

# JavaScript, maîtriser la Programmation Orientée Objet

Cours Pratique de 3 jours - 21h

Réf : JAO - Prix 2024 : 2 070€ HT

Vous maîtriserez les mécanismes de la programmation Objet en JavaScript. Vous étudierez en détail son modèle Objet en mettant en œuvre notamment le concept de classes et d'héritage. Vous découvrirez également les mécanismes avancés du JavaScript et l'intérêt de l'intégrer avec du TypeScript.

## OBJECTIFS PÉDAGOGIQUES

À l'issue de la formation l'apprenant sera en mesure de :

Comprendre les particularités de la Programmation Orientée Objet avec JavaScript

Créer des classes avec leurs constructeurs et leurs propriétés

Mettre en œuvre le principe de l'héritage

Articuler le TypeScript avec la syntaxe JavaScript

## LE PROGRAMME

dernière mise à jour : 06/2021

### 1) Vision d'ensemble des classes

- Rappels sur la POO.
- Le mécanisme des classes.
- Définition des propriétés et méthodes public, private.
- Création de classe Template.
- Comprendre le mécanisme de prototype, \_\_proto\_\_.
- Objet et typage des objets TypeOf, instanceof.

### 2) Constructeur et propriétés

- Comprendre les mécanismes de constructeur.
- Constructeur basé sur Object.create.
- Constructeur literal.
- Principe de scope safe.
- Création par singleton, pattern.
- Méthodes de création des propriétés, portée (private, public, static) des propriétés.
- Portée des méthodes, mécanismes de polymorphisme.
- Préserver et gérer le Context this.

*Travaux pratiques : Mise en œuvre des différentes méthodes de création de classe.*

### 3) Les concepts d'héritage

- Comprendre les mécanismes d'héritage en JavaScript.
- Héritage simple, multiple.
- Héritage par prototype.

*Travaux pratiques : Création d'une classe héritée simple avec et sans les prototypes.*

### 4) Comprendre le type Object

- Création Object literal, avec un constructeur.
- Création de propriétés et méthodes.

#### PARTICIPANTS

Développeurs et chefs de projets Web.

#### PRÉREQUIS

Programmeurs ayant des connaissances JavaScript de base.

#### COMPÉTENCES DU FORMATEUR

Les experts qui animent la formation sont des spécialistes des matières abordées. Ils ont été validés par nos équipes pédagogiques tant sur le plan des connaissances métiers que sur celui de la pédagogie, et ce pour chaque cours qu'ils enseignent. Ils ont au minimum cinq à dix années d'expérience dans leur domaine et occupent ou ont occupé des postes à responsabilité en entreprise.

#### MODALITÉS D'ÉVALUATION

Le formateur évalue la progression pédagogique du participant tout au long de la formation au moyen de QCM, mises en situation, travaux pratiques...

Le participant complète également un test de positionnement en amont et en aval pour valider les compétences acquises.

#### MOYENS PÉDAGOGIQUES ET TECHNIQUES

- Les moyens pédagogiques et les méthodes d'enseignement utilisés sont principalement : aides audiovisuelles, documentation et support de cours, exercices pratiques d'application et corrigés des exercices pour les stages pratiques, études de cas ou présentation de cas réels pour les séminaires de formation.
- À l'issue de chaque stage ou séminaire, ORSYS fournit aux participants un questionnaire d'évaluation du cours qui est ensuite analysé par nos équipes pédagogiques.
- Une feuille d'émargement par demi-journée de présence est fournie en fin de formation ainsi qu'une attestation de fin de formation si le stagiaire a bien assisté à la totalité de la session.

#### MODALITÉS ET DÉLAIS D'ACCÈS

L'inscription doit être finalisée 24 heures avant le début de la formation.

#### ACCESSIBILITÉ AUX PERSONNES HANDICAPÉES

Vous avez un besoin spécifique d'accessibilité ? Contactez Mme FOSSE, référente handicap, à l'adresse suivante psh-accueil@orsys.fr pour étudier au mieux votre demande et sa faisabilité.

- Duplication d'objet par valeur, référence.
- Gestion du contexte this avec Object.apply et Object.call.
- Démystifier les propriétés de Object.
- Modules : Object, AMD.

*Travaux pratiques* : Construction d'une classe, factory à base de Object.

### 5) Mécanismes JavaScript avancés

- Déclaration des variables, typage, nomenclature, accès au DOM.
- Comprendre le mécanisme de Hoisting.
- Promises et la Programmation asynchrone.
- Echange Ajax d'objets structurés et serialization de multiples formats (texte, image, JSON).
- Création d'un Namespace static, Namespace objet, module pattern, namespace proxy.
- Fonctions anonymes, closures.
- Comprendre les principes du callback.

*Travaux pratiques* : Gestion des solutions de stockage localStorage pour conserver image, objets...

### 6) Pourquoi utiliser TypeScript ?

- Typage de données.
- Constructeurs, instances (méthode et propriété), interfaces.
- Création de Class, Objects (private, public, static...).
- Getters, Setters.
- Héritage, surcharge.
- Gestion du context.
- Polymorphisme.

*Travaux pratiques* : Création d'un projet en programmation objet en Typescript, usage de solution Node.js.

## LES DATES

---

Nous contacter