

PROGRAMME DE FORMATION #GIT-1

PRÉSENTATION

Git est un outil indispensable qui permet de travailler en équipe sur un projet versionné. Lorsqu'il est mis en place, sa maîtrise est nécessaire à toute évolution et bon maintien du projet. Venez découvrir comment cette formation répond à ces besoins.

OBJECTIFS DE LA FORMATION

- Comprendre la philosophie Git et l'intérêt du versioning
- Travailler sur un projet en équipe versionné
- Maîtriser Git en lignes de commandes
- Améliorer la résolution de bugs grâce à Git

PROFIL DU STAGIAIRE

- Développeur, Chef de projet
- Novice ou Débutant sur Git
- Connaissance des bases des consoles de commandes
- Maîtriser son poste de travail

CONDITIONS DE RÉALISATION

- Durée : **1 jour**
- Formations Intra-Entreprise dans **toute la France** (demander un [devis personnalisé](#))
- Sessions de **4 à 10** stagiaires
- La formation alterne **théorie et pratique** pour permettre au stagiaire d'en comprendre et maîtriser le contenu tout en permettant d'**évaluer** sa progression
- Nécessite une salle dédiée avec écran de projection
- Un poste par stagiaire avec **droits administrateur** (installation de logiciels, modification de variables d'environnement etc.)
- Accès Internet externe **ouvert** (https, ssh etc.) disponible au formateur et aux stagiaires
- Le formateur doit pouvoir connecter son propre poste de travail au réseau et à l'écran

PROGRAMME DÉTAILLÉ (MATIN)

Introduction

- Le choix du versioning
- Les outils de versioning existants
- Le choix de Git et ses Fonctionnalités

Préparer son environnement

- Installer et configurer Git
- Création d'un dépôt Git local

Comprendre le fonctionnement de Git

- Les différents termes Git
- Structure d'un dépôt Git
- Les différents objets gérés par Git

Apprendre les commandes basiques Git

- Ajouter des fichiers au staging
- Obtenir des informations sur l'état de la copie de travail
- Consulter les modifications en cours
- Supprimer les modifications courantes
- Versionner l'état courant grâce au commit
- Analyser l'historique
- Afficher le contenu d'un commit spécifique
- Mettre de côté un travail en cours

Comprendre le système de branches

- Comprendre le principe de branche
- Apprendre à utiliser les branches

Apprendre les commandes basiques sur les branches

- Créer une branche
- Naviguer entre les branches
- Supprimer une branche
- Renommer une branche

PROGRAMME DÉTAILLÉ (APRÈS-MIDI)

Manipulation avancée des branches par merge et rebase

- Le merge avec fast-forward
- Le merge sans fast-forward
- Le rebase

Gérer des conflits entre branches

- Le cas de conflits simples
- Le cas de conflits complexes impactant plusieurs commits
- Le rebase interactif

Palier aux erreurs humaines fréquentes

- L'oubli de changement de branche sur une seconde console ouverte
- Réparer le dernier commit effectué
- Supprimer les derniers commits
- Fixer plusieurs commits en réécrivant l'historique
- Annuler un ancien commit sans réécriture de l'historique
- Retirer des données sensibles de l'historique

Approfondir les commandes basiques (bonus)

- Ajouter des petites portions de fichiers au staging
- Consulter les modifications ajoutées au staging
- Aller plus loin avec git diff
- Aller plus loin dans l'analyse de l'historique
- Tagger une révision
- Améliorer la configuration d'un projet local

TARIFS & CONTACT

- **4 à 10 stagiaires : 450€ HT** par stagiaire par jour de formation
- Demander un [devis personnalisé](#)
- Formateur(s) : [Alexandre Blondeau](#)
- Contact : alexandre@horus.io