

# GIT - La gestion de versions décentralisée

Référence : R8-0015-02



OUTILS



Vos projets impliquent des développements répartis géographiquement, ou des intervenants nomades. Issu du monde du développement des logiciels libres, GIT a été pensé dès le début pour fonctionner de façon décentralisée, c'est l'une des clefs de son succès.

## OBJECTIFS

Cette formation vous permettra d'acquérir les compétences nécessaires pour configurer et utiliser GIT au quotidien pour gérer votre code source. A l'issue de la formation, vous serez en mesure de :

- ★ Comprendre le fonctionnement de GIT.
- ★ Suivre les modifications de vos projets.
- ★ Utiliser le mode distribué avec un serveur.
- ★ Adapter GIT à la gestion de votre contexte projet.
- ★ Intégrer GIT à votre environnement de développement.

## SOMMAIRE

- Introduction
- Les bases
- Mode distribué avec serveur
- L'utilisation des branches
- Flux de développement
- Utilisation des sous modules
- Conclusion



Publics



Développeurs,  
adminsitrateurs.

Durée



1 jour.  
40% de pratique,  
60% de cours.

Pré-requis



Avoir déjà utilisé un  
gestionnaire de versions.  
Commandes Unix.

Taille du groupe : ≤ 10 pers.



Formation





## CONTENU PEDAGOGIQUE

### Introduction

- Qu'est-ce qu'un logiciel de gestion de versions ?
- Différences entre logiciel centralisé et distribué
  - version centralisée
  - version distribuée
  - version distribuée avec serveur
- Généralités

### Principes de GIT

- Sécurité
- Etats des fichiers

### Installation

- Différence entre OS ?
- Configuration
- Kit de survie

### Git au quotidien

- Workflow d'utilisation
- Créer un dépôt
- Formatage des commits
- Cloner un dépôt
- Ignorer des fichiers
- Historique des fichiers
- Retour en arrière

### Mode distribué avec serveur

- Protocoles de communication
- Ajouter un remote
- Récupérer des données
- Process de mise à jour
- Reverser sur le serveur
- Tagger une version

### Les branches

- Récupération et création d'une branche
- Passage d'une branche à l'autre
- Différence Merge et Rebase

### GIT sur un projet

- Présentation de GIT Flow

### Utilisation de sous-module

- Initialiser un sous-module
- Cloner un projet avec sous-module
- Mettre à jour un projet avec sous-module
- Les erreurs à éviter

### D'autres commandes

- Reflog
- Stash
- Revert

### Conclusion

- Synthèse
- Liens utiles