Pythagore F.D.

Apprendre à apprendre

UX151

Gestion de versions avec GIT

Durée: 2 jours

1140 €	15 au 16 mars	20 au 21 septembre
	14 au 15 juin	22 au 23 novembre

Public:

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

Objectifs:

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, etc ..

Connaissances préalables nécessaires:

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

Programme:

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 etc ...
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, clônage
Travaux pratiques :
Création d'un premier dépôt.
Utilisation de la ligne de commande pour les opérations de base.
Enregistrement de modifications simples.
Clônage d'un référentiel existant.

Pythagore F.D.

Apprendre à apprendre

UX151

Gestion de versions avec GIT

Gestion des développements : Etude des commandes principales de manipulation des fichiers :
add, status, diff, commit, ...
Gestion des branches :
branch, checkout, merge, log, stash, etc ...
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,etc...
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 patchs : 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 patchs sur un exemple de projet complet.

Administration : Tâches d'administration : nettoyage des arborescences, vérification de la cohérence de la base de données,
état du service git
Travaux pratiques :
Installation d'un dépôt privé centralisé pour une gestion de sources collaborative
import de développements externes avec fast-import

Pythagore F.D.

Apprendre à apprendre

UX151

Gestion de versions avec GIT

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 officielle
GitKraken, client graphique

Bonnes pratiques : Echanges par rapport aux contextes projets et à l'organistion des équipes pour savoir définir l'utilisation de git la plus adaptée à chaque contexte projet.