

Objectifs

- Maîtriser la syntaxe et les fonctionnalités introduites par ES6
- Comprendre les principes fondamentaux du framework ReactJS
- Expliquer la syntaxe JSX, ainsi que la création et l'intérêt des composants
- Définir le rôle des props (propriétés) et leur utilisation dans React
- Comprendre le fonctionnement de la navigation avec React et le routage via react-router-dom
- Présenter le concept de Hooks dans React
- Gérer les événements et réaliser des affichages conditionnels en JSX
- Construire des listes, utiliser des fragments, et gérer les formulaires avec React
- Approfondir le concept de Redux
- Envisager les perspectives d'évolution et appliquer les bonnes pratiques du développement React

Public concerné et prérequis

Développeurs, chefs de projet.

Prérequis

Disposer de connaissances pratique du développement Web.

Maîtrise et pratique de JavaScript (ES5 minimum).

Qualification des intervenants

Spécialiste du domaine.

Moyens pédagogiques et techniques

Le module alternera éléments d'enseignement didactiques et théoriques avec une prise en main progressive des modules à travers des cas pratiques et des exemples exposant les principales problématiques que le stagiaire rencontrera dans la gestion du sujet de manière à acquérir une réelle autonomie opérationnelle.

NEOACADEMY

Outils pédagogiques :

L'ensemble du cours est repris sous forme de projection par vidéoprojecteur.

PC professionnel de l'apprenant et logiciel.

Ateliers de mise en pratique d'expérience.

En début de formation à chaque participant est appelé par le formateur pour un bilan personnalisé des besoins.

Durée, effectifs

21 heures.

6 stagiaires.

Programme

1 - Présentation initiale et rappels sur ES6

- Outils et Environnements de développement intégrés (IDE)
- L'extension pour navigateur React Developer Tools
- Packaging, gestionnaire de paquets npm
- Transpileur EcmaScript
- Déclaration de variables avec let, variables locales et constantes
- Typage et types natifs
- Paramètres optionnels, valeurs par défaut
- Classes et interfaces en programmation
- Gestion des modules et import/export
- Fonctions fléchées (arrow functions)

2 - Le framework React.js

- Concepts fondamentaux : pourquoi choisir React face à ses concurrents et sa conception originelle
- Philosophie basée sur la composition en composants
- Méthodologies de développement : à partir de zéro (personnalisé), intégration dans une application existante, utilisation d'un générateur d'applications React (create-react-app)
- Le DOM Virtuel et le processus de reconciliation (mise à jour efficace)

NEOACADEMY

3 - La syntaxe JSX et les composants

- Définition d'un élément React (types, attributs, éléments enfants)
- Connexion avec le DOM (ReactDOM.render())
- Une syntaxe innovante : JSX
- Le plugin Babel pour transformer JSX en JavaScript
- Les règles de syntaxe du JSX (expression intégrée, sécurité XSS, balise parente unique)
- Les attributs JSX
- Les composants : définition, avantages (réutilisabilité)
- Les composants en classe
- Les composants fonctionnels (approche récente)
- Imbrication de composants (balises imbriquées)

4 - Les propriétés (props)

- Définition (passage d'informations, lecture seule)
- Transmission de props
- Accès aux props dans les composants fonctionnels / classes
- La prop spéciale children

5 - L'état (state) et le cycle de vie des composants

- Définition (stockage des données, individualisation du composant)
- Initialiser l'état
- Utiliser la méthode setState et ses deux modes (synchrone/asynchrone)
- Le cycle de vie du composant
- Montage (componentDidMount)
- Mise à jour (componentDidUpdate)
- Démontage (componentWillUnmount)
- Meilleures pratiques (setState asynchrone, éviter setState dans le constructeur)

6 - Les Hooks (crochets)

- Définition et rôle
- Différences entre Hooks et composants en classe
- Le hook d'état (useState)
- Le hook d'effet (useEffect) et ses dépendances
- Différents modes du hook useEffect : initialisation, mise à jour, nettoyage
- Règles à respecter pour l'utilisation des Hooks
- Création de Hooks personnalisés (custom hooks)

7 - La gestion des événements

- Syntaxe pour gérer les événements dans JSX
- Méthodes pour gérer les événements (handlers)
- Techniques pour lier le contexte d'exécution au handler (bind(), fonctions fléchées, etc.)
- Objet événement (event object)
- Passage de paramètres supplémentaires aux handlers
- Transmission d'un handler en tant que propriété (props)

8 - Affichage conditionnel et listes

- Rendu conditionnel avec raccourcis logiques (&&, opérateurs ternaires)
- Manipulation de listes avec des fonctions avancées (map, filter, etc.)
- Utilisation des clés (key) pour optimiser le DOM Virtuel
- Les fragments React (pour grouper plusieurs éléments sans ajouter de nœud supplémentaire)

9 - La gestion des formulaires

- L'état du composant comme source unique de vérité (source of truth)
- Composant contrôlé (controlled component)
- Attribut universel de valeur : value
- Soumettre un formulaire
- Les composants non contrôlés (par exemple, input de type file)
- Les références (refs) pour accéder directement aux éléments DOM

10- Le routage et la navigation dans une application monopage

- Créer une SPA avec des URLs mémorisables (bookmarks)
- La bibliothèque react-router-dom (version 5)
- Le système de routage (Router)
- Les liens de navigation (Link)
- Les routes définies (Route)
- Le composant Switch pour gérer plusieurs routes
- Gestion des paramètres dans l'URL
- Navigation imbriquée entre différentes vues

11- Introduction à Redux et architecture flux

- Immutabilité des données partagées
- Les composants d'ordre supérieur (HOC)
- Problèmes liés à la gestion d'état dans une application complexe
- Les systèmes de gestion d'état (state management)
- Le pattern architecture flux (actions, dispatcher, store, etc.)
- Présentation de Redux et installation
- Les actions (events)
- Les réducteurs (reducers)
- Le magasin global (store)
- Intégration avec React (react-redux)
- Le composant Provider pour encapsuler l'application
- Les containers (contenant)
- Le connecteur HOC (Higher-Order Component) connect
- Méthodes pour lier l'état et dispatch : mapStateToProps et mapDispatchToProps

12 - La validation et les tests

- Introduction à Jest (framework de tests) : configuration, nettoyage, blocs describe/it
- La bibliothèque React Testing Library (nettoyage, rendu, simulation d'événements)

Modalités d'évaluation des acquis

Chaque stagiaire devra effectuer un test de connaissance sous la forme d'un questionnaire oral et écrit avec le formateur reprenant les principaux points de la formation.

Sanction visée

Certificat de réalisation.

Matériel nécessaire pour suivre la formation

L'apprenant aura à sa disposition le matériel informatique professionnel, logiciel prérequis et accès à l'environnement informatique de l'entreprise si besoin.

NEOACADEMY

267 Le Mont Roty 76160 FONTAINE SOUS PREAUX
Tél. 06.86.40.13.94 – boucourt.charlotte@neoacademy.fr
SAS au capital de 10000 € – RCS Rouen – SIRET 94003713800010 – NAF 70.22

Conditions de déroulement

Formation réalisée en présentiel ou en classe virtuelle Formation en INTRA/INTER.

Alternance d'apports de connaissances théoriques, puis mise en pratique.

Délais moyens pour accéder à la formation

La planification de votre formation sera finalisée et inscrite à notre calendrier selon vos disponibilités et contraintes.

Accessibilité aux personnes en situation de handicap

Lors de l'inscription à nos formations, nous étudions avec le candidat en situation de handicap et à travers un questionnaire les actions que nous pouvons mettre en place pour favoriser son apprentissage. Pour cela, nous pouvons également nous appuyer sur un réseau de partenaires nationaux préalablement identifiés.

Taux de réussite à la formation

Taux de satisfaction de la formation

Tarif

Nous consulter.