

## Git introduction

### INTRODUCTION

Lors de la création d'un site web vous serez amené à travailler en équipe. Pour cela vous allez donc devoir apprendre la gestion de projet en groupe.

Git est ce qu'on appelle un logiciel de gestion de versions. Ses objectifs sont :

- travailler à plusieurs sur le projet sans écraser les modifications des autres
- savoir qui à modifié quoi tout au long de la vie du projet
- conserver un historique des modifications du projet
- conserver les anciennes versions du projet

Il existe de nombreux logiciels de gestion de version (SVN, par exemple), mais Git est un des plus puissants et des plus utilisés dans le monde informatique.

Son but est de gérer l'évolution du contenu sous la forme d'une arborescence, tout en restant simple d'utilisation.

### En savoir plus

Git a été créé par Linus Torvald, qui a également créé le noyau Linux.

Sur internet, un des outils les plus utilisés vient d'ailleurs de l'utilisation de Git : il s'agit de Github, un site sur lequel vous pouvez héberger gratuitement vos projets et où tout le monde peut apporter sa contribution. Github est devenu une gigantesque librairie de projets, dont certains très connus : Bootstrap, VueJS, etc.

### Comment ça marche ?

Git va conserver la source du projet sur un serveur et chacun va récupérer le projet en local. Lorsque vous allez passer vos modifications sur le serveur, Git va vérifier que vos fichiers modifiés n'ont pas été modifiés par quelqu'un d'autre en parallèle et vous prévenir si c'est le cas.

En cas de « conflit » (si quelqu'un a modifié les mêmes fichiers que vous), Git vous indiquera exactement où le code pose problème.

Git s'utilise aussi très bien pour un projet sur lequel vous êtes seul, pour avoir une version opérationnelle et faire des tests sur une autre version du projet. Notamment sur des projets ambitieux.

### Pour résumer

Git est un outil essentiel de travail, en tant que développeur, nous sommes amenés à utiliser au quotidien. Cet outil permet de :

- Créer des points de sauvegardes de notre travail
- De mettre notre projet en sécurité sur des serveurs distants
- Permet de travailler de manière collaborative

### LE TERMINAL

Il est conseillé de commencer à utiliser git avec le terminal. Tout d'abord nous allons donc apprendre à utiliser le terminal.

Le terminal est en quelque sorte l'ancêtre de programmes comme les systèmes d'exploitation. C'est une application qui vous permet de « dialoguer » avec votre ordinateur avec des lignes de commandes.

Il sert à notamment à :

- naviguer dans une arborescence de fichiers ;
- créer, déplacer et supprimer des dossiers et des fichiers ;
- lancer des programmes.

Pour ouvrir votre terminal, vous pouvez :

- Cliquez sur *Applications > Accessoires > Terminal*.
- Ou écrivez terminal dans la barre de recherche d'application puis sélectionnez-le.
- Ou utilisez le raccourci *Ctrl + Alt + T*.

## NAVIGUER DANS UNE ARBORESCENCE

Pour savoir à quel emplacement vous vous trouvez, il suffit d'utiliser la commande :

```
pwd
```

Pour connaître le contenu du répertoire dans lequel vous êtes ,il suffit d'utiliser la commande :

```
ls
```

Vous pouvez également demander le contenu d'un autre répertoire par exemple :

```
ls monRepertoire
```

**À noter :** le terminal est sensible aux majuscules, attention à bien écrire les noms des répertoires sinon il ne pourront pas être trouvés.

Si un répertoire comporte un espace il faut alors l'écrire entre guillemet, par exemple :

```
ls 'Mon Repertoire'
```

**À noter :** Après avoir tapé 2 ou 3 caractères, il est possible d'appuyer sur la touche **tab** (au-dessus de "verrouillage majuscule") qui permet l'autocomplétion, c'est à dire, de compléter la suite du nom du dossier ou fichier. Cela évite les erreurs de frappe et est très appréciable pour la saisie de nom parfois long et complexe. Peut être utilisé plusieurs fois si plusieurs noms ont une partie commune.

Pour vous déplacer dans un répertoire il faut utiliser la commande `cd` et indiquer le nom du répertoire, par exemple :

```
cd monRepertoire
```

Vous pouvez vous déplacer de plusieurs répertoires en indiquant le chemin, par exemple :

```
cd monRepertoire/sousRepertoire
```

Pour sortir d'un répertoire il suffit d'utiliser :

```
cd ../
```

## CRÉER, DÉPLACER ET SUPPRIMER DES ÉLÉMENTS

Pour créer un répertoire il faut utiliser la commande `mkdir` puis indiquer le nom du nouveau répertoire, par exemple :

```
mkdir nouveauRepertoire
```

Il est aussi possible de créer un fichier, il faudra bien sûr préciser son extension pour savoir quel type de fichier on va créer. Pour cela on va utiliser la commande :

```
touch nouveauFichier.html
```

Pour copier un fichier, il faut utiliser la commande `cp` puis indiquer le nom du fichier à copier et donner l'emplacement de la copie, par exemple :

```
cp nouveauFichier.html nouveauRepertoire/
```

Si vous souhaitez copier un répertoire il faudra ajouter `-r` :

```
cp -r nouveauRepertoire monRepertoire/
```

Pour déplacer un répertoire ou un fichier, il faut utiliser la commande `mv` puis indiquer le nom de l'élément à déplacer et son nouvel emplacement, par exemple :

```
mv nouveauFichier.html nouveauRepertoire/
```

Pour supprimer un fichier, il faut utiliser la commande `rm` puis indiquer le nom du fichier à supprimer, par exemple :

```
rm nouveauFichier.html
```

Si vous souhaitez supprimer un répertoire il faudra ajouter `-r` :

```
rm -r nouveauRepertoire
```

Il existe une infinité de commandes.

**À noter :** Vous pouvez parcourir les dernières commande utilisées avec la touche « flèche du haut » de votre clavier.