GIT : Gestion des sources avec GIT

Savoir installer, configurer et utiliser GIT dans un contexte de gestion quotidienne des sources d'un projet.

Stage pratique
Qualité du logiciel - Software Craftsmanship

Code :
OGIT

Durée :
2 jour(s) (14,00 heures)

Exposés : **60 %**
Cas pratiques : **40 %**
Echanges d'expérience : **20 %**

Inter-entreprises :
Prochaines sessions disponibles sur notre site web.
Tarif : 1 590,00 € HT / participant

Intra-entreprise :
Tarifs et dates sur demande.

DESCRIPTION

Git est un système de contrôle de versions distribué gratuit et en licence open source. Il a été conçu pour traiter aussi bien des petits ou de très gros projets avec rapidité et efficacité. Git surclasse les autres outils de SCM comme Subversion, CVS, Perforce et ClearCase par sa performance, la taille des dépôts et des fonctionnalités uniques.

À l'issue de cette **formation GIT**, les participants auront appris à configurer et utiliser GIT dans un contexte de gestion quotidienne des sources d'un projet. Ils vont également apprendre à choisir un workflow de collaboration correspondant à leurs projets parmi les différents proposés. Les outils associés et en particulier GitLab seront également présentés au cours de cette formation.

OBJECTIFS PEDAGOGIQUES

À l'issue de cette formation GIT, vous aurez acquis les connaissances et compétences nécessaires pour :

- Connaître le fonctionnement d'un système de contrôle de versions distribué
- Savoir installer et connaître les commandes de bases de Git
- Être capable d'organiser son équipe de développement et un projet autour de Git
- Savoir créer et initialiser un dépôt Git
- Connaître les workflows et les rôles relatifs à la gestion des sources d'un projet
- Comprendre comment gérer et organiser les fichiers et les branches Git

PUBLIC CIBLE

Ce cours GIT s'adresse principalement à des administrateurs système, développeurs, chefs de projets ou architectes souhaitant apprendre à utiliser un système de contrôle de versions distribué pour collaborer autour d'un projet commun. Le projet peut réunir plusieurs collaborateurs d'une même entreprise mais aussi des intervenants ou consultants d'entreprises différentes.

PRE-REQUIS

Bien qu'aucune connaissance technique ne soit strictement attendue, des notions théoriques préalables du cycle de vie du développement d'un logiciel ainsi que les éléments fondateurs d'un gestionnaire de version (commits, branches) faciliteront la compréhension de la logique de GIT.

J'évalue mes connaissances pour vérifier que je dispose des prérequis nécessaires pour profiter pleinement de cette formation en faisant ce test.

METHODE PEDAGOGIQUE

Formation avec apports théoriques, échanges sur les contextes des participants et retours d'expérience pratique des formateurs, complétés de travaux pratiques et de mises en situation.

PROFIL DES INTERVENANTS

Cette formation est dispensée par un·e ou plusieurs consultant·es d'OCTO Technology ou de son réseau de partenaires, expert·es reconnus des sujets traités.

Le processus de sélection de nos formateurs et formatrices est exigeant et repose sur une évaluation rigoureuse leurs capacités techniques, de leur expérience professionnelle et de leurs compétences pédagogiques.

MODALITÉS D'ÉVALUATION ET FORMALISATION À L'ISSUE DE LA FORMATION

L'évaluation des acquis se fait tout au long de la session au travers des ateliers et des mises en pratique. Afin de valider les compétences acquises lors de la formation, un formulaire d'auto-positionnement est envoyé en amont et en aval de celle-ci. Une évaluation à chaud est également effectuée en fin de session pour mesurer la satisfaction des stagiaires et un certificat de réalisation leur est adressé individuellement.

PROGRAMME PEDAGOGIQUE DETAILLE

Jour 1

L'ARCHITECTURE DE TRAVAIL PROPOSÉE PAR GIT ET SON ÉCOSYSTÈME

- Connaitre l'historique des systèmes de versionning Git : de la gestion de versions centralisée à la gestion de versions

décentralisée
- Étudier les concepts de Git (organisation du dépôt, stockage, répertoires)
- Analyser les avantages induits (travail local, intégrité)
- La richesse de l'écosystème Git : les solutions de gestion de serveur, les interactions avec les environnements d'intégration et de déploiement continu

INSTALLER ET METTRE EN PLACE GIT

- Savoir installer Git quel que soit le système d'exploitation à disposition
- Comprendre la configuration de GIT
- Les alias de commande
- Les différentes aides disponibles
- Création d'un dépôt Git

Travaux pratiques

- Installation et configuration d'un client Git en ligne de commande
- Exploration d'un dépôt Git
- Configuration du client Git

APPRENDRE LES COMMANDES DE BASE

- Les trois états d'un fichier
- Vérification du statut des fichiers
- Ajout de fichiers (nouveau ou déjà suivi)
- Ignorer des fichiers
- Valider les changements
- Suppression de fichiers
- Déplacement de fichiers
- Exploitation de l'historique des commits
- Assimiler les commandes d'annulation
- Reflog : retrouver des informations pour annuler une opération

Travaux pratiques

- Production de commits dans le cadre des bonnes pratiques
- Interrogation de l'historique en mettant en place des filtres de recherche complexes sur un dépôt existant
- Manipulation des commandes pour récupérer un fichier supprimé, revenir à un état antérieur, corriger le contenu d'un ou plusieurs commits
- Utilisation de vscode pour comprendre le mode graphique de

Git

Jour 2

GIT ET LES BRANCHES

- Illustration des branches sur un exemple
- La définition des branches Git, de simples pointeurs
- Comprendre le changement de branche et les précautions à prendre
- La suppression de branche : impact sur le projet
- La fusion de branches : les différents cas en présence
- Outils de résolution de conflit
- L'opération de rebase, différence avec la fusion

Travaux pratiques

- Création de branches et tests de suppression puis recréation
- Travail sur la fusion de branches : travail sur un dépôt existant pour régler les conflits proposés, annulation d'un merge
- Réutilisation du même dépôt pour appliquer un rebase, résolution de conflit

SERVEURS GIT, EXEMPLE DE GITLAB

- La solution GitLab, apports
- Mise en place d'un dépôt nu
- Les différents protocoles d'accès
- Accès SSH avec utilisateur unique, échanges de clé privé/publique
- Exploiter les outils d'information et de modification d'un dépôt avec GitLab
- Modèle de données GitLab, rôle et gestion des droits et administration
- Configuration du client

Travaux pratiques

- Mise en place d'un dépôt distant sur le serveur école
- Échanges de clés SSH
- Clône de dépôt

ORGANISER UN PROJET COLLABORATIF AUTOUR DE GIT

- Gérer les interactions avec le serveur : le travail collaboratif
- Les dépôts distants, commandes de gestion
- Présentation des commandes fondamentales : pull,push et fetch
- Les branches distantes, création, synchronisation et suivi
- Les workflows de branches
- Définition d'un workflow et importance pour un projet GITGitflow : gestion de release longue
- Présentation de Github : gestion de fonctionnalités
- Gitlab : branches d'environnement et branches de développement
- Workflow avec intégrateur : la merge request (ou pull request)

Travaux pratiques

- Création d'un projet d'équipe sur GitLab
- Mise en place des droits à partir du workflow choisi
- Création et proposition de branche via le mécanisme de la merge request
- Proposition de revue de code et fusion

Accessibilité

L'inclusion est sujet important pour OCTO Academy.
Nos référent·es sont à votre disposition pour faciliter l'adaptation de votre formation à vos besoins spécifiques.
Pour les contacter : academy.accessibilite@octo.com