

# Git

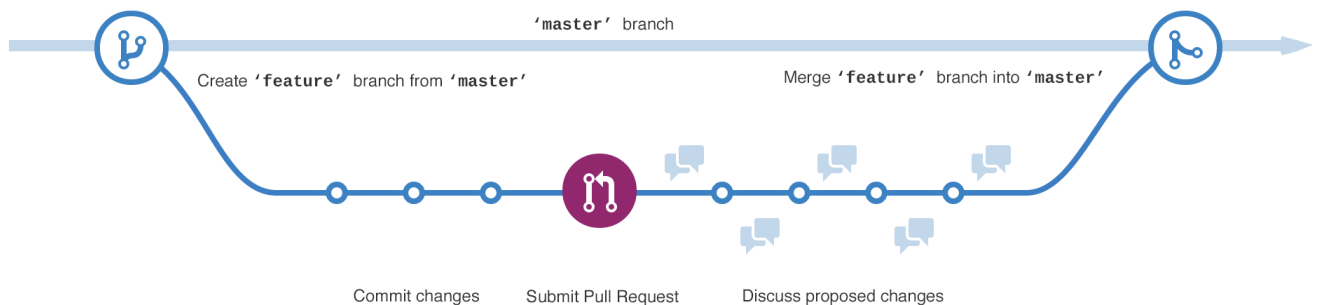
## Logiciel de versionning

Git est un système de gestion de versions partagé qui fut développé en 2005 par le créateur de Linux Linus Thorvalds

Les logiciels de contrôle de version conservent la trace de chaque modification apportée au code dans une base de données spéciale.

Si une erreur est commise, les développeurs peuvent revenir en arrière et comparer les versions antérieures du code pour aider à corriger l'erreur tout en minimisant les perturbations pour tous les membres de l'équipe.

<https://www.atlassian.com/git/tutorials/what-is-version-control>



## Installer git sur votre machine

Vérifier si git est installé

```
git --version
```

<https://git-scm.com/download/linux>

Interfaces graphiques

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

Ne stockez jamais vos dépôts Git à la racine du site Web  
ou bien configurez votre serveur Web de sorte  
que l'accès au répertoire Git soit impossible  
pour les personnes extérieures.

## Sécuriser votre dépôt git

<https://en.internetwache.org/dont-publicly-expose-git-or-how-we-downloaded-your-websites-sourcecode-an-analysis-of-alexas-1m-28-07-2015/>

## Créer un dossier de projet

Dans le terminal allez dans ce dossier

```
cd chemin/start
```

## Initialiser votre projet git

```
git init
```

## Créer la branche principale

```
git config --global init.defaultBranch <name>
```

## Cloner le répertoire de votre serveur en local

```
git clone https://monsite
```

## Vérifier le statut du répertoire

```
git status
```

## Historique des commits

```
git log
```

La commande « git log » liste les commits générés par ordre anti chronologique

## Refaire un commit pour des modifications oubliées

```
git commit --amend
```

## Reprendre le dernier commit

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

## Supprimer le dernier commit

```
git reset --hard HEAD~1
```

## Créer un compte Github

<https://github.com/>

<https://guides.github.com/activities/hello-world/>

GitHub est une plateforme d'hébergement de code pour le versioning et la collaboration. Elle vous permet, ainsi qu'à d'autres personnes, de travailler ensemble sur des projets, où que vous soyez.

### 1. **Créer un dépôt (repository) sur GitHub**

Ajouter un fichier README.md

### 2. **Créer une nouvelle branche**

### 3. **Faites un commit de vos changements dans README**

Ecrivez un message qui décrit vos changements.

### 4. **Ouvrez une PullRequest**

Les demandes de modification sont au cœur de la collaboration sur GitHub. Lorsque vous ouvrez une demande de retrait, vous proposez vos modifications et demandez à quelqu'un de les réviser, de les reprendre et de les fusionner dans sa branche.

### 5. **Merger votre PullRequest**

Ajouter les changements sur la nouvelle branche dans la branche principale

## Partager vos projets en ligne avec GitHub pages

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

1. Cliquer sur Settings
2. En bas de la page cliquez sur [GitHub pages](#)
3. Choisir une branche
4. Laisser sur root
5. Save
6. Votre URL est prête <http://user-name.github.io/projet/>

### Résumé

Settings => GitHub Pages => Choisir la branche + root => Save

Récupérer l'URL <https://web-god.github.io/burger/>

`Git clone https://github.com/Web-God/burger.git`

`Git branch site` - créer une branche

`Git checkout site` - aller sur la branche site

`Git add .` ajouter les modifications pour le commit

`Git commit -m «new change»` - ajouter les modifs dans HEAD avec un **message**

`Git status` - vérifier l'état des commits

### **Sur la branche principale**

`Git merge nombranche` - fusionner la branche de modifs dans la branche principale

`Git branch -d nombranche` - supprimer la branche de modifs

`Git push origin site` - pousser les modifications sur le serveur

## **Créer des raccourcis**

`git config --global alias.st status`

Avant de faire une modification, il faut récupérer le travail de vos collègues. effectuer sur la branche principale avec :

`Git pull origin site`

`Git fetch`

## **New feature**

### **Toujours faire les modifications sur la nouvelle branche**

`Git branch feat-navigation` => créer une nouvelle branche

`Git checkout feat-navigation` => aller sur cette nouvelle branche

Effectuer les modifications

`Git status`

`Git add .` (Tous les fichiers)

`Git commit -m « add navigation »`

`Git checkout site` => aller sur la branche principale

`Git merge feat-navigation` => merger la branche des modifications sur la branche principale

`git push origin site`

## Supprimer une branche

Git branch -d feature-logo

## Ignorer fichiers / dossiers

Créer un fichier .gitignore à la racine du dossier

```
.vscode/  
node_modules/  
package-lock.json  
package.json  
assets/.DS_Store  
.gitignore
```