





Git & GitHub avec Visual Studio Code

Un guide pratique pour bien démarrer avec Git et GitHub depuis VS Code, avec une touche fun et pédagogique ! 🚀

Sommaire

1.  Quelques raccourcis
 2.  Mise en place de Git
 3.  Créer un projet **sans** versioning
 - 3.1 Local
 - 3.2 Lier à GitHub
 - 3.2.1 En cas d'erreur
 - 3.3 Récupérer les changements
 4.  Créer un projet **avec** versioning
 - 4.1 En local
 - 4.2 Créer des branches (versions)
 - 4.3 Lier à GitHub
 5.  Petit récap + bonus
-

1. Quelques raccourcis utiles dans VS Code

Action	Raccourci
Ouvrir terminal	Ctrl + J
Command Palette	Ctrl + Shift + P
Menu Git	Ctrl + Shift + G
Commit rapide	Ctrl + Shift + P → "Git Commit"
Rechercher un fichier	Ctrl + P

2. Mise en place de Git

Commandes à entrer dans Git Bash ou un terminal :

```
git config --global user.name "Ton Nom"
git config --global user.email "ton-email@example.com"
git config --list
```

```
# Utiliser 'main' par défaut au lieu de 'master'  
git config --global init.defaultBranch main
```

3. Créer un projet sans versioning

3.1 En local

Structure typique :

```
mon-projet/  
├── css/  
├── js/  
├── img/  
├── fonts/  
├── index.html  
└── .gitignore
```

Ouvrir Git Bash ou terminal dans le dossier :

```
git init  
git branch  
# Si master, renommer :  
git branch -m main  
git status  
git add .  
git commit -m "Ajout d'une fonctionnalité"
```

3.2 Lier à GitHub

1. Créer un dépôt vide sur GitHub (même nom recommandé)
2. Récupérer l'URL du dépôt
3. Puis :

```
git remote add origin https://github.com/utilisateur/mon-projet.git  
git branch -M main  
git push -u origin main  
git status
```

3.2.1 En cas d'erreur :

```
git pull --rebase origin main  
git push origin main  
git push --force origin main  
git status
```

3.3 Pour récupérer les changements :

```
git pull origin main
```

4. Créer un projet avec versioning

4.1 En local

```
git init  
git branch -m main  
git status  
git add chemin/vers/fichier
```

```
git commit -m "Ajout d'une fonctionnalité"
git log
```

Modifier un dernier commit :

```
git commit --amend -m "Nouveau message"
```

Ignorer des fichiers :

- Créer .gitignore
- Exemple pour ignorer le dossier img :

```
img/
git commit -m "Mise à jour du .gitignore"
git log
```

Annuler le dernier commit :

```
git reset --soft HEAD~1
```

Modifier un ancien commit (ex : 2e) :

```
git rebase -i HEAD~2
# Remplacer "pick" par "reword" → ESC, :wq
```

Voir les différences :

```
git diff
# ou sur un fichier spécifique
git diff nomFichier.ext
```

Astuce :

```
git commit -am "Commit + Add combiné"
```

4.2 Créer des branches

```
git branch test
git checkout test
# Commits sur test
```

```
git checkout main
git merge test
git branch -d test
git branch
```

4.3 Lier à GitHub

```
git remote add origin https://github.com/utilisateur/mon-projet.git
git push -u origin main
git status
```

5. 🧠 Petit récap + bonus

📌 Ressources utiles :

- Guide Git complet : <https://git-scm.com/book/fr/v2/>
- Télécharger Git : <https://git-scm.com/downloads>

- Extension VS Code GitHub Pull Requests :
<https://marketplace.visualstudio.com/items?itemName=GitHub.vscode-pull-request-github>

📌 Vérifier la version Git :

```
git --version
```

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 main	remettre version la plus récente

Versionner sur un dépôt distant

git clone lien-github.com	récupérer travail depuis dépôt distant
git push -u origin main	pousse les modifications vers serveur
git push -f origin main	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 mods avant commit

Commit son projet sur Github

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

