GitHub, il peut choisir une licence spécifique pour indiquer comment les autres peuvent utiliser, modifier et distribuer son code. Les licences open source courantes incluent la licence MIT, la licence Apache et la licence GNU GPL.