

Formation JavaScript: Maîtriser les objets et les techniques avancées (JSADVANFR, 5 jours)

Description

Formation JavaScript : Maîtriser les objets et les techniques avancées couvre la programmation orientée objet avec JavaScript et explore des techniques avancées telles que les appels asynchrones, les promesses, les travailleurs et plus encore. La formation commence par des concepts JavaScript avancés tels que les classes, l'héritage et le polymorphisme. Vous apprendrez à utiliser le HTML dynamique avec JavaScript et le DOM, ainsi que JSON et Ajax pour le chargement partiel de données. Le cours se poursuit par une discussion détaillée sur la consommation de micro-services à l'aide d'appels asynchrones. La discussion commence par un regard sur l'anatomie des appels asynchrones, y compris les événements et les gestionnaires d'événements. L'utilisation de callbacks pour le traitement des échecs et des succès est explorée en détail, y compris l'utilisation de l'objet XMLHttpRequest et l'utilisation de promesses. Nous explorons également les travailleurs JavaScript tout en approfondissant la mise en œuvre correcte de tous ces concepts à l'aide d'objets JavaScript et de modèles de données d'entreprise. Ce cours vous apprend à faire à peu près tout ce que vous voulez avec JavaScript.

Tarifs

- Tarification: \$3,350/person
- Rabais de 10% lorsque vous inscrivez 3 personnes.

Plan de cours

Révision des principes fondamentaux de JavaScript

À propos de l'inclusion de JavaScript dans une page HTML

À propos de Scope et JavaScript

Création de fonctions JavaScript

Déclarer des variables JavaScript : var versus let

La puissance du pointeur de fonction

Manipuler le HTML avec JavaScript

Création de classes avec JavaScript

Explorer les nombreuses formes de fonctions

À propos de la relation entre les objets et les classes

À propos des prototypes et des objets JavaScript

Création de classes avec JavaScript

Comment déclarer des variables et des fonctions membres privées ?

Mise en œuvre des propriétés des objets

Comprendre la signification de "ceci" et de "soi"

Écriture d'un constructeur

Appeler un objet à partir de HTML

Création de sous-classes avec JavaScript

Explorer les généralisations et l'héritage

Le polymorphisme magique

La signification de this et self dans une sous-classe

Création de sous-classes et appel de constructeurs

Appel des méthodes et des propriétés d'une sous-classe

Comment tout hérite d'un objet

Tout mettre en place

Gestion des erreurs et utilisation des objets JavaScript courants

La différence entre les tableaux et les dictionnaires

Comprendre les classes de collection : Tableaux et cartes

Extension des objets JavaScript : Tableau et carte

La magie d'Iterable : Boucler une collection

Ajout et suppression d'éléments de collection

Création d'objets d'exception personnalisés

Création d'un gestionnaire d'erreurs personnalisé

Utiliser l'héritage et le polymorphisme pour généraliser la collecte d'erreurs

Organiser le code avec des modules

La nécessité des paquets de code en JavaScript

Création d'un module en Vanilla JavaScript

Les nombreuses formes d'exportation

Les nombreuses formes d'importation

À propos des modules et du champ d'application

Exportation de la fonctionnalité vers le HTML

Architecture d'application et appel de ressources distantes

L'anatomie d'un appel Web JavaScript

Que sont les micro-services et pourquoi sont-ils importants ?

Exploration de JSON et de l'objet XMLHttpRequest

À propos de la sérialisation des objets en JSON

Appeler des actifs Web de manière synchrone

Interagir avec le HTML

Faire des appels asynchrones

Comprendre les événements et les gestionnaires d'événements

L'anatomie d'un appel asynchrone : Appels et rappels

Appeler des microservices à l'aide de XMLHttpRequest

Mise en œuvre de fonctions de rappel pour les succès et les échecs

Création de la bibliothèque d'objets pour stocker les résultats

Génération de contenu HTML dynamique

Exploration de la logique de charge

Tout mettre en place dans le HTML

Utilisation des Promesses

À propos des appels asynchrones et du modèle Promise

Création d'une promesse avec des rappels de réussite et d'échec

À propos du chaînage de promesses

À propos de la gestion des erreurs des promesses

Utiliser les Promesses pour rendre tout asynchrone

Intégration de la sémantique des appels par le biais d'objets

Utilisation des travailleurs

La nécessité des Worker Threads en JavaScript

Comprendre l'architecture des communications des travailleurs

À propos de postMessage et onMessage

Mise en œuvre du travailleur : onMessage et postMessage

Mise en œuvre du client : onMessage et postMessage

Démarrer le travailleur, obtenir des mises à jour et obtenir un résultat

Traitement des erreurs du travailleur

Faire des appels asynchrones avec des travailleurs

Appels d'authentification et nettoyage

Re-Factoring du code dupliqué

Nettoyage de l'HTML

Transmettre des données JSON à des microservices grâce à la sérialisation

À propos de la visibilité des données dans la sérialisation JSON

Explorer les interactions avec des plates-formes différentes

Terminer les choses !