JavaScript : paradigmes, programmation objet et fonctionnelle

Développement ▸ Web avancé

Présentation

JavaScript est un langage avec une histoire riche ayant façonné sa syntaxe. Un langage d'abord pensé pour de la manipulation de DOM (Document Object Model) et une programmation événementielle, où ses conventions se retrouvent agrémentées de celles de nombreuses bibliothèques comme jQuery ou Dojo. Puis, ce langage a dépassé l'environnement du navigateur, avec PhoneGap et NodeJS d'abord, puis Electron et React Native pour ne citer qu'eux. Si JavaScript a inspiré des évolutions dans différents langages de programmation, JavaScript a aussi pioché de bonnes idées ailleurs, influençant sa syntaxe et sa logique de programmation ; on citera à titre d'exemple la norme ECMAScript 6 apportant à JavaScript le concept de classes. Ainsi, dans JavaScript, cohabitent notamment un paradigme fonctionnel avec les logiques de map/reduce, un paradigme orienté objet avec le mot-clé class et un paradigme orienté prototype avec la définition flexible des structures. Ce cumul d'approches différentes, qui peut sembler incohérent ou obscur, a pourtant tout son intérêt selon vos objectifs.
À travers le prisme de la programmation orientée objet, cette formation vous propose d'étudier par la pratique différentes approches de programmation en JavaScript. À l'issue de cette formation, vous serez en mesure non seulement d'analyser les tendances de développement connexes à JavaScript pour vous aider à anticiper ses évolutions futures, mais également de mieux envisager les philosophies et choix de paradigmes d'autres langages.

Objectifs

- Comprendre la notion de paradigme
- Découvrir les différents paradigmes utilisés au sein de JavaScript
- Savoir utiliser les différentes manières de créer des instances et classes
- Comprendre les avantages et inconvénients des différents paradigmes
- Analyser l'état actuel de JavaScript pour comprendre les évolutions à venir

Moyens pédagogiques

Formation alternant la théorie et la pratique, avec un intervenant présentant la formation en direct. Possibilité de réaliser cette formation en face-à-face dans nos locaux, dans vos locaux d'entreprise ou à distance via visioconférence.
Une fois terminée, vous disposerez d'une attestation de suivi de la formation, ainsi que des contenus de cours relatifs au contenu présenté.

Evaluation des acquis

Questionnaires à choix multiples (QCM) en fin de formation.
L'intervenant formulera également une appréciation sur vos acquis pour l'entièreté de la formation.

Durée : 14 heures (2 jours)
Référence : DE-077
Niveau : Débutant
Formation inter-entreprises :

Formation en présentiel et distanciel
Tarif : 1150 € HT

Formation intra-entreprises :

Formation disponible et adaptable sur demande
Tarif : sur devis

Contact

Anthony DUPUIS
Conseiller formation
07 68 53 96 11
anthony@ls-a.fr

Public :

Toute personne souhaitant améliorer ses compétences en développement informatique.

Avez-vous les prérequis ?

Une connaissance du langage JavaScript équivalent à la formation DE-078 (JavaScript niveau 1 : dynamiser son site web) est nécessaire

Dernière modification :

08/11/2024

Document non contractuel. ls-a Formation est une marque déposée de ls-a. Tous droits réservés.

JavaScript : paradigmes, programmation objet et fonctionnelle

Développement ▸ Web avancé

Programme de la formation

1- Paradigmes de programmation

Qu'est-ce qu'un paradigme ?
Que sont les paradigmes utilisés en JavaScript ?
Différences entre l'orienté objet et l'orienté prototype
Rôle de la programmation fonctionnelle dans JavaScript

2- Objets et orienté prototype en JavaScript

Concepts d'orienté objet : encapsulation, polymorphisme, héritage, agrégation
Éléments d'orienté objet : constructeur, getter et setter, propriétés, méthodes
Les descripteurs de propriétés (Property Descriptor)
Duplicité du mot-clé « new » en JavaScript
Les fonctions « bind », « call » et « apply »
Affectation dynamique de propriétés et méthodes

3- Modèle basé sur des objets littéraux

Que sont les objets littéraux ?
Quelle compatibilité avec les grands principes de la POO ?
Avantages et inconvénients de ce modèle

4- Modèle basé sur les objets issus de fonctions

Que sont les objets issus de fonctions ?
Quelle compatibilité avec les grands principes de la POO ?
Avantages et inconvénients de ce modèle

5- Modèle basé sur les instances de classes

Comment créer des classes et des instances avec JavaScript ?
Quelle compatibilité avec les grands principes de la POO ?
Avantages et inconvénients de ce modèle

6- Conclusion

Clés pour choisir son modèle selon les besoins
Bonnes pratiques pour cumuler ces modèles dans votre projet
Qu'attendre des prochaines versions d'ECMAScript ?