+33 (0)1 48 10 09 46
formation@ambient-it.net
140 Avenue Jean Lolive 93500 Pantin - France

Mis à jour le 06/06/2025

S'inscrire

Formation JavaScript : Programmation avancée

3 jours (21 heures)

Présentation

Découvrez notre formation « JavaScript, programmation avancée », un parcours conçu spécifiquement pour maîtriser les dernières évolutions du langage. Parfaitement adaptée aux développeurs expérimentés, elle vous offre une maîtrise approfondie du langage pour des applications web riches, robustes et sécurisées de ES6 à ES2025.

Apprenez à écrire du code clair et performant en manipulant les fonctions avancées du langage : closures, générateurs, classes modernes, générateurs, modules et gestion fine des contextes d'exécution. Expérimentez directement grâce à des ateliers pratiques et réalistes tout au long de votre apprentissage. Vous approfondirez particulièrement la gestion du code asynchrone, maîtrisant promesses, async/await, Fetch API, et les techniques avancées de gestion d'événements. Vous explorerez les Web Components natifs, l'architecture modulaire, ainsi que des micro-frameworks JavaScript légers comme Alpine.js ou Lit, afin de sélectionner la technologie la plus pertinente selon chaque contexte. Renforcez la qualité de votre code avec les outils les plus efficaces du marché (ESLint, Jest, Prettier), tout en assurant une sécurité optimale face aux vulnérabilités courantes (XSS, CSP). Vous maîtriserez également les meilleures pratiques de sécurité pour protéger efficacement votre site. Enfin, préparez-vous dès maintenant aux futures évolutions du langage, telles que le Pattern Matching ou l'immutabilité native, pour rester constamment à la pointe de JavaScript.

Cette formation JavaScript de perfectionnement vous présentera la dernière version du langage : JavaScript 15 – ES2024.

Objectifs

- Maîtriser les fonctionnalités avancées de JavaScript ES6
- Savoir construire une application utilisant les features avancées de JavaScript
- Parcourir et modifier le DOM avec la bibliothèque JQuery
- Gérer les API JavaScript HTML5
- Utiliser JavaScript pour différents cas d'usage : applications web, navigateurs, logiciels, serveurs, etc.
- Appréhender le développement côté serveur avec Nodejs
- Maîtriser le débogage
- Comprendre en profondeur la Programmation Orientée Objet

Public visé

- Développeurs front-end
- Architectes techniques
- UI designer
- Chefs de projets
- Ingénieur d'études

Pré-requis

- Avoir suivi notre formation initiale sur JavaScript
- Connaissance de HTML et CSS
- Connaissance de base de XML

Programme de notre formation JavaScript : Programmation avancée

Introduction au JavaScript moderne avec ES6+

- Compatibilité des navigateurs et des runtimes
- Comment choisir la bonne version au bon moment ?
- Les outils à avoir pour bien démarrer (VS Code, DevTools, npm, yarn, Webpack, etc.)
- Variables modernes (let, const) – portée de bloc
- Fonctions fléchées (Arrow functions)
- Déstructuration (objets/tableaux)
- Opérateurs rest/spread (...) sur tableaux et objets
- Nouveaux opérateurs de sécurité (?., ??)
- Template literals (chaînes interpolées)
- Composition avec Mixins (réutilisabilité, modularité)
- Immutabilité (principes et bonnes pratiques)
- Atelier pratique : Installez votre environnement JavaScript moderne (VS Code, npm, DevTools), puis créez un script simple intégrant les nouvelles syntaxes (déstructuration, opérateurs avancés, fonctions fléchées, etc.) afin de manipuler efficacement des données structurées.

Évolutions ECMAScript (ES6 ? ES2024)

- ES6 (2015) : Classes, Modules, let/const, Arrow functions
- ES7 (2016) : Exponentiation, Array.includes()
- ES8 (2017) : Async/Await, Object.values()
- ES9 (2018) : Rest/Spread objets, Regex améliorés
- ES10 (2019) : Array.flat(), Object.fromEntries()
- ES11 (2020) : Optional chaining (?.), Nullish coalescing (??)
- ES12 (2021) : Promise.any(), Séparateurs numériques (_)
- ES13 (2022) : Champs privés (#), Méthodes privées
- ES14 (2023) : findLast(), améliorations Regex
- ES15 (2024) : Ergonomie, précision des types, optimisation performance

- Le futur ES2025 : pattern matching, structures immuables (Records/Tuples)
- Atelier pratique : Implémentez aux choix 3 nouvelles fonctionnalités d'ES6+.

Fonctions avancées en JavaScript

- Closures (fonctions fermées sur un contexte lexical)
- Contexte d’exécution (this) et méthodes call/apply/bind
- Fonctions fléchées vs fonctions classiques (liaison du this)
- Fonctions anonymes immédiates (IIFE)
- Générateurs (fonction* avec yield) et itérateurs personnalisés
- Atelier pratique : Implémenter une fonction à closure permettant de créer un compteur à valeur privée (mémoire d’état interne).

Programmation orientée objet en JS

- Prototypes et héritage prototypal
- Fonctions constructrices et mot-clé new
- Classes ES6 (syntaxe class, héritage avec extends, champs privés #)
- Propriétés et méthodes d’instance vs statiques
- Atelier pratique : Concevoir une petite hiérarchie d’objets (classe et sous-classe) pour modéliser un cas concret mettant en œuvre un héritage.

Architecture & patterns JS

- Modularité et SOLID en JS
- Patterns Observateur, Factory, Singleton
- Injection de dépendances
- Maintenabilité et évolutivité
- Atelier pratique : Implémenter le pattern Observateur pour notifier les composants.

Modules et organisation du code

- Modules ES6 (import/export)
- Bundler et transpilation (Webpack, Babel)
- Gestion des dépendances avec npm (fichier package.json, scripts)
- Atelier pratique : Refactoriser une application existante en modules ES6 et la construire à l’aide d’un bundler (ex : Webpack) pour optimiser le chargement.

JavaScript Avancé asynchrone

- Gestion de l’asynchrone et boucle événementielle (event loop)
- Modèle d’exécution : event loop, stack, task & micro-task queues
- Rappels de fonctions (callbacks) et inversion de contrôle
- Promesses JavaScript (objets Promise, chaînes then/catch)
- Syntaxe moderne async/await pour les promesses
- Promises : création, chaînage, gestion d’erreurs, Promise.allSettled

- async / await : parallélisme, annulation avec AbortController
- Itérateurs / générateurs asynchrones (for-await-of)
- API Web asynchrones : fetch, Streams, Web Workers
- Appels HTTP avec l'API Fetch (requêtes AJAX modernes)
- Atelier pratique : Consommer des données depuis une API externe en utilisant fetch et async/await afin d'afficher des résultats dynamiques dans la page.

Composants Web natifs (Web Components)

- Éléments HTML personnalisés (API Custom Elements)
- Shadow DOM (encapsulation du DOM et des styles)
- Templates HTML et slots pour le contenu interne des composants
- Cycle de vie des Web Components (méthodes de callback comme connectedCallback, etc.)
- Atelier pratique : Créer un composant Web personnalisé (Custom Element) avec Shadow DOM pour encapsuler un widget réutilisable et stylé de manière isolée.

Qualité du code et tests avancés

- Analyse statique avec ESLint (linting et règles de codage)
- Formatage de code automatique avec Prettier
- Tests unitaires automatisés avec Jest
- Atelier pratique : Mettre en place ESLint et Prettier sur un projet existant, puis écrire et exécuter un test unitaire simple avec Jest pour valider une fonction.

Tour d'horizon des Micro-frameworks JavaScript légers

- Alpine.js (framework minimaliste s'intégrant via de simples attributs HTML)
- Lit (bibliothèque légère pour construire des Web Components réactifs)
- Svelte (framework compilé, sans runtime lourd côté navigateur)
- Atelier pratique : Expérimenter un de ces micro-frameworks en implémentant une fonctionnalité simple (par exemple un compteur interactif) afin de comparer leur approche à celle du JS natif (vanilla).

Sécurité et bonnes pratiques

- Failles XSS et injections côté client (sécurité du code navigateur)
- Politique de sécurité de contenu (Content Security Policy)
- Compatibilité des versions ECMAScript (cibles navigateurs et transpilation avec Babel)
- Choix technologiques adaptés (JavaScript natif vs frameworks selon les besoins du projet)
- Atelier pratique : Analyser une base de code existante pour corriger les failles XSS potentielles et appliquer les bonnes pratiques de sécurité tout en adaptant le code aux contraintes de compatibilité.

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.

Organisme de formation enregistré sous le numéro 11 75 54743 75. Cet enregistrement ne vaut pas agrément de l'État.
© Ambient IT 2015-2025. Tous droits réservés. Paris, France - Suisse - Belgique - Luxembourg