

GIT

Description de la formation GIT

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 , GitHub seront également présentés au cours de cette formation.

Objectifs

Objectif opérationnel :

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

Objectifs pédagogiques :

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

À qui s'adresse cette formation ?

Public :

Ce cours GIT s'adresse à des développeurs, chefs de projet et architectes désireux de comprendre GIT.

Prérequis :

Aucun pré-requis n'est nécessaire pour suivre cette formation, cependant la connaissance d'un SCM (Définition Supply Chain Management

Le Supply Chain Management définit l'ensemble des ressources, moyens, méthodes, outils et techniques destiné à piloter le plus efficacement possible la chaîne globale d'approvisionnement depuis le premier fournisseur jusqu'au client final.) Est un plus.

Contenu du cours GIT

S'initier à GIT

- Connaître l'historique des systèmes de versionning Git : un système de versionning distribué

- Etudier les concepts de Git (organisation du repository, stockage, répertoires)
- Analyser les avantages induits (travail local, intégrité)
- Panorama des offres autour de l'outil
- Installer et mettre en place le GIT
- Savoir installer GIT à partir des sources
- Savoir installer GIT à partir des binaires
- Comprendre la configuration et mise en place de GIT

Les différentes aides disponibles Création d'un repository Git

Travaux Pratiques

Installation et configuration d'un GIT local

Apprendre les commandes de base

Rappel sur les 3 états d'un fichier

Vérification du statuts des fichiers

Ajout de fichiers (nouveau ou déjà suivi)

Ignorer des fichiers

Valider les changements

Suppression de fichiers

Déplacement de fichiers

Accès à l'historique

Assimiler les commandes d'annulation

Auto-complétion et Alias

Travaux Pratiques

Manipulation de sources d'un projet

Git et les branches

Illustration des branches sur un exemple

Les branches Git, de simples pointeurs, différence avec les tags

Commandes de gestion de branche et de basculement

La fusion de branche

Outils de résolution de conflit

L'opération de rebase, différence avec la fusion

Travaux Pratiques

Création de branches fusion et rebasement, gestion de conflit

Serveurs Git, exemple de Gitlab

Mise en place d'un dépôt nu

Les différents protocoles d'accès

Accès SSH avec utilisateur unique, échanges de clé privé/publique

La solution Gitlab, apports

Modèle de données GitLab, rôle et gestion des droits et administration

Mise en place de dépôt

Configuration du client

Travaux Pratiques

Installation Gitlab, Mise en place d'un dépôt distant, Echanges de clés SSH, Clone de dépôt

Workflow de collaboration

Introduction et pattern référencé

Les dépôts distants, commandes de gestion Workflow centralisé, présentation des commandes pull,push et fetch

Les branches distantes, création, synchronisation et suivi

Le pattern Gitflow et son application

Workflow avec intégrateur et son application, le pull request

Les workflows proposés par Gitlab, le merge request

Travaux Pratiques

Travail en équipe, jouer un workflow centralisé et un workflow Gitlab avec revue de code et merge request

Pour aller plus loin sur GIT

Le reflog et les commandes avancées

Personnalisation de Git avec les hooks

Migration SVN vers Git

Travaux Pratiques

Ajout et Rebase interactif, Mise en place de hook client et serveur (via Gitlab)

Avis des participants