CROSSTHINK

JavaScript avance

JavaScript avance

Public et pré-requis

Cette formation ne nécessite pas de prérequis.

Objectifs

JavaScript avance

Moyens pédagogiques

•Présentation du formateur et du programme.
•Présentation et écoute de chacun de stagiaires.
•Apports didactiques pour apporter des connaissances communes.
•Mises en situation de réflexion sur le thème du stage et des cas concrets.
•Méthodologie d'apprentissage interactive et participative.
•Exercices et études de cas concrets.
•Temps d'échanges.
•Accompagnement pédagogique individualisé.

Formateur

Les formateurs de CROSSTHINK sont des experts de leur domaine, disposant d'une expérience terrain qu'ils enrichissent continuellement. Leurs connaissances techniques et pédagogiques sont rigoureusement validées en amont en interne.

Suivi de l'execution et évaluation des résultats

•Accueil des stagiaires dans une salle dédiée à la formation / A distance
•Feuilles de présence.
•Documents supports de formation projetés.
•Mise à disposition du stagiaire des documents et supports de formation.
•Tout au long et/ou à l'issue de la formation : Evaluation des acquis des stagiaires via des exercices, des QCM, des QUIZZ, des mises en situation et/ou des cas pratiques.
•Enquête de satisfaction.
•Attestation de fin de formation.

JavaScript avance

EN BREF

2 jours

990€

CONTACT

contact@crossthink.fr

04.83.43.47.48

PROCHAINES DATES

Nous consulter

CROSSTHINK

JavaScript avance

Programme

Rappels importants du langage
•Les éléments de premier ordre dans JavaScript
•La portée des données
•Les fonctions
•Objets ou fonctions ?
•Le débogage avec les meilleurs outils actuels

Pièges du langage
•Typage faible
•”Hoisting”
•Contexte et variables globales
•Changement de contexte

Programmation objet
•Différentes façons de créer des objets
•Constructeurs
•valeur de ”this” dans un objet
•Prototype et prototype
•Diverses façons d’hériter
•Visibilité
Exemples de travaux pratiques (à titre indicatif)
•Conception d’une application qui exploite les fonctions
•Transformation de l’application, permettant d\\’exploiter les objets et l\\’héritage

Importance de la modularisation
•Créer un module en JavaScript
•Utilisation des modules

Programmation fonctionnelle
•Fonctions anonymes
•Fonctions immédiates
•Fonctions internes
•Redéfinition
•Surcharge et mot-clé (arguments)
•Closure

Structuration et qualité du code
•Séparation en multiple fichiers
•Définition de modules
•Asynchronous Module Definition (AMD)
•AMD avec Require.js
•Impacts des \\”closures\\” sur la lisibilité
•Qualité avec JSHint et JSLint
Exemples de travaux pratiques (à titre indicatif)
•Modularisation avec Require.js de l\\’application développée le jour précédent
•Utilisation des outils de Lint

Evolutions récentes : ECMAScript 6 et 7
•Présentation des langages dérivés de JavaScript
•TypeScript
•Dart
•CoffeeScript
•ECMAScript 6 et ses nouveautés

CROSSTHINK

JavaScript avance

•Classes
•Modules
•Fonctions fléchées
•Promise
•Nouvelles méthodes de ”Object”…

Frontière entre JavaScript et HTML 5

•Validation des formulaires en JavaScript
•Solutions de stockage
•LocalStorage
•SQLite
•JSON
•Web Sockets
•Web Workers

JavaScript et Node.js

•Différences dans les langages
•Asynchronisme
•REST serveur en Node.js et REST client en JavaScript
•Présentation d\\’une solution Web 100% JavaScript
Exemples de travaux pratiques (à titre indicatif)
•Exploitation des API de HTML 5 depuis JavaScript
•Utilisation des dernières fonctionnalités de ES7 et 8, avec \’asynchronisme (wait/async), promises
•Intégration d’un serveur simple Node.js dans la solution

Certification (en option)

•Prévoir l’achat de la certification en supplément
•L’examen (en français) sera passé le dernier jour, à l’issue de la formation et s’effectuera en ligne
•Il s’agit d’un QCM dont la durée moyenne est d’1h30 et dont le score obtenu attestera d’un niveau de compétence

Certification