



Développeur Web HTML 5, CSS 3 et JavaScript

Vue.js 3 - Fonctionnalités avancées

2 jours (14h00) | ★★★★★ 4,6/5 | JVSVue-AV | Évaluation qualitative de fin de stage |
Formation délivrée en présentiel ou distanciel

Formations Informatique > Langages et développement > Développeur Web HTML 5, CSS 3 et JavaScript

Contenu mis à jour le 13/10/2023. Document téléchargé le 21/06/2024.

Objectifs de formation

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

- Décrire les aspects avancés du Vue.js
- Créer des composants plus réutilisables
- Développer une application plus optimisée.

Modalités, méthodes et moyens 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

Prérequis

Maîtriser JavaScript et connaître les fondamentaux de Vue.js.

Public concerné

Développeurs Vue.js souhaitant produire des applications Vue.js plus optimisées et qualitatives.

Cette formation :

- Est animée par un consultant-formateur dont les compétences techniques, professionnelles et pédagogiques ont été validées par des diplômes et/ou testées et approuvées par l'éditeur et/ou par M2i Formation
- Bénéficie d'un suivi de son exécution par une feuille de présence émarginée par demi-journée par les stagiaires et le formateur.

Programme

Jour 1

Approfondissement de Vue Router

- Présentation détaillée des fonctionnalités avancées de Vue Router
- Utilisation du mode "historique" et configuration server-side
- Gestion des paramètres d'URL et des routes imbriquées
- Navigation par le code et redirection
- Gestion des erreurs et affichage des messages flash
- Implémentation de la pagination
- Utilisation des gardes pour sécuriser les routes

Exemples de travaux pratiques (à titre indicatif)

- Créer une application Vue.js qui utilise Vue Router pour gérer la navigation entre différentes pages
- Mettre en place un système de pagination pour afficher une liste d'articles ou de produits
- Utiliser les gardes de navigation pour restreindre l'accès à certaines pages en fonction de l'état de l'utilisateur

Les slots

- Exploration approfondie des slots dans Vue.js 3
- Définition et utilité des slots
- Injection de contenu dans un template en utilisant les slots
- Utilisation des slots pour rendre les composants plus génériques et réutilisables
- Interaction entre les composants parents et les composants enfants via les slots
- Utilisation des props avec les slots
- Introduction à la directive v-slot et utilisation des slots nommés
- Compréhension de la portée des slots dans les composants Vue.js
- Utilisation des slots à nom dynamique pour une plus grande flexibilité dans la composition des composants

Exemples de travaux pratiques (à titre indicatif)

- Créer un composant de mise en page flexible qui utilise des slots pour afficher du contenu dynamique (en-tête, contenu principal, pied de page)
- Utiliser les slots nommés pour permettre aux utilisateurs de personnaliser l'apparence du composant en fonction de leurs besoins

Les composants asynchrones

- Présentation détaillée de l'importance et de l'utilisation des composants asynchrones dans Vue.js 3
- Utilisation de la fonction `defineAsyncComponent` pour charger les composants à la demande
- Gestion des erreurs et de l'état lors du chargement asynchrone des composants

Exemples de travaux pratiques (à titre indicatif)

- *Modifier une application existante pour charger certains composants de manière asynchrone en utilisant la fonction `defineAsyncComponent`*
- *Gérer l'état de chargement et les erreurs lors du chargement asynchrone des composants*

Jour 2

Vuex et Pinia

- Comparaison approfondie entre Vuex et Pinia dans le contexte de Vue.js 3
- Avantages et cas d'utilisation de chaque solution
- Création d'un store de données d'état avec Vuex dans une application Vue.js 3
- Exemple d'implémentation similaire avec Pinia et démonstration des différences

Exemples de travaux pratiques (à titre indicatif)

- *Créer un store Vuex ou Pinia pour gérer l'état global d'une application*
- *Utiliser les getters, mutations et actions pour effectuer des opérations sur l'état global depuis les composants de l'application*
- *Mettre en place des modules Vuex ou des stores Pinia pour organiser le code de l'application de manière modulaire*
- *Ecrire des tests unitaires pour les composants Vue.js en utilisant Jest*
- *Tester les méthodes, les propriétés calculées et les événements émis par les composants*
- *Utiliser les mocks pour simuler des dépendances externes et isoler les composants pendant les tests*

La performance

- Explication approfondie des améliorations de performance apportées par Vue.js 3
- Code splitting amélioré et chargement à la demande
- Utilisation des directives `v-once` et `v-memo` pour optimiser le rendu des composants
- Intégration du lazy loading des routes avec "dynamic import" pour améliorer les performances de chargement

Exemples de travaux pratiques (à titre indicatif)

- *Optimiser le chargement des routes en utilisant le lazy loading avec "dynamic import"*
- *Utiliser les directives `v-once` et `v-memo` pour réduire le nombre de rendus inutiles dans les composants*
- *Mesurer les performances de l'application avant et après l'optimisation pour évaluer l'impact des changements sur la vitesse de chargement et l'expérience utilisateur*

Les "Portals" et "Teleports"

- Présentation approfondie des concepts de "Portals" et "Teleports" dans Vue.js 3
- Utilisation de "Teleport" pour rendre des composants à l'extérieur de l'arbre DOM de l'application
- Implémentation de Portals pour un rendu à un autre emplacement de l'arbre DOM

Exemples de travaux pratiques (à titre indicatif)

- *Créer un composant modal réutilisable qui utilise "Portal" pour rendre son contenu à l'extérieur de l'arbre DOM de l'application*
- *Utiliser "Teleport" pour placer le composant modal à un emplacement spécifique de l'interface utilisateur, tel qu'une modale globale au-dessus de l'application*

Le contenu de ce programme peut faire l'objet d'adaptation selon les niveaux, prérequis et besoins des apprenants.

Modalités d'évaluation des acquis

- En cours de formation, par des études de cas ou des travaux pratiques
- Et, en fin de formation, par un questionnaire d'auto-évaluation

Accessibilité de la formation

Le groupe M2i s'engage pour faciliter l'accessibilité de ses formations. Les détails de l'accueil des personnes en situation de handicap sont consultables sur la page Accueil et Handicap.

Modalités et délais d'accès à la formation

Les formations M2i sont disponibles selon les modalités proposées sur la page programme. Les inscriptions sont possibles jusqu'à 48 heures ouvrées avant le début de la formation. Dans le cas d'une formation financée par le CPF, ce délai est porté à 11 jours ouvrés.