

Plan de cours N° : 1156

Durée : 2 jours (14h)

PARTICIPANTS / PRE-REQUIS

Tout développeur, chef de projet, architecte, souhaitant utiliser Git comme gestionnaire de versions.

Connaissance des processus de développement et d'un langage de programmation, et des bases Unix/Linux. Les travaux pratiques se déroulent sur Linux.

OBJECTIFS PEDAGOGIQUES

Comprendre les principes d'un gestionnaire de version distribué, les apports de Git. Savoir le mettre en oeuvre pour gérer les codes sources d'un projet, les versions, corrections de bugs...

MOYENS PEDAGOGIQUES

Réflexion de groupe et apports théoriques du formateur

Travail d'échange avec les participants sous forme de réunion-discussion

Utilisation de cas concrets issus de l'expérience professionnelle

Validation des acquis par des exercices de synthèse

Alternance entre apports théoriques et exercices pratiques (en moyenne 30 et 70%)

Remise d'un support de cours.

Assistance post-formation d'une durée de 1 an sur le contenu de la formation via notre adresse mail dédiée formateurs@atp-formation.com

MOYENS PERMETTANT LE SUIVI DE L'EXECUTION ET DES RESULTATS

Feuille de présence signée en demi-journée,

Evaluation des acquis tout au long de la formation,

Questionnaire de satisfaction,

Attestation de stage à chaque apprenant,

Positionnement préalable oral ou écrit,

Evaluation formative tout au long de la formation,

Evaluation sommative faite par le formateur ou à l'aide des certifications disponibles.

MOYENS TECHNIQUES EN PRESENTIEL

Accueil des stagiaires dans une salle dédiée à la formation, équipée d'ordinateurs, d'un vidéo projecteur d'un tableau blanc.

MOYENS TECHNIQUES DES CLASSES A DISTANCE

A l'aide d'un logiciel comme Teams, Zoom etc... un micro et éventuellement une caméra pour l'apprenant, suivez une formation en temps réel et entièrement à distance. Lors de la classe en ligne, les apprenants interagissent et communiquent entre eux et avec le formateur.

Les formations en distanciel sont organisées en Inter-Entreprise comme en Intra-Entreprise. L'accès à l'environnement d'apprentissage (support de cours, labs) ainsi qu'aux preuves de suivi et d'assiduité (émargement, évaluation) est assuré.

Les participants recevront une convocation avec lien de connexion

Pour toute question avant et pendant le parcours, une assistance technique et pédagogique est à disposition par mail et par téléphone auprès de notre équipe par téléphone au 04.76.41.14.20 ou par mail à contact@atp-formation.com

ORGANISATION

Les cours ont lieu de 9h00-12h30 13h30-17h00.

PROFIL FORMATEUR

Nos formateurs sont des experts dans leurs domaines d'intervention

Leur expérience de terrain et leurs qualités pédagogiques constituent un gage de qualité.

ACCESSIBILITE

Les personnes atteintes de handicap souhaitant suivre cette formation sont invitées à nous contacter directement, afin d'étudier ensemble les possibilités de suivre la formation.

MISE A JOUR

01/01/2021

Plan de cours N° : 1156

Durée : 2 jours (14h)

Présentation de Git

- La notion de gestionnaire de versions distribué
- Historique de Git, licence.
- Présentation des principes techniques de Git
Architecture, les objets stockés
- Les différentes utilisations de Git
Utilisation d'applicatifs stockés sous Git
Développement, partage de codes
Gestions de modifications, de correctifs
- Aperçu des types de workflows possibles

Prise en main

- La commande Git
Options principales
- Installation et configuration de Git
- Présentation des notions de base
Référentiel, index, répertoire de travail, clonage
- Travaux pratiques
Création d'un premier dépôt
Utilisation de la ligne de commande pour les opérations de base
Enregistrement de modifications simples
Clonage d'un référentiel existant

Gestion des développements

- Etude des commandes principales de manipulation des fichiers
add, status, diff, commit...
- Gestion des branches
branch, checkout, merge, log, stash...
- Travaux pratiques
Mise en oeuvre sur un projet exemple représentatif des principaux cas d'utilisation
- Ajout, modification, suppression de fichiers et répertoires
- Gestion des commits
- Création de branches
Navigation entre branches, fusion de branches
- Résolution des conflits
- Intérêt des branches temporaires

Travail collaboratif

- Objectif
Partage et mise à jour de projets
- Fonctionnalités requises
Mise à disposition des objets
Analyse des modifications, intégration...
- Définition des rôles
Développeurs, intégrateurs
- Notion de dépôt local et dépôt centralisé
- Etude des commandes
fetch, pull, push, remote...
- Pour le contrôle de fichiers
show, log, diff...
- Gestion des patches
apply, rebase, revert...
- Travaux pratiques
Connexion à un référentiel
- Synchronisation avec un référentiel distant
Utilisation des tags pour identifier des commits
Création et application de patches sur un exemple de projet complet

Administration

- Tâches d'administration
Nettoyage des arborescences
Vérification de la cohérence de la base de données
Etat du service Git
- Travaux pratiques
Installation d'un dépôt privé centralisé pour une gestion de sources collaborative
Import de développements externes avec fast-import

Compléments

- Interagir avec des référentiels partagés via GitHub
Exemples de projets sur GitHub, GitLab
- Présentation d'outils complémentaires
Gerrit, un système de revue de code
Gitweb, l'interface web
GitKraken, client graphique

Bonnes pratiques

- Echanges par rapport aux contextes projets et à l'organisation des équipes pour savoir définir l'utilisation de Git la plus adaptée à chaque contexte projet