ECMASCRIPT 6, METTRE EN OEUVRE LE NOUVEAU JAVASCRIPT

REF **EMOJS**

- ***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***

3

Prix : 2?090 € € / HT

OUTILS PÉDAGOGIQUES

MODALITÉS D'ÉVALUATION

MODALITÉS DE FINANCEMENT

MODALITÉS ET DÉLAIS D'ACCÈS

OBJECTIFS PÉDAGOGIQUES

ACCESSIBILITÉ

LES POINTS FORTS DE LA FORMATION

PRÉ-REQUIS

- Bonne connaissance de JavasScript (ES5) est requise. Connaissance de base des composants du web.

MODALITÉS ET DÉLAIS D'ACCÈS

ATTESTATION OBTENUE

EFFECTIF DE LA FORMATION

CERTIFICATION

MODALITÉ PÉDAGOGIQUE

Cours dispensé en mode présentiel avec une alternance d'apports théoriques et méthodologiques, et de mises en situations pratiques

PROCHAINES SESSIONS

Le 14/03/2022
Le 20/06/2022
Le 14/03/2022
Le 20/06/2022
Le 14/03/2022
Le 20/06/2022

PROGRAMMES DE ECMASCRIPT 6, METTRE EN OEUVRE LE NOUVEAU JAVASCRIPT

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, polifylls, navigateurs serveurs.
- Environnement et outils pour le développeur.

ECMASCRIPT 6, METTRE EN OEUVRE LE NOUVEAU JAVASCRIPT

REF **EMOJS**

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.

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.

Nouvelles API JavaScript avec ES6

- Promise, async / await : gestion des traitements asynchrones.
- Object.API : revisiter les méthodes.

Modules JavaScript

- Asynchronous Module Definition ou CommonJS.
- Système natif de gestion des modules.
- Gestion et résolution des dépendances.
- Chargement dynamique.

Mise en production

- Stratégie : « transpiler », « package manager » et système de module.
- Choix du « transpiler » : présentation des solutions (Traceur, Babel).
- Typage : mise en oeuvre 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…).