



Formez votre équipe à Git et collaborez plus efficacement !

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 kernel.org
- 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 la philosophie d'un gestionnaire de source décentralisé.
- Comprendre les concepts de Git.
- Être autonome avec la ligne de commande.
- Connaître les workflows et les bonnes pratiques Git en entreprise.
- Chaque point théorique sera accompagné de travaux pratiques pour permettre au stagiaire d'appréhender de façon très concrète l'outil.

Contact

Anne NICOLAS – anicolas@ossflow.fr

+33 (0)7 81 63 46 11

<https://ossflow.fr/formation>



JOUR 1

Introduction

- Les gestionnaires de version existants
- Décentralisation du code
- Git et ses avantages
- Fonctionnalités et commandes de base

Avant de commencer

- Structure d'un dépôt Git
- Les différents objets gérés par git
- Installation et configuration du client Git

Les commandes de base

- Créer un dépôt local
- Ajout et versioning de fichiers
- Supprimer, déplacer et renommer des fichiers
- Analyser l'historique des révisions
- Obtenir des informations sur l'état de la copie de travail
- Ignorer des fichiers

Aller plus loin avec les commandes basiques

- Staging de portions de fichiers
- Filtrage avancé du log
- Affichage de versions antérieures sans checkout
- Utiliser git-reset

Branches, fusion et conflit

- Création et gestion des branches
- Les étapes de la résolution de conflit

JOUR 2

Collaboration, branches et dépôts distants

- Un ou plusieurs dépôts
- La commande git remote
- Auto-tracking de branches distantes
- Gestion des branches distantes
- Gestion de la synchronisation

Bien gérer ses commits

- Les implications de la modification d'historique
- Correction / revert de commits et/ou de merge
- Réorganiser ses commits
- Mettre de côté un travail en cours
- Tagger ses commits

Les principaux workflow

- Les différents workflows git
- gitflow
- gitlab
- github