

Formation JavaScript avancé

Description de la formation javascript avancé

Avec l'essor des interfaces riches et les nombreuses nouveautés JavaScript apportées par le HTML5, qui permettent désormais de faire jusqu'à une application complète, les besoins en JavaScript sont devenus aussi importants que ceux des langages serveur (PHP, Java, etc.) étant donné l'animation qui règne autour de Javascript (front end, back end, full stack, frameworks, mobile, etc.).

Il n'est donc plus possible d'avoir une approche simplement basique de JavaScript et c'est un langage qui est incontournable pour tout développeur Web. Il est complexe avec beaucoup de subtilités à connaître et il faut désormais pousser l'abstraction et l'organisation à un niveau supérieur, grâce la Programmation Orientée Objet (POO) notamment.

Cependant, même si vous apprendrez à programmer Objet en Javascript, cette formation cherche aussi à vous munir des bons outils à tous les niveaux : conformité, documentation, tests, performances, debugger, etc. et à les utiliser avec recul (l'outil ne doit pas faire oublier la conception et la réflexion en amont).

Vous apprendrez également à mettre en œuvre les nouveautés apportées par EcmaScript (ES6, ES7, etc.) pour que votre code soit plus robuste, lisible, et facile à maintenir.

Les développeurs Javascript sont assistés par de plus en plus de frameworks qui sont incontournables dans les projets professionnels. Aussi, vous disposerez d'une vue claire sur le panorama actuel des frameworks Javascript.

Objectifs

A l'issue de cette formation JavaScript avancée vous saurez :

- Faire de la Programmation Objet en JavaScript
- Construire vos programmes Javascript de façon professionnelle (classe, mixin, modules, etc.)
- Respecter les bonnes pratiques de codage avec recul

- Maîtriser les outils du développeur Javascript (debug, test, performances, conformité, documentation, etc.)
- Manipuler les Web Services en Javascript
- Intégrer les nouveautés du langage dans vos programmes avec EcmaScript et les versions plus récentes
- Comprendre le rôle et la complémentarité des principaux frameworks Javascript (Angular, React, Node, etc.)

À qui s'adresse cette formation ?

Public :

Cette formation Javascript avancée s'adresse aux développeurs ou au chef de projet.

Prérequis :

Au minimum connaître les bases de HTML, CSS et Javascript.

Avoir suivi la formation "JavaScript, les fondamentaux" (IJDH) ou disposer d'un niveau équivalent.

Contenu du cours javascript avancé

Programmation objet en Javascript

Intérêts pour le développeur et les projets (lisibilité, maintenance, évolutions, etc.)

Les concepts de base (classe, méthode, propriété, objet)

Créer ses propres objets et utiliser des objets existants

Exemples concrets en Javascript

Constructeur et Prototype

Contexte (this)

Mise en œuvre de l'héritage en Javascript

Aller plus loin en POO avec Javascript ...



ITgate

Training

Your Gateway to Excellence

Les classes natives : String, Array, Object, etc.

Augmenter ou réécrire ?

Visibilité : propriétés publiques / privées

Visibilité : méthodes publiques / privées / privilégiées

Les différentes formes d'héritage

Appels parents : call / apply / bind

Méthodes et propriétés statiques

Méthodes abstraites et interfaces

Duck Typing et Javascript

Programmation objet

Maîtriser le langage Javascript

Types de données et subtilités (constantes, variables, énumération, etc.)

Les différentes formes d'incrémentement

Manipulation de flux binaire

Arrays : manipulation avancée

Règles de transtypage et comparaisons

Savoir utiliser les expressions régulières

Aspects avancés des fonctions en Javascript (contrôle des paramètres, etc.)

Gestion des erreurs : exceptions

Ajax, JSON, Web services

Rappels sur Ajax

JSON et (dé)sérialisation

Traitements asynchrones (promesses, fetch, async/await, etc.)

Consommer des web services en Javascript

Un exemple de Javascript côté serveur avec Node.js

Les différentes formes de stockage des données (Local, Cookies, SQLite, etc.)

Bonnes pratiques en JavaScript



ITgate

Training

Your Gateway to Excellence

Mise en page du code avec ESLint

Imposer des règles de codage « propre »

Portée : encapsulation / namespace

Documenter votre code source « proprement » avec JSdoc

Performances, tests et débogage

Goulot d'étranglement des phases de chargement d'une page Web

Analyse fine des performances de bout en bout avec Navigation Timing API

Rendre votre code Javascript plus efficace avec Closure Compiler

Gestion mémoire et Garbage collector

Synthèse des outils de tests (JUnit, SugarTest, etc.)

Tester efficacement la compatibilité multi navigateurs

Debugger avec Chrome Devtools et Firefox Developer

Industrialisation du développement

Gestion de paquets et dépendances

Comment gérer les nombreuses bibliothèques externes ?

Bien comprendre Node et NPM

L'outil yarn

Gérer son projet et ses paquets en ligne de commande

Construire son application et automatiser l'installation de dépendances

Industrialiser les processus de compilation et mise en production de ses applications

EcmaScript et TypeScript

Vocabulaire et normes (Ecma, typescript, EcmaScript, Javascript, ES10/ES2019, etc.)

Les changements nets apportés de la version ES6

Les apports de TypeScript

Les Types et types spéciaux

Les interfaces, les classes, les fonctions, les mixins

Les génériques

Types avancés



ITgate

Training

Your Gateway to Excellence

Vue générale des frameworks JavaScript

Limites du JavaScript natif

Comprendre le rôle des principaux frameworks Javascript : Angular, Vue, React, ReactNative, NodeJS, Express JS

Points communs et différences techniques

Quel framework Javascript pour quel projet ?

Les tendances du marché

Exemple d'application WEB avec Angular

Exemple d'application WEB avec VueJS

Travaux Pratiques

Tous les modules de ce programme font l'objet de travaux pratiques dont voici quelques exemples :

- Développement d'une calculatrice graphique en Javascript permettant de profiter pleinement des concepts objets
- Installation de dépendances avec NPM
- Réécriture de codes sources avec la nouvelle notation ES6
- Communication depuis Ajax avec des API Rest distantes
- Vue JS:
Création d'un projet, de composants, de mixins, de plugins, et de templates.
Utilisation des plugins vue-resources (requêtes AJAX sur une API), vue-router (pour gérer le routing de l'application), vue-validator (pour valider les formulaires) et vuex (state manager pour Vue.js)
- Angular:
Création d'un projet, interaction avec un formulaire, intégration CSS design dans le projet, exécution de requêtes WEB JSON