



## Git Expert : Aller plus loin dans l'utilisation de Git

Git est l'un des gestionnaires de version les plus évolués à l'heure actuelle, et permet de s'adapter facilement à différents workflows. Une bonne connaissance de l'outil par toute l'équipe et le choix d'un modèle de gestion de version adapté au projet permettront de simplifier la maintenance et d'accélérer le développement du projet en facilitant la collaboration entre les différents développeurs.

Pour tirer parti des nombreuses fonctionnalités de Git, votre équipe doit avoir une pratique solide et commune de l'outil.

Pour cela, nous vous proposons une formation adaptée à votre besoin s'appuyant sur :

- une connaissance avancée du mode d'organisation de nombreux projets Open Source et notamment de distribution Linux
- une expérience du développement sur des projets importants (code / nombre de développeurs),

Nos formateurs sont expérimentés sur Git ainsi que dans la formation et la conception de formations.

### Durée

2 jours (14 heures)

### Objectifs pédagogiques

- Comprendre le fonctionnement interne de Git
- Maîtriser la gestion de fichiers dans un dépôt Git
- Maîtriser les commits et leur qualité
- Optimiser la gestion de branches locales et distantes
- Utiliser les outils avancés de gestion de dépôts Git

### Contact

Anne NICOLAS – [anicolas@ossflow.fr](mailto:anicolas@ossflow.fr)

+33 6 59 11 75 55

<https://ossflow.fr/formation>



## JOUR 1

### Fonctionnement interne de git : objets et branches

- les objets à la base du stockage des modifications
- fonctionnement interne des branches et de leur fusion
- branches locales et branches distantes : le dessous des cartes

### Travailler sur les fichiers : espaces de travail, attributs

- gestion de zones de travail multiples avec git worktree
- les attributs git : définition et utilisation
- fonctionnalités complémentaires : lock de fichiers, diff avancé...

### Travailler sur les commits : patches, notes, historique

- signer les commits
- reflog : le log qui sauve votre projet
- gestion de patches : générer et envoyer des patches, réappliquer les patches
- gestion de notes : ajouter de l'information aux commits, gestion des notes dans le dépôt
- gestion et recherche avancées de l'historique

## JOUR 2

### Travailler sur les branches : historique, merges, reflog, rebase, rebase

- options avancées de la fusion de branches
- usages avancés du rebase
- réécriture avancée de branche avec filter-branch et filter-repo
- enregistrer les résolutions de conflit pour les automatiser avec git rerere
- gérer les références dans la configuration : rebase

### Travailler sur les dépôts : sous-modules, lfs

- les sous-modules : cas d'utilisation, concepts de base et structure du dépôt, gestion des sous-modules
- Optimiser le versioning des fichiers volumineux dans Git avec LFS : configuration du serveur, utilisation côté client