

# ECMAScript 6, mise en oeuvre anticiper la migration des applications

Cours Pratique de 3 jours - 21h

Réf : SPT - Prix 2024 : nous consulter

Le langage JavaScript est devenu un langage incontournable dans les développements Web et mobiles tant côté client que côté serveur avec notamment Node.js. Cette formation vous permettra d'utiliser en production dès aujourd'hui les nouvelles fonctionnalités des dernières versions EcmaScript (2015, 2016, 2017).

## OBJECTIFS PÉDAGOGIQUES

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

Découvrir les nouveautés syntaxiques du JavaScript

Maîtriser les nouvelles API

Utiliser ES6 en production

Anticiper l'évolution des frameworks du Front-End

Se préparer aux nouvelles méthodes de développement JavaScript

## MÉTHODES PÉDAGOGIQUES

Développement. Migration d'applications existantes. 50% de théorie illustrée par 50% de pratique.

## TRAVAUX PRATIQUES

Réalisation d'une application et portage en modules ES6.

## LE PROGRAMME

dernière mise à jour : 08/2018

### 1) De JavaScript ES5 à ES2017

- EcmaScript : évolution et "roadmap".
- ES5, rappel des bonnes pratiques. Etablir les règles de programmation.
- ES5 "use strict" et méthodes moins connues.
- API issues de la communauté JavaScript.
- Support courant pour ES6 : compilateurs, polyfills, navigateurs serveurs.
- Environnement et outils pour le développeur.

*Travaux pratiques* : Vous programmerez, à partir d'exemples fournis, un micro framework applicatif JavaScript.

### 2) Evolutions syntaxiques fondamentales

- Constantes et variables de bloc. Assignment destructurée.
- Chaînes de caractères : multiligne, template, formatage.
- Fonction, paramètres par défaut, opérateurs "rest/spread".
- "Arrow Function" : portée lexicale. Usages.
- Fonction génératrice : "yield" et "generator object".
- Fonction itératrice : "iterator".
- Objet littéral : évolution.

*Travaux pratiques* : Migration du code vers ES6.

### 3) POO, nouveautés pour la conception Objet

- Modèles de classe et héritage. Méthodes statiques.
- Création de "proxy". Design pattern.
- Nouveaux types : Set, Weakset, Map, Weakmap.

## PARTICIPANTS

Développeurs JavaScript confirmés, architectes et chefs de projets.

## PRÉREQUIS

Bonne connaissance de JavaScript (ES5) est requise. Connaissance de base des composants du Web.

## 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.

- Objets natifs héritables.

*Travaux pratiques* : Organisation et structuration du code pour une approche Objet par classe. Révision des "Design Pattern".

#### 4) Nouvelles API JavaScript avec ES6

- Promise, async/await : gestion des traitements asynchrones.

- Object.API : revisiter les méthodes.

*Travaux pratiques* : Développement d'un service de données Ajax utilisant l'API des Promise.

#### 5) Modules JavaScript

- Asynchronous Module Definition ou CommonJS.

- Système natif de gestion des modules.

- Gestion et résolution des dépendances.

- Chargement dynamique.

*Travaux pratiques* : Structurer son développement par modules.

#### 6) Mise en production

- Stratégie : "transpiler", "package manager" et système de module.

- Choix du "transpiler" : présentation des solutions (Traceur, Babel).

- Typage : mise en œuvre de TypeScript, Flow.

- Pourquoi "packager" son code ? Avantages et solutions.

- Création de package avec npm, bower et jspm.

- Choisir entre RequireJS, Browserify, WebPack et SystemJS.

- Interactions avec les autres outils (linting, test...).

*Travaux pratiques* : Création et intégration de package dans le développement.

## LES DATES

---

Nous contacter