[sfeir] Institute

SFEIR Institute | ECMASCRIPT

ECMAScript

Des bases solides pour développer des applications fullstack modernes

2 jour(s) / 14h

Public cible

- Développeurs, intégrateurs, architectes logiciel, chefs de projet technique

Programme

- Avoir suivi le cours JavaScript: Les fondamentaux du langage ou avoir des connaissances équivalentes
- Connaissances de base en HTML

Programme

1. Introduction au standard ECMAScript
    - Une brève histoire de ECMAScript
    - JavaScript vs ECMAScript
    - ECMA International, le comité TC39 et le processus d'évolution d'ECMAScript
    - Les grandes fonctionnalités de ECMAScript, des débuts à aujourd'hui
    - Compatibilité des navigateurs web et de Node.js avec ECMAScript
    - Transpilation et backward compatibility
    - Ateliers:
        - Exploration des outils de compatibilité: caniuse, kangax,...
        - Configuration d'un environnement de développement ECMAScript et mise en oeuvre d'un transpiler

2. Les bases de ECMAScript
    - Déclaration de variables et temporal dead zone
    - Blocs de code
    - Scopes
    - Strict mode et ECMAScript
    - Ateliers:
        - Mise en oeuvre de let et const
        - Etude de cas: scopes
3. Les opérateurs et les paramètres
    - Exponentiation
    - Destructuring
    - Rest
    - Spread
    - Fonctions: valeurs par défaut des paramètres
    - Fonctions: paramètres nommés
    - Fonctions: valeurs de retour multiples
    - Ateliers:
        - Mise en oeuvre des nouveaux opérateurs rest, spread et destructuring
        - Mise en oeuvre des opérateurs avec les fonctions
4. Template Literals
    - String interpolation
    - Multi-line strings
    - Tagged template literals et tag functions
    - Ateliers:
        - Mise en oeuvre de l'interpolation de chaîne de caractère en place de la concaténation
        - Implémentation d'un DSL avec les tagged template literals
5. Arrow Functions
    - Définition et syntaxe des arrow functions
    - Pièges syntaxique
    - Lexical variables
    - Ateliers:
        - Implémentation de collection pipelines avec les Arrow Functions
        - Etudes de cas: events handlers, classes
6. OOP et Classes
    - Les évolutions de la syntaxe objet littérale
    - Les méthodes de Object
    - Déclaration de classe et instanciation d'objets
    - Classes vs Constructor Function et prototypes

    - Method Definitions
    - Héritage
    - Méthodes statiques
    - Ateliers:
        - Définition et dérivation de classe
        - Création d'erreurs personnalisées en sous-classant Error
7. Programmation asynchrone
    - Rappels sur la programmation asynchrone en JavaScript: call stack, callbacks, event loop
    - Promises
    - Async / await et fonctions asynchrones
    - Gestion d'erreur dans des programmes asynchrones
    - Ateliers:
        - Création et utilisation de promises
        - Enchaînement et composition de promises
        - Ecriture de code asynchrone avec les fonctions async
8. Symbols
    - Que sont les symbols
    - Cas d'utilisation des symbols
    - L'API symbol
    - Ateliers:
        - Création et utilisation des symbols
9. Itérateurs et Générateurs
    - L'interface Iterable et les itérables
    - L'interface iterator
    - Boucle for-of et autres constructions de langage pour itérer
    - Les générateurs
    - Générateurs comme itérateurs, observateurs, coroutines
    - Itérateurs et générateurs asynchrones
    - Itération asynchrone et for-await-of
    - Ateliers:
        - Utilisation et implémentation d'itérables
        - Mise en oeuvre des générateurs
10. Modules
    - Les bases des modules en JavaScript
    - Import et Export en détail
    - Utilisation des modules ES6 dans les navigateurs
    - Ateliers:
        - Modularisation d'une base de code existante

11. Collections, données structurées et types
    - Set
    - Map
    - Typed Arrays et modèle mémoire
    - Évolutions des types: String, Array, Object, Number et Math, Regexp,...
    - Ateliers:
        - Mise en oeuvre des types et structures de données
12. Réflection
    - L'objet Reflect
    - Les objets Proxy
    - Ateliers:
        - Méta programmation avec les proxies