

Programme de formation GIT

Maîtrisez l'utilisation de GIT pour un meilleur contrôle des versions de vos projets

Durée

2 jours / 14 heures

Description

Git est un logiciel de gestion de versions décentralisé qui permet à ses utilisateurs de développer sur leur propre dépôt. Git facilite ensuite la fusion des différents dépôts sur un dépôt global du projet, et fonctionne aussi bien sur des petits que des gros projets.

Cette formation Git, qui associe théorie et pratique, donne une vision très complète de cet outil et permet de comprendre le modèle et l'architecture de ce logiciel de contrôle de versions, tout en apprenant à configurer et utiliser Git dans un contexte de gestion quotidienne des sources d'un projet web. Les outils associés tels que GitLab, GitHub, etc. seront également présentés

Objectifs pédagogiques

- Gérer les fonctionnalités de GIT
- Gérer efficacement les dépôts locaux et distant de GIT
- Réagir rapidement et faire face au maximum de situations d'urgence

Public

Cette formation Git s'adresse à toute personne désirant améliorer sa gestion de version, et accroître sa productivité avec GIT

Pré-requis

Notions de réseaux, utilisation de la ligne de commande et Shell script

Méthodes pédagogiques

50 % théorie / 50 % pratique

Profil intervenant

L'ensemble de nos formations sont animées par des formateurs expérimentés possédant une expérience terrain éprouvée.

Modalités d'évaluation

L'évaluation des acquis se fait tout au long de la session au travers d'ateliers de mise en pratique des notions et concepts abordés pendant la formation.

Programme

Gestion de version et contrôle de source : Pourquoi?

- Historique des systèmes de gestion de version
- Points forts de Git

Concepts et philosophie

- Concepts principaux de Git
- Vocabulaire associé

Configuration

Les bases

- Commit
- La vie des fichiers (modifications, suppressions, déplacement, ...)
- Créer un tag

Retrouver ce que l'on a fait

- Visualisation des actions
- Parcours de l'historique
- Pointer un commit / une révision
- Pointer un intervalle de commit

Les branches

- Définition
- Gestion (création, renommage, suppression, ...)
- Changer de branche
- Fail, j'ai oublié de faire une branche

Les fusions (ou merge)

- Fusion
- Fast-forward
- Git merge
- Les rebases
- Merge vs. Rebase
- Résolution des conflits

Collaboration & dépôts distants

- Git remote
- Protocoles de communication
- Branches distantes
- Git fetch, Git pull, Git push

Nettoyer son historique

- Avantages d'un historique bien propre
- Commiter avec soin
- Réparer un commit

- Supprimer / annuler un commit
- Récupérer un seul commit d'une autre branche

Le déboggage avec Git

- Annotations de fichiers
- Recherche binaire

Gérer l'urgence

- Git stash
- Git worktree

Git sur le serveur

- Protocoles de communication (rappel)
- Installation basique
- Exemple : Création d'un dépôt accessible via SSH
- Clés SSH
- Configuration du serveur SSH + Git
- Autres solutions
- Github
- Gitlab

Aperçu du moteur (Git Internals)

- Comment fonctionne Git de l'intérieur ?
 - Comment automatiser des tâches ?
1. Les objects
 2. Les références
 3. Les Packfiles
 4. Les hooks

Workflow

- Exemples de workflow qui fonctionne sur la plupart des projets
- Pistes pour les adapter à vos projets

Submodules et subtrees

- Submodules
- Modifier un submodule
- Les subtrees
- Subtrees : Modification
- Subtrees : diff