

## EcmaScript 6, mettre en oeuvre le nouveau JavaScript anticiper la migration des applications



SII-121 3 Jours (21 Heures)

### Description

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 de la dernière version EcmaScript 2015.

### À qui s'adresse cette formation ?

#### Pour qui

Développeurs JavaScript confirmés, architectes et chefs de projets.

#### Prérequis

Aucune

### Les objectifs de la formation

- Découvrir les nouveautés syntaxiques du JavaScript Maîtriser les nouvelles API Utiliser ES6/ES2015 en production Anticiper l'évolution des Frameworks du Front-End Se préparer aux nouvelles méthodes de développement JavaScript

### Programme de la formation

#### De JavaScript ES5 à ES2015

- ES5, JS.
- next, ES6 / ES2015 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.

## Evolutions syntaxiques fondamentales

- Constantes et variables de bloc.
- Assignation 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 ES2015.

## POO, nouveautés pour la conception objet

- Modèles de classe et héritage.
- Méthodes statiques.
- Création de "proxy".
- Design pattern.
- Nouveaux types : Set, Weakset, Map, Weakmap.
- Objets natifs héritables.
- Travaux pratiques Organisation et structuration du code pour une approche Objet par classe.
- Révision des "Design Pattern".

## Nouvelles API JavaScript avec ES6

- Promise : gestion des traitements asynchrones.
- Object.
- API : revisiter les méthodes.
- Travaux pratiques Développement d'un service de données Ajax utilisant l'API des Promise.

## 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.

## Mise en production

- Stratégie : "transpileur", "package manager" et système de module.
- Choix du "transpiler" : présentation des solutions.
- Mise en oeuvre de TypeScript, Traceur, Babel.
- 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.