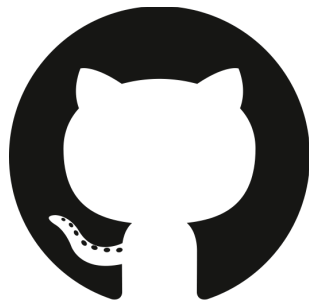
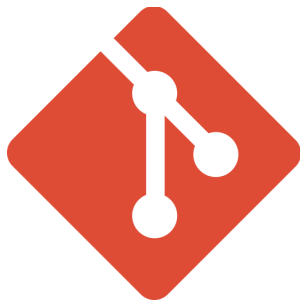




GDSC UCB

# Apprendre Git et Github



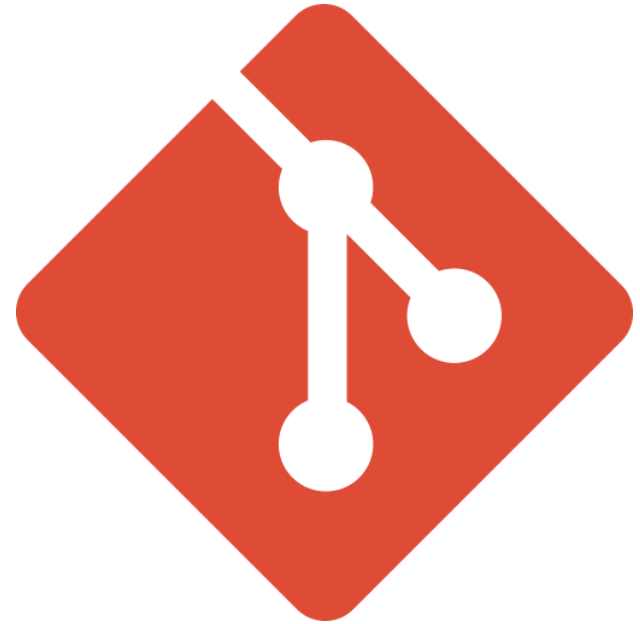
AKONKWA USHINDI Isaac  
FullStack Developer

# GIT

Git est un logiciel de gestion de versions décentralisé. C'est un logiciel libre et gratuit, créé en 2005 par **Linus Torvalds**, auteur du noyau Linux, et distribué selon les termes de la licence publique générale GNU version 2. Le principal contributeur actuel de Git, et ce depuis plus de 16 ans, est **Junio C Hamano**, un software engineer et hacker Japonais.

## Langages de programmation

**Git** a été programmé en Python, C, C++, Perl, script shell, langage de commande d'outil et plein d'autre mais ceux là sont les primordiaux



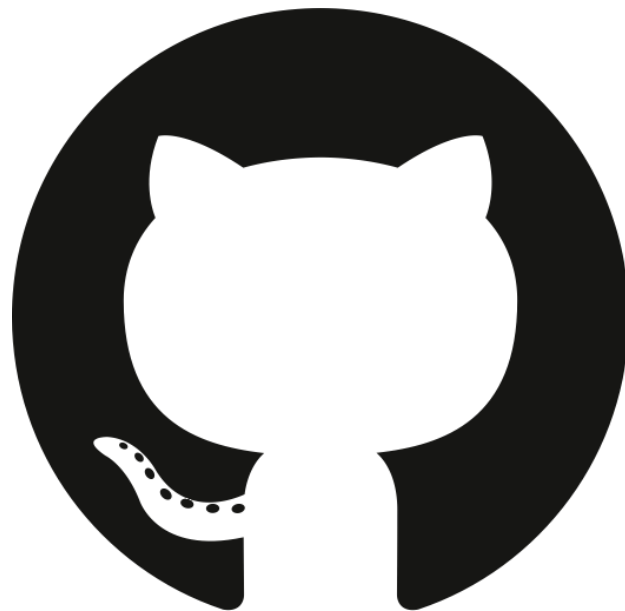
Lien de téléchargement:

<https://git-scm.com/downloads>

# GITHUB

GitHub est une plateforme de développement collaboratif basée sur Git. Lancé en 2008, GitHub est devenu un lieu central pour le développement de logiciels open source, ainsi qu'un outil essentiel pour de nombreux développeurs professionnels. GitHub intervient dans plusieurs aspects entre autres:

- L'hébergement de code
- Le contrôle de version
- La collaboration
- Le gestion des problèmes
- Le gestion de projets
- Etc.



Mon GitHub

<https://github.com/akon2020/>



# Par où commencer?

## La configuration de l'identité

La première chose à faire après l'installation de Git est de renseigner votre nom et votre adresse mail. Ces informations sont importantes car toutes les validations dans Git utilisent cet information

Pour la configuration du nom, tapez cette commande dans votre terminal

**git config --global user.name 'votre nom entre ces guimets'**

Viens le tour de l'adresse mail avec

**git config --global user.email 'votre email entre ces guimets'**

```
lookup.KeyValue  
f.constant(['em  
=tf.constant([G  
.lookup.StaticV  
_buckets=5)
```

# Les commandes fondamentales de Git

```
lookup.KeyValue  
f.constant(['em  
=tf.constant([G  
lookup.StaticV  
_buckets=5)
```

# Les commandes fondamentales de Git

Git regorge plusieurs mais nous n'allons traiter que les plus utiliser :

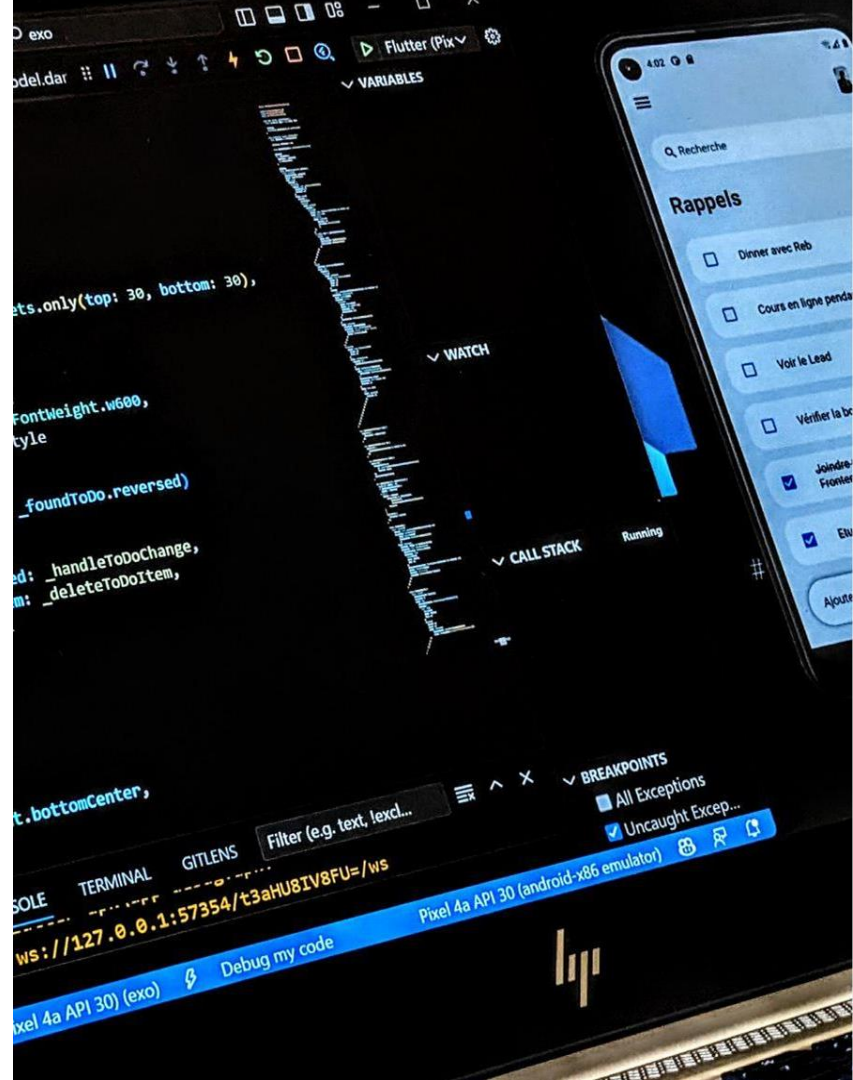
- **git init** : Cette commande est utilisée pour initialiser un nouveau dépôt Git dans le répertoire local
- **git clone** : Celle-ci est utilisée pour cloner un dépôt Git existant dans un dépôt distant.
- **git add** : Pour ajouter des fichiers à l'index pour ainsi faire le suivis avec Git.
- **git commit** : On utilise cette commande pour enregistrer les modifications apportées aux fichiers dans un dépôt Git.
- **git status** : Elle est utilisé pour affiche l'état des fichiers dans le répertoire de travail et de l'index.

# Les commandes fondamentales de Git

- **git diff** : cette manipulation affiche les différences entre les modifications de fichiers qui ne sont pas encore enregistrées.
- **git log** : Cette commande affiche l'historique des commits dans le dépôt.
- **git branch** : Cette commande fait trois choses manipulation; elle affiche, crée et supprime des branches.
- **git checkout** : Celle-ci permet de basculer entre les différentes branches.
- **git merge** : Elle fusionne les modifications de différentes branches.
- **git pull** : Elle est utilisée pour récupérer les modifications du dépôt distant et les fusionne avec la branche locale.
- **git push** : Pour Envoyer les modifications locales vers le dépôt distant.
- **git remote** : On utilise cette commande pour gérer les références aux dépôts distants.

**Voici un lien où vous  
pourrez apprendre à  
utiliser Git de manière  
pratique avec des  
exemples illustratifs**

**[https://learngitbranching.js.org/?locale=fr\\_FR](https://learngitbranching.js.org/?locale=fr_FR)**





**La collaboration lors de la conception d'un projet complexe reste la meilleure façon d'avancer rapidement et de faire mieux.**

**Ainsi, se sent fière d'un travail accompli.**



# Conclusion

Git est un outil puissant et indispensable pour tout développeur de logiciels. En maîtrisant les commandes fondamentales de Git, les développeurs peuvent travailler de manière efficace, collaborer avec d'autres membres de l'équipe, maintenir un historique clair des versions et gérer leurs projets de développement de manière professionnelle. Que ce soit pour le développement de logiciels open source ou pour des projets commerciaux, Git offre une solution robuste et fiable pour la gestion des versions de code source.



**Il nous reste plus qu'à marier Git et GitHub pour comprendre véritablement comment ces deux outils peuvent améliorer non seulement notre rentabilité mais aussi notre expérience en tant que développeur**

```
lookup.KeyValue  
f.constant(['en  
=tf.constant([G  
.lookup.StaticV  
_buckets=5)
```



**Il nous reste plus qu'à marier Git et GitHub pour comprendre véritablement comment ces deux outils peuvent améliorer non seulement notre rentabilité mais aussi notre expérience en tant que développeur**

```
lookup.KeyValue  
f.constant(['en  
=tf.constant([G  
lookup.StaticV  
_buckets=5)
```



**Trouvez ce support à ce dépôt distant**

**<https://github.com/Akon2020/GDSC-git>**

**Appliquez les instructions qui seront visible  
dans notre le dépôt distant pour avoir une  
petite idée de comment ça marche**

```
lookup.KeyValue  
f.constant(['en  
=tf.constant([G  
lookup.StaticV  
_buckets=5)
```

# Merci pour votre attention

👋 Place au question 👋

```
lookup.KeyValue  
f.constant(['en  
=tf.constant([G  
.lookup.StaticV  
_buckets=5)
```