

Mis à jour le 12/01/2026

S'inscrire

## Formation JetBrains WebStorm

3 jours (21 heures)

### Présentation

JetBrains WebStorm est un IDE JavaScript/TypeScript conçu pour accélérer le développement front-end et Node.js grâce à l'auto-complétion intelligente, la navigation de code et le debug intégré. Idéal pour fiabiliser un projet React/Vue/Angular, standardiser un workflow d'équipe et gagner du temps sur la qualité et les tests.

Cette formation vise à rendre WebStorm immédiatement opérationnel dans vos projets : configuration de l'environnement, compréhension des inspections, refactorings, gestion des dépendances et intégration des outils modernes (linters, formatters, tests). L'objectif est de produire du code plus lisible, plus sûr et plus rapide à maintenir.

L'approche est 100% pratique : démonstrations guidées, ateliers sur un projet existant, exercices de debug et de refactoring. Vous repartez avec une configuration WebStorm reproductible (paramètres, raccourcis, templates), un workflow Git intégré et une checklist d'outillage (ESLint/Prettier/tests) adaptée à votre stack.

### Objectifs

- Configurer WebStorm pour un projet JavaScript/TypeScript.
- Naviguer efficacement dans une base de code et ses dépendances.
- Déboguer une application front-end et Node.js avec breakpoints.
- Appliquer des refactorings sûrs et automatiser la qualité (ESLint/Prettier).
- Industrialiser le workflow avec Git, tests et tâches npm.

### Public visé

- Développeurs front-end (React, Vue, Angular).
- Développeurs Node.js / full-stack.
- Tech leads souhaitant standardiser l'outilage.

### Pré-requis

- Bonnes bases en JavaScript et notions de TypeScript.
- Connaissance d'un gestionnaire de paquets (npm ou yarn).
- Notions de Git (commit, branch, merge).
- Compréhension des scripts de build/test (npm scripts).

## Pré-requis techniques

- WebStorm installé (dernière version stable recommandée).
- OS : Windows, macOS ou Linux.
- RAM : 8 Go minimum, 16 Go recommandés.
- Node.js LTS installé + accès à un terminal.
- Accès à un dépôt Git de projet (local ou interne) pour les ateliers.

## Programme de formation JetBrains WebStorm

[Jour 1 - Matin]

### Prise en main de WebStorm et configuration de l'environnement

- Installer WebStorm et activer la licence (JetBrains Account, offline si besoin)
- Configurer le projet : Node.js, npm/yarn/pnpm, TypeScript, ESLint/Prettier
- Tour d'horizon de l'interface : Project, Editor, Terminal, Tool Windows
- Productivité : navigation, recherche, refactorings de base, Live Templates
- Atelier pratique : Créer un projet JavaScript/TypeScript et appliquer une configuration ESLint + Prettier.

[Jour 1 - Après-midi]

### Édition intelligente, inspections et refactoring

- Comprendre les inspections : erreurs, warnings, quick-fixes et intentions
- Refactorings clés : Rename, Extract (variable/fonction), Move, Inline, Safe Delete
- Auto-imports, gestion des modules, navigation entre symboles et usages
- Documentation et lecture de code : types, JSDoc/TS, structure et breadcrumbs
- Atelier pratique : Refactorer un module existant (extraction de fonctions, renommage, réorganisation) en s'appuyant sur les inspections.

[Jour 2 - Matin]

### Debug et exécution : Node.js, scripts et configurations

- Run/Debug Configurations : Node, npm scripts, TypeScript (ts-node), env vars
- Debugger : breakpoints, watches, call stack, step over/into/out
- Debug asynchrone : promises, async/await, gestion des exceptions
- Outils intégrés : Terminal, HTTP Client, logs et console
- Atelier pratique : Déboguer une API Node/Express avec points d'arrêt, variables d'environnement et inspection d'objets.

[Jour 2 - Après-midi]

## Tests, qualité et automatisation dans WebStorm

- Configurer et exécuter les tests : Jest, Vitest, Mocha (run, debug, watch)
- Couverture de code et analyse des résultats de tests
- Intégration ESLint/Prettier : corrections rapides, formatage à la sauvegarde
- Tasks et scripts : npm/yarn/pnpm, macros, actions et raccourcis personnalisés
- Atelier pratique : Mettre en place une suite de tests + couverture et corriger les problèmes de linting via quick-fixes.

[Jour 3 - Matin]

## Git et collaboration : workflow efficace dans l'IDE

- Configurer Git : remotes, branches, .gitignore, gestion des line endings
- Workflow quotidien : commit, amend, stash, rebase interactif, cherry-pick
- Résolution de conflits avec merge tool et comparaison avancée
- Code review : diff, annotations, historique, recherche dans l'historique
- Atelier pratique : Réaliser une feature en branche, rebase interactif, résolution de conflits et préparation d'un historique propre.

[Jour 3 - Après-midi]

## Optimisation de la productivité et personnalisation avancée

- Keymap et actions : navigation avancée, multi-cursors, structural search/replace
- Templates et génération : Live Templates, File Templates, postfix completion
- Gestion des performances : indexation, exclusions, caches, gros monorepos
- Plugins utiles et réglages d'équipe : partage de configuration, EditorConfig
- Atelier pratique : Construire un “setup d'équipe” (templates, keymap, EditorConfig, inspections) et l'appliquer sur un mini-projet.

## 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.