

Les API de HTML et CSS – Pour les développeurs

3 jours (21 heures)

Perfectionnement

Présentiel ou classe à distance

à partir de

1905 € HT par personne (TVA en vigueur) Repas inclus en présentiel

Contenu mis à jour le 25/09/2026. Document téléchargé le 04/10/2026.

Objectifs

A l'issue de cette formation, vous serez capable de :

- Identifier les API HTML 5 et leur utilité
- Valider des formulaires avec Web Forms 2
- Dessiner avec Canvas et SVG
- Gérer la déconnexion
- Communiquer avec les WebSockets
- Gérer les tâches longues avec les Web Workers
- Réaliser des animations avec CSS 3
- Décrire le Responsive Design et les liens entre HTML 5 et la mobilité
- Relater l'intérêt d'utiliser les IA comme ChatGPT pour aider à la conception de pages HTML 5 / CSS 3.

Les compétences visées par cette formation font l'objet d'une évaluation.

À qui s'adresse cette formation ?

Développeurs, architectes, chefs de projets techniques.

Prérequis

Avoir suivi le cours JVS-IN "JavaScript – Fondamentaux" ou pratiquer le langage JavaScript.

Programme

1 Pendant votre formation

Les API de HTML et CSS – Pour les développeurs

Jour 1 – Matin

HTML moderne et accessibilité

- Structurer une page avec les balises sémantiques modernes
- Comprendre l'arbre d'accessibilité et les rôles ARIA implicites
- Enrichir avec ARIA quand HTML ne suffit pas : états et propriétés
- Gérer la navigation clavier et le focus

Formulaires natifs

- Exploiter les types de champs modernes : date, color, range, datalist
- Valider côté client : Constraint Validation API, attributs natifs
- Styliser les états : :user-valid, :user-invalid
- Piloter la validation en JavaScript avec setCustomValidity

Exemples de travaux pratiques (à titre indicatif)

- Auditer une page existante avec axe-core et corriger les défauts d'accessibilité
- Construire un formulaire d'inscription validé sans framework

CSS Houdini et API CSS programmables

- Etendre CSS via les Worklets : Paint API, Layout API
- Déclarer des propriétés typées : Properties and Values API (@property)
- Construire dynamiquement une feuille de style : Constructable Stylesheets
- Manipuler les styles avec CSS Typed OM

Exemple de travaux pratiques (à titre indicatif)

- Créer un effet visuel personnalisé via Paint API et l'utiliser sur un composant

Jour 1 – Après-midi

Fetch et requêtes HTTP

- Remplacer XHR par fetch : Request, Response, Headers
- Annuler une requête avec AbortController
- Traiter les réponses en flux : ReadableStream
- Envoyer des données binaires et des fichiers : FormData, Blob

Exemple de travaux pratiques (à titre indicatif)

- Implémenter un upload avec barre de progression et annulation

CORS et sécurité réseau

- Comprendre le modèle Same-Origin et CORS
- Gérer les preflights et les credentials
- Maîtriser CSP et Trusted Types pour mitiger XSS

Server-Sent Events

- Recevoir des événements en flux unidirectionnel : EventSource
- Gérer reconnexion automatique et identifiants d'événements
- Comparer SSE et WebSocket selon le cas d'usage

Exemple de travaux pratiques (à titre indicatif)

- Afficher en temps réel un flux de notifications via SSE

WebSocket et WebTransport

- Etablir une connexion bidirectionnelle WebSocket
- Gérer messages, erreurs, reconnexion
- Découvrir WebTransport : communication moderne sur HTTP/3

Exemple de travaux pratiques (à titre indicatif)

- Construire un mini-chat WebSocket avec gestion robuste de reconnexion

Jour 2 – Matin

Audio et vidéo

- Intégrer media via les balises et leurs API
- Contrôler la lecture en JavaScript : currentTime, playbackRate
- Gérer les pistes texte et sous-titres : track, TextTrack

Exemple de travaux pratiques (à titre indicatif)

- Construire un lecteur vidéo personnalisé avec sous-titres et raccourcis clavier

Canvas et OffscreenCanvas

- Dessiner en 2D : chemins, transformations, ImageData
- Déporter le rendu hors du thread principal avec OffscreenCanvas
- Composer Canvas et CSS

SVG programmable

- Manipuler le DOM SVG depuis JavaScript
- Animer en SMIL ou via Web Animations API
- Combiner SVG et CSS : variables, masques, filtres

Exemple de travaux pratiques (à titre indicatif)

- Construire un graphique interactif en SVG avec animations Web Animations API

WebGL et WebGPU

- Situer WebGL dans l'écosystème actuel
- Découvrir WebGPU : pipeline moderne, compute shaders
- Choisir entre WebGL, WebGPU et bibliothèques (Three.js)

Jour 2 – Après-midi

Web Workers

- Comprendre le modèle mono-thread du JavaScript
- Déléguer des traitements lourds à un Worker dédié
- Communiquer via postMessage et Transferable Objects
- Partager des données via Shared Workers et SharedArrayBuffer

Exemple de travaux pratiques (à titre indicatif)

- Création d'un programme utilisant les Web Workers afin de paralléliser des traitements de calculs scientifiques

Service Workers et PWA

- Enregistrer un Service Worker et gérer son cycle de vie
- Mettre en cache et servir hors-ligne : Cache API, stratégies cache-first / network-first
- Déclarer une PWA installable : manifest.json, icônes, scope
- Gérer notifications push et synchronisation en arrière-plan

Exemple de travaux pratiques (à titre indicatif)

- Transformer une application existante en PWA installable et fonctionnelle hors-ligne

WebAssembly

- Comprendre le modèle Wasm : modules, mémoire linéaire, imports / exports
- Charger et instancier un module Wasm depuis JavaScript
- Identifier les cas d'usage : calcul intensif, portage de code natif

Exemple de travaux pratiques (à titre indicatif)

- Exécuter une bibliothèque Wasm depuis une page Web et mesurer la performance vs JS

Jour 3 – Matin

Stockage local moderne

- Choisir l'API adaptée : localStorage, sessionStorage, IndexedDB, Cache API
- Modéliser des données dans IndexedDB : object stores, index, transactions
- Accéder au système de fichiers utilisateur : File System Access API
- Mesurer les quotas : Storage Estimate API

Exemple de travaux pratiques (à titre indicatif)

- Construire une application de notes hors-ligne avec IndexedDB et File System Access

Les Observers

- Détecter l'apparition d'éléments : IntersectionObserver

- Réagir aux changements de taille : ResizeObserver
- Suivre les mutations DOM : MutationObserver
- Détecter visibilité et focus : Page Visibility API

Web Components

- Créer un Custom Element autonome
- Isoler le style avec Shadow DOM
- Composer avec slot et template
- Distribuer un composant réutilisable indépendamment du framework

Exemple de travaux pratiques (à titre indicatif)

- Développer un composant carte produit autonome utilisable sans framework

Jour 3 – Après-midi

Permissions et Device APIs

- Comprendre le modèle de permissions du navigateur : Permissions API
- Exploiter la géolocalisation et l'orientation
- Accéder à la caméra et au micro : getUserMedia
- Etablir des flux pair-à-pair avec WebRTC
- Déclencher des notifications système : Notification API
- Partager du contenu : Web Share API
- Accéder au presse-papier : Clipboard API

Exemple de travaux pratiques (à titre indicatif)

- Intégrer caméra, géolocalisation et notifications dans une application existante

Sécurité côté client

- Chiffrer et signer avec Web Crypto API
- Durcir avec Content Security Policy et Trusted Types
- Comprendre les implications HTTPS, COOP, COEP
- Isoler des contextes avec iframe sandbox

Exemple de travaux pratiques (à titre indicatif)

- Ajouter chiffrement local et CSP stricte à une application existante

Les apports de l'Intelligence Artificielle (IA) dans la création de pages Web

- Panorama des assistants : ChatGPT, Claude, Copilot, Gemini, modèles locaux
- Formuler des prompts efficaces pour HTML et CSS : contexte, contraintes, accessibilité, navigateurs cibles
- Repérer les pièges spécifiques : sélecteurs hallucinés, ARIA mal posé, anti-patterns d'accessibilité, code legacy
- Vérifier le code généré : DevTools, Lighthouse, axe-core, validation HTML

Exemple de travaux pratiques (à titre indicatif)

- Générer un composant accessible avec un assistant IA, le faire auditer par axe-core et corriger les défauts détectés

Financement

Plusieurs dispositifs de financement sont accessibles via les OPCO (Opérateurs de Compétences), organismes agréés par le ministère du Travail dont le rôle est d'accompagner, collecter et gérer les contributions des entreprises au titre du financement de la formation professionnelle.

Pour plus d'information, une équipe de gestionnaires SKOLAE Formation spécialisée vous accompagne dans le choix de vos formations et la gestion administrative.

Nous contacter pour en savoir plus →

Modalités pédagogiques

Formation délivrée en présentiel ou distanciel* (blended-learning, e-learning, classe virtuelle, présentiel à distance).

Le formateur alterne entre méthode** démonstrative, interrogative et active (via des travaux pratiques et/ou des mises en situation).

Variables suivant les formations, les moyens pédagogiques mis en oeuvre sont :

- Ordinateurs Mac ou PC (sauf pour certains cours de l'offre Management), connexion internet fibre, tableau blanc ou paperboard, vidéoprojecteur ou écran tactile interactif (pour le distanciel)
- Environnements de formation installés sur les postes de travail ou en ligne
- Supports de cours et exercices

En cas de formation intra sur site externe à M2I, le client s'assure et s'engage également à avoir toutes les ressources matérielles pédagogiques nécessaires (équipements informatiques...) au bon déroulement de l'action de formation visée conformément aux prérequis indiqués dans le programme de formation communiqué.

* nous consulter pour la faisabilité en distanciel

** ratio variable selon le cours suivi

Accessibilité aux personnes en situation de handicap

Détails de la session

Du 30 novembre 2026 au 2 décembre 2026 (3 jours / 21 heures)

Lille

Du 30 novembre 2026 au 2 décembre 2026

Lille

- 30 novembre 2026 9h00 – 16h00
- 1 décembre 2026 9h00 – 16h00
- 2 décembre 2026 9h00 – 16h00