

Formation Javascript/EcmaScript : Initiation + Approfondissement

Durée :	5 jours
Public :	Développeurs Web - Webmasters - Intégrateurs
Pré-requis :	Connaissance en HTML et CSS
Objectifs :	Apprendre et maîtriser les bases de JavaScript, du DOM et de l'Ajax
Sanction :	Attestation de fin de stage mentionnant le résultat des acquis
Taux de retour à l'emploi:	Aucune donnée disponible
Référence:	SIT100292-F
Note de satisfaction des participants:	4,52 / 5

Introduction

- Présentation
- Panorama des environnements JavaScript
- Versions et standard ECMAScript

Écrire des instructions en Javascript

- Les variables et les types de données
- Les opérateurs
- Portée des variables var, let, const
- Les instructions de bloc
- Les structures de contrôle (conditionnelles et itératives)

Déclarer et appeler des fonctions

- Définir une fonction, syntaxe déclarative et expressions
- Transmettre des paramètres
- Retourner une valeur
- Définir des paramètres par défaut
- La console (log, debug, info, error, ...)

Interagir avec le document via le DOM

- Présentation, historique et version
- Les objets window et document
- Accéder aux éléments (getElementById, querySelector, querySelectorAll)
- Découvrir les événements
- Liste des événements de souris et de clavier (click, keypress, mouseover, etc)

Gestion des fenêtres et boîtes de dialogues

Atelier : Faire afficher et masquer des balises HTML

Manipuler des tableaux

Présentation et structures d'un tableau
Déclaration de tableau (littéral, constructeur)
Ajout et suppression de données (push(), splice())
Parcourir un tableau (forEach, for of)
Les fonctions spécifiques (slice, pop, shift, reverse, etc)
Les fonctions find() et findIndex()

Appréhender le concept d'objet

Présentation
Les objets littéraux
Ajouter & accéder aux propriétés
Les property shorthand

Exploiter des sucres syntaxiques

Les littéraux de templates
Affectation par décomposition
Rest Parameter & Spread Operator
Les fermetures
Les fonctions fléchées

Atelier : Filtrer des données affichées dans une liste

Déclencher des traitements au travers d'événements

Principe des écouteurs d'évènement
Les évènements attacher au navigateur (load, unload, resize, etc)
Les évènements de formulaire (submit et reset)
Récupérer et manipuler l'évènement
Les fonctions preventDefault() et stopPropagation()

Atelier : Valider les données d'un formulaire

Gérer dynamiquement les attributs

Manipuler les attributs (getAttribute & setAttribute)
Modifier le contenu d'une balises (innerHTML, innerText, innerContent)
Liste des propriétés css en javascript
Modifier le css à l'aide de l'attribut style
Ajout & suppression de class (className, classList)

Parcourir et transformer le DOM

Parcourir l'arbre DOM (parentNode, childNode)
Les fonctions parentChild() et firstChild()
Créer et insérer une balise (createElement, appendChild, createTextNode)
Les fonctions NextSibling et PreviousSibling

Supprimer une balise (removeChild)

Atelier : Réordonner une liste avec du drag & drop

Aller plus loin avec les objets

Le prototypage
Déclaration de méthodes
L'association d'objet rapide (Object.assign())
Les chaînes de caractères et les dates
Les méthodes associées aux chaînes de caractères
La classe Math
Internationalisation: format de dates et monétaires

Utiliser les syntaxes de classe

Déclaration de Classe
Constructeur et accesseur
Les variables et méthodes de classe (static)
Héritage
Iterator et Generator
Les collections (Set et Map)

Atelier : Créer le jeu du chifumi en orienté objet

Exploiter les traitements asynchrone avec les promesses

Définition
Créer une promesse
Les fonctions resolve() et reject()
Attacher nos callbacks à une promise (then(), catch())

Atelier : Récupérer des données à l'aide de Promesse

Dialoguer avec un serveur (Ajax, XMLHttpRequest)

Principe de communication asynchrone
Instancier et préparer un objet XMLHttpRequest
Gestion des erreurs
Barre de progression
Protocole HTTP, méthodes et status
Sérialisation et désérialisation de données
Utilisation des données: Le format XML, le format JSON

Atelier : Soumettre et lire des données provenant d'une API REST