

Mis à jour le 29/09/2025

S'inscrire

Formation Jujutsu

3 jours (21 heures)

Formation Jujutsu

Notre formation vous présente Jujutsu (jj), un nouveau système de gestion de version pensé pour dépasser les limites de Git. Basé sur le concept de changesets mutables, il permet de réécrire et restructurer l'historique avec plus de souplesse, tout en restant interopérable avec Git.

Conçu pour les équipes modernes, il favorise la collaboration, la gestion des branches implicites et l'intégration fluide dans les pipelines CI/CD.

Cette formation Jujutsu vous permettra de maîtriser ce nouvel outil, d'adopter des workflows plus efficaces que Git, et de l'intégrer dans vos projets DevOps.

Vous apprendrez à gérer les changesets, collaborer en équipe, sécuriser vos processus CI/CD et piloter vos projets de développement avec une meilleure visibilité.

À l'issue de cette formation, vous serez capable de déployer Jujutsu dans vos environnements, de former vos équipes et d'améliorer vos pratiques de versionning.

Comme toutes nos formations, celle-ci repose sur la dernière version stable de [Jujutsu](#) et ses nouveautés.

Objectifs

- Comprendre les concepts fondamentaux de Jujutsu et des changesets.
- Migrer et interopérer avec des dépôts Git existants.
- Intégrer Jujutsu dans les workflows DevOps et CI/CD.
- Gérer efficacement l'historique, les branches et les merges.
- Piloter des projets agiles avec Jujutsu comme moteur de versionning.

Public visé

- Ingénieurs DevOps
- Développeurs backend et fullstack
- Chefs de projet techniques

Pré-requis

- Connaissance des bases de Git et du contrôle de version
- Notions de CI/CD et pratiques DevOps

Programme de formation Jujutsu

[Jour 1 - Matin]

Introduction à Jujutsu et aux systèmes de contrôle de version (VCS)

- Pourquoi Jujutsu : limites de Git et approche change-centric
- Concepts clés : changesets, historique mutable, interopérabilité Git
- Installation et configuration initiale de jj
- Tour des commandes de base : jj init, jj status, jj commit
- Import/export avec des dépôts Git
- Atelier pratique : Créer un dépôt jj et gérer ses premiers changements.

[Jour 1 - Après-midi]

Gestion de l'historique et des changesets

- Commits Git vs changesets jj
- Réécriture fluide : amend, reword, réorganisation
- Multiples heads et navigation dans l'historique (jj log)
- Restauration et rollback en toute sécurité
- Bonnes pratiques de description des changements
- Atelier pratique : Restructurer un historique complexe.

Branches, merges et collaboration avec Git

- Branches implicites jj : principes et implications
- Synchronisation jj - Git
- Merge et rebase intelligents, résolution de conflits
- Workflows GitFlow vs approches jj
- Migration incrémentale depuis Git
- Atelier pratique : Migrer un workflow Git existant vers jj.

[Jour 2 - Matin]

Flux collaboratifs et revues

- Publication et partage des changesets
- Stratégies pour équipes distribuées
- Résolution de conflits interactive
- Relecture, validation et intégration continue
- Traçabilité et audit des opérations
- Atelier pratique : Collaboration multi-développeurs simulée.

[Jour 2 - Après-midi]

Intégration CI/CD

- Jenkins / GitLab CI / GitHub Actions : déclencher sur changesets
- Gestion artefacts et versions avec jj
- Tests de performance dans le pipeline
- Stratégies blue/green et canary supervisées
- Reporting et qualité continue
- Atelier pratique : Pipeline CI/CD basé sur jj.

Pilotage projet

- Gouvernance des branches et suivi des livrables
- Indicateurs de progression et visibilité sur l'historique
- Documentation et auditabilité
- Alignement Agile & DevOps
- Cadences, rituels et synchronisation
- Atelier pratique : Construire un workflow projet avec jj.

[Jour 3 - Matin]

Personnalisation et cas avancés

- Aliases, scripts et productivité
- Intégrations IDE, Docker, Kubernetes
- Squash, amend, rebase multi-changesets
- Sauvegarde et restauration de dépôts
- Cas d'usage open source et entreprise
- Atelier pratique : Personnaliser jj pour l'environnement cible.

[Jour 3 - Après-midi]

Optimisation et bonnes pratiques

- Stratégies de branches efficaces avec jj
- Gardez un dépôt clair et cohérent
- Performance sur gros dépôts
- Monorepo : comparaisons Git vs jj
- Pièges à éviter et checklists
- Atelier pratique : Audit et optimisation d'un workflow DevOps.

Mise en production et adoption

- Déploiement de jj dans une équipe existante
- Migration progressive depuis Git
- Retours d'expérience d'early adopters
- Intégration avec SRE et DevSecOps
- Feuille de route et conduite du changement
- Atelier pratique : Plan de migration d'équipe.

Sociétés concernées

Cette formation s'adresse à la fois aux particuliers ainsi qu'aux entreprises, petites ou grandes, souhaitant former ses équipes à une nouvelle technologie informatique avancée ou bien à acquérir des connaissances métiers spécifiques ou des méthodes modernes.

Positionnement à l'entrée en formation

Le positionnement à l'entrée en formation respecte les critères qualité Qualiopi. Dès son inscription définitive, l'apprenant reçoit un questionnaire d'auto-évaluation nous permettant d'apprécier son niveau estimé sur différents types de technologies, ses attentes et objectifs personnels quant à la formation à venir, dans les limites imposées par le format sélectionné. Ce questionnaire nous permet également d'anticiper certaines difficultés de connexion ou de sécurité interne en entreprise (intraentreprise ou classe virtuelle) qui pourraient être problématiques pour le suivi et le bon déroulement de la session de formation.

Méthodes pédagogiques

Stage Pratique : 60% Pratique, 40% Théorie. Support de la formation distribué au format numérique à tous les participants.

Organisation

Le cours alterne les apports théoriques du formateur soutenus par des exemples et des séances de réflexions, et de travail en groupe.

Validation

À la fin de la session, un questionnaire à choix multiples permet de vérifier l'acquisition correcte des compétences.

Sanction

Une attestation sera remise à chaque stagiaire qui aura suivi la totalité de la formation.

