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 differents 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 developement du projet en facilitant la collaboration entre les differents developeurs.

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

Contact

Anne NICOLAS – anicolas@ossflow.fr
+33 6 59 11 75 55
https://ossflow.fr/formation

SCOP/SARL à capital variable - Capital de 6 000 € - SIRET: 532 262 268 00016
NAF-APE: 6201Z - RCS/RM: Paris 532 262 268 - Num. TVA: FR73532262268

Plan de 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

Travailler sur les fichiers : espaces de travail, attributs

- gestion de zones de travail multiples avec git worktree
- les attibuts git : définition et utilisation

Travailler sur les commits : patches, notes, historique

- 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 dépot
- gestion et recherche avancées de l'historique

JOUR 2

Travailler sur les branches : historique, merges, reflog, refspecs, 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 : refspecs

Travailler sur les dépots : sous-modules, lfs

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