
Programme de Formation

Git : Maîtriser le contrôle de versions et le travail collaboratif

Description

Maîtrisez Git pour versionner efficacement votre code : commits, branches, merges et collaboration en équipe avec GitHub. Formation pratique orientée DevOps.

Points Forts

Formation dynamique alternant théorie et exercices concrets pour comprendre Git, créer et gérer des dépôts, maîtriser les branches et les fusions, collaborer efficacement et intégrer des workflows de travail en équipe.



Public visé

- Développeurs (tous langages)
- Architectes logiciels
- Chefs de projets techniques
- Ingénieurs DevOps
- Consultants IT



Objectifs pédagogiques

- Connaître les principes de fonctionnement d'un gestionnaire de versions distribué
- Découvrir par la pratique la philosophie de Git et ses apports
- Créer et initialiser un dépôt avec Git
- Manipuler les commandes de Git pour gérer les fichiers et les branches
- Mettre en œuvre un projet en mode collaboratif avec Git

Organisation

Durée : 14 heures

Mode d'organisation : Mixte

Contenu pédagogique



Description

Jour 1 : Fondamentaux Git

Placer Git dans les systèmes de gestion de version

- Historique de Git
- Git un système de gestion de version décentralisé
- Principes de fonctionnement de GIT : snapshots vs per file
- Les « états » de GIT (worktree, stage, database)

Installer et configurer Git

- Installation de GIT en fonction des plateformes,
- Configuration initiale,
- Réglages par défaut en fonction des environnements,
- Les attributs Git



Initialiser un dépôt Git local

- Initialiser un dépôt Git local
- Cloner un dépôt Git distant
- Le principe des branches Git

Pratiquer Git au quotidien

- Commit, Revert, Reset : gérer les commits,
- Rebase, Merge : gérer les fusions,
- Stash : éviter des commits inutiles
- RE-RE-RE : REuse REcorded RESolution éviter les conflits de fusion récurrents,
- Optimiser les commandes avec les raccourcis

Jour 2 : Collaboration avancée et workflows

Comprendre et utiliser l'historique Git

- Log : lire l'histoire des commits
- Réécrire les commits
- Reflog : la mémoire de Git
- Bisect Dissect : résoudre un bug
- Tagger

Utiliser le modèle de branches Git Flow

- Initialiser git flow dans un projet,
- Démarrer, terminer une « feature »
- Démarrer, terminer une « release »
- Démarrer, terminer un « hotfix »

Travailler en équipe avec Git

- Créer un dépôt distant sur Github,
- Gestion des utilisateurs,
- Clone ou Fork : cas d'utilisation,
- Fetch, Pull, Push : gérer le dépôt distant,
- Cherry-Pick : éviter des back-merges complets,
- Pull-requests : revue de code et fusion

Utiliser les sous-modules et les sous-arbres

- Cas d'utilisation
- Travailler avec des SubModules
- Travailler avec des Subtrees

Utiliser les hooks pour créer un CI/CD

- Présentation des hooks git,
- Implémenter un CD avec post-receive



Prérequis

- Pratique du développement logiciel (tout langage)
- Connaissances de base en ligne de commande
- Savoir pratiquer Java avec Eclipse est recommandé
- Un questionnaire de positionnement sera réalisé au préalable pour adapter la formation à vos besoins



Modalités pédagogiques

- Exercices concrets
- Cas pratiques
- Quiz d'évaluation des connaissances



Moyens et supports pédagogiques

- Mise à disposition de postes de travail
- Vidéoprojecteur, tableau blanc, supports numériques
- Documentation pédagogique numérique



Modalités d'évaluation et de suivi

- Positionnement en amont de la formation :
 - Un quiz de consolidation des pré-requis sera administré en amont de la formation
- Suivi « pendant » :
 - Feuilles de présence
 - Exercices pratiques
 - Évaluation « fin de formation »
 - Évaluation des acquis en fin de formation
 - Formulaires d'évaluation de la formation
- Évaluation à froid :
 - Suivi post-formation : Questionnaire de satisfaction à j+30