

Formation

Développement web avec ReactJS

Maîtrisez le développement d'applications web modernes en ReactJS : composants, hooks, appels API, routage, gestion d'état et bonnes pratiques. Formation disponible en présentiel et en classe virtuelle, orientée pratique et projet.

Formation pratique orientée projet

Encadrement par un experts React

4 jours
28 h

Format
Atelier
pratique

Modalité
Présentiel
Distanciel

Niveau
Intermédiaire

INSCRIPTION / RÉSERVATION



Je m'inscris
maintenant



OBJECTIFS PÉDAGOGIQUES

- Comprendre l'architecture d'une application web moderne basée sur ReactJS.
- Savoir créer, composer et réutiliser des composants React avec JSX, props et state.
- Utiliser les hooks (useState, useEffect, etc.) pour gérer l'état et le cycle de vie.
- Intégrer des API REST, gérer les formulaires, la navigation et les erreurs.
- Appliquer les bonnes pratiques de structuration, performance et déploiement.

PUBLIC CIBLE

- Développeurs web souhaitant monter en compétences sur ReactJS.
- Ingénieurs logiciels, intégrateurs front-end, développeurs full-stack.
- Étudiants ou personnes en reconversion ayant des bases en développement web.

PRÉREQUIS

- Bonne maîtrise du HTML5 et CSS3.
- Connaissances pratiques en JavaScript (ES6+ : fonctions fléchées, modules, etc.).
- Connaissances de base des concepts web (HTTP, API REST) fortement recommandées.

PROGRAMME DE LA FORMATION – DÉTAILLÉ

Introduction à React et rappels JavaScript moderne

Contexte et positionnement

- Évolution du développement front-end et Single Page Applications (SPA).
- React dans l'écosystème JavaScript : forces, limites et alternatives.

Rappels JavaScript ES6+

- Let/const, fonctions fléchées, template strings.
- Destructuration, opérateurs spread et rest, modules.
- Promise, async/await et gestion des appels asynchrones.

Mise en place de l'environnement

- Installation de Node.js et npm.
- Création d'un projet React avec Vite ou Create React App.
- Structure de projet, scripts npm, outils de développement.

Fondamentaux de React : composants et JSX

Composants fonctionnels

- Comprendre le modèle composant et la hiérarchie.
- Création de composants fonctionnels simples.
- Organisation des composants par fonctionnalités.

JSX et rendu

- Syntaxe JSX, expressions dans le JSX.
- Conditions et listes (map, key) dans le JSX.
- Styles CSS, inline styles et bibliothèques de composants.

Props et composition

- Passage de données avec les props.
- Composants contrôlés par les parents.
- patrons de composition (children, composants réutilisables).

PROGRAMME DE LA FORMATION – DÉTAILLÉ

Gestion de l'état et hooks de base

useState et gestion de l'état local

- Déclarer et initialiser un état local.
- Mettre à jour l'état de manière synchrone et asynchrone.
- Gestion de listes, formulaires simples et interactions utilisateur.

useEffect et cycle de vie

- Effets de bord : chargement de données, timers, écouteurs.
- Dépendances d'un effet et pièges à éviter.
- Nettoyage des effets (cleanup) et prévention des fuites mémoire.

Autres hooks utiles (aperçu)

- useRef pour référencer des éléments DOM ou stocker des valeurs mutables.
- useMemo et useCallback pour l'optimisation (introduction).

Formulaires, validation et navigation

Formulaires contrôlés

- Inputs contrôlés vs non contrôlés.
- Gestion des événements, champs multiples et formulaires complexes.
- Messages d'erreur et gestion de la validation côté front.

React Router (navigation)

- Installation et configuration de React Router.
- Routes, liens, navigation programmatique.
- Routes dynamiques, paramètres et pages 404.

Intégration des API et gestion d'état avancée

Consommation d'API REST

- Rappels HTTP et format JSON.
- Appels API avec fetch ou axios.
- Gestion du chargement, des erreurs et des états intermédiaires.

PROGRAMME DE LA FORMATION – DÉTAILLÉ

Gestion d'état globale (introduction)

- Context API : création de contextes, providers et consumers.
- Partage d'état entre plusieurs composants.
- Introduction à une librairie de gestion d'état (Redux ou équivalent) – conceptuellement.

Qualité, performance et bonnes pratiques

Tests et debug

- Outils de debug (React DevTools, console).
- Introduction aux tests unitaires et de composants (Jest, Testing Library – aperçu).

Performance et optimisation

- Re-rendu des composants, clés, mémoïsation.
- Code splitting, lazy loading, chargement conditionnel.

Structuration et bonnes pratiques

- Organisation des dossiers (components, pages, services, hooks).
- Nommage, réutilisabilité, lisibilité du code.
- Gestion des variables d'environnement et configuration.

Déploiement et mini-projet de synthèse

Préparation au déploiement

- Build de l'application (mode production).
- Déploiement sur une plateforme cloud (Netlify, Vercel, Azure Static Web Apps – exemples).

Mini-projet de fin de formation

- Conception et réalisation d'une application React complète (liste de tâches, catalogue produits, dashboard, etc.).
- Mise en œuvre : routage, formulaires, appels API, gestion d'état.
- Revue de code, conseils personnalisés et axes d'amélioration.