



GitHub

Git est un système de gestion de version. Il est essentiel pour les développeurs surtout lorsqu'ils travaillent sur des projets en collaboration. Github permet de :

1. **Gérer les versions** : Git va enregistrer votre code à chaque fois que vous décidez de "push" (envoyer) votre travail. Cela permet si besoin de revenir en arrière en choisissant une version précédente de notre code.
2. **Collaborer** : Git permet aussi de travailler facilement en équipe sur le même projet. Chacun a sa copie du code et ils peuvent le fusionner.
3. **Branching et fusion (Branching and Merging)** : Avec Git, il est possible de créer différentes branches sur un projet. Cela permet de ne pas mettre sur la branche principal un code qui n'a pas été corrigé. Sur des projets de groupe, il est intéressant que chacun travaille sur une branche différente pour ne pas que les fichiers rentrent en conflit et de les fusionner lorsque tout le monde est sûr de sa version.
4. **Sauvegarde et sécurité** : Git vous permet tout simplement d'avoir une copie de votre code. Si vous choisissez de mettre votre code en privée, seul vous pourrez y accéder.
5. **Gain de temps** : Git permet d'automatiser de nombreuses tâches de gestion de versions. Il est plus efficace que de gérer manuellement des fichiers et des dossiers, et il facilite la recherche de modifications spécifiques.
6. **Communauté et support** : Git est a une grande communauté. Il est possible d'y trouvezr des ressources, tutos, solutions à des bugs.

Installer Git sur Linux

Tapez ceci dans un terminal :

```
sudo apt update
```

puis

```
sudo apt install git
```

On vérifie que tout soit bien installer en vérifiant la version.

```
git --version
```

Maintenant, il va falloir indiquer à notre console quel est notre compte Github.

```
git config --global user.name "Votre Nom"
```

```
git config --global user.email "votre@email.com"
```

Si un jour Github vous demande le mot de passe de votre compte, vous ne devez pas mettre le mot de passe que vous avez choisi mais un token (c'est une façon de sécurisé un peu plus votre compte Github).

Pour trouver le token :

- aller dans settings
- aller dans developer settings
- accéder à Personal access tokens
- Générez un nouveau token (classique)
- Configurez les autorisations : cocher repo, admin:public_key, admin:repo_hook, user, admin:pgp_key, admin:ssh_signing_key

Le token que l'on vous aura générer sera le mdp à entrer sur votre terminal.

Push dans un repo qui n'existe pas encore

Sur Github, créer un nouveau repository. Ensuite, dans le dossier que vous voulez push (votre dossier ou se trouve votre code) :

```
git init
```

```
git add .
```

```
git commit -m "Premier commit"
```

```
git remote add origin URL_du_dépôt
```

```
git push -u origin nom_de_la_branche
```

Création d'une branche et modification des fichiers

```
git checkout -b nom_branche
```

Modifier le fichier html en ajoutant un pied de page.

Le push dans le repo sur la bonne branche.

Push dans un repo qui a déjà été initialisé

```
git add .
```

```
git commit -m "nomDuCommit"
```

```
git push -u origin nom_de_la_branche
```

Fusionner les branches

Revenir à la branche principale.

```
git checkout master
```

```
git merge nom_de_la_branche_a_merge
```

S'il y a des conflits, les résoudre.

```
git add .  
git commit -m "Resolved merge conflicts"  
git push origin master
```

Vérifier que les changements ont bien été effectués.

Vous pouvez supprimer la nouvelle branche

```
git branch -d nom_de_la_branche
```



GitHub Cheat Sheet

Versionner son travail

Versionner en local

git init	initialise le dépôt (se mettre sur le bon dossier), mieux à faire depuis Github.com
git add .	ajoute toutes les modifications (le . symbolise tout)
git commit -m "explication"	créer un nouveau commit. git add pousse les fichiers en zone d'index, git commit les sauvegarde réellement dans un nouveau commit

Gérer les commits

git log	liste des commits
git log -n2	affiche les 2 derniers commits
git show sha-1	voir commit spécifique (cliquer molette souris pour coller)
git checkout sha-1	remettre la version du sha-1
git checkout master	remettre version la plus récente

Versionner sur un dépôt distant

git clone lien-github.com	récupérer travail depuis dépôt distant
git push -u origin master	pousse les modifications vers serveur
git push -f origin master	pousse de force des modifications (à manipuler avec précaution)

Naviguer dans Git Bash

pwd	savoir dans quel dossier je suis
mkdir "dossier"	créer un dossier (Make Directory)
touch fichier.txt	créer fichier
ls	liste le dossier courant
ls -la	liste tout plus précisément que ls
cd dossier	aller dans le dossier (Change Directory)
cd ..	Remonter d'un dossier

Initialisation de Git

git config --global user.name	"Mon Nom"
git config --global user.email	mon@mail.com
git config --global --list	Affiche nom et mail

Autres commandes

git status	état du fichier
git diff	affiche les modifs avant commit

Commit son projet sur Github

```
git add .
git commit -m "message"
git push -u origin master
```

