

Programme de formation PWA Progressive Web Apps

● Objectifs

Alors que le modèle des applications mobiles natives tend à s'essouffler, une nouvelle approche de développement a fait son apparition. Les Progressive Web Applications, qui mêlent le meilleur du web et le meilleur des applications natives ont le vent en poupe. Poussées par Google notamment, les PWA se veulent rapides, fiables, engageantes et disposent de fonctionnalités jusqu'alors réservées aux applications mobiles : accès à la caméra, support hors-ligne ou encore notifications push. Notre formation PWA Progressive Web Apps s'inscrit dans ce changement de paradigme, amené à façonner le Web de demain. Elle vous apportera toutes les connaissances et compétences nécessaires pour le développement d'applications web progressives, pour faire passer l'expérience utilisateur à un tout autre niveau et coller aux nouveaux usages d'un marché en constante mutation. Après avoir passé en revue les nouveaux concepts liés aux PWA, vous apprendrez toutes les bonnes pratiques pour créer et optimiser ces étonnantes applications web !

● Pré requis

Connaissances basiques en JavaScript, HTML et CSS

● Durée

3 jours

● Public

Architectes, Chefs-de-projet, Développeurs

● Plan de formation

Introduction à la formation PWA

Progressive Web Apps

Présentation générale des Progressive Web Apps
Historique et positionnement
La dimension « progressive » des PWAs
PWA vs. applications mobiles natives et Single Page Applications
Architecture et composants principaux d'une PWA
Avantages et limitations
Exemple de cas pratique : Création d'une première application web progressive

Web App Manifest

Créer un fichier manifest.json
Composants (background_color, description, dir, etc)
Configurer les propriétés du Web App Manifest
Web App Install Banner
Compatibilité avec les différents navigateurs mobiles et desktop
Exemples de cas pratiques : Création, configuration, déploiement et test d'un manifest

Service Worker

Les APIs Cache et Fetch
Rappels sur les promises, ES2015
Qu'est-ce qu'un service worker ?
Comprendre l'intérêt de l'utilisation des services workers
Cycle de vie : enregistrement, installation, activation
Périmètre d'action, événements pris en charge
Support des navigateurs
Exemple de cas pratique : Création de votre propre service worker

Gestion du cache

Options de stockage
Identifier les objets à mettre en cache
Cache statique et cache dynamique
Mettre des fichiers en cache avec un service worker
Stratégies de cache et gestion du cache avancée
Implémenter la solution IndexedDB
Travailler avec IndexedDB et les promises

Application Shell

Exemples de cas pratiques : Mise en cache de fichiers avec un service worker, stockage de données avec IndexedDB

Responsive Design

Considérations pour créer des pages web responsives

Les Media Queries, nouveautés de CSS3

Utiliser les images de manière responsive

Ajouter des animations

Exemples de cas pratiques : Styliser son application pour la rendre responsive, utilisation des media queries

Notifications et Push de données

Les APIs Notifications et Push

Options des notifications et gestion des autorisations

Bonnes pratiques pour l'implémentation de notifications push (timing, cible, pertinence, design)

Utiliser Background Sync pour gérer les tâches de fond

Synchroniser des données dans le service worker

Exemples de cas pratiques : Création et affichage de notifications, utilisation de l'API

Background Sync pour différer l'envoi de données

Outils

Bibliothèques et frameworks

Outils de développement des navigateurs

Workbox pour la gestion des service workers

Lighthouse, en tant qu'extension et depuis la ligne de commande

Débogage

Exemples de cas pratiques : Installation de Lighthouse, audit et optimisation d'une application web

Utilisation d'APIs pour interagir avec l'appareil et aller plus loin

GPS, vibration et autres fonctionnalités d'application native

Simplifier le paiement, lire les QR Codes, etc