

## Vue.js 3 - Développement d'applications Web

Développer une application Single Page App (SPA) complète avec le framework JavaScript Vue.js

 Présentiel ou en classe à distance



3 jours (21 h)

Prix inter : 2.150,00 € HT

Forfait intra : 5.550,00 € HT

Réf.: LI270



Vue.js est un framework JavaScript permettant de créer des interfaces utilisateur interactives et performantes. Il se distingue par sa facilité d'apprentissage, sa flexibilité et son intégration harmonieuse avec d'autres bibliothèques ou projets. Utilisant un système de composants réactifs et une API intuitive, Vue.js est adapté aux petits projets comme aux applications complexes grâce à sa structure modulaire. Cette formation Vue.js a pour objectif de découvrir ses principes clés, d'installer et configurer un environnement avec vue-cli et Vite, et d'être initié aux API Option et Composition. Le programme explore les composants, les Single File Components (SFC), et les méthodes de liaison. Il inclut aussi la gestion des événements, des formulaires et de l'état de l'application avec Pinia.

### A qui s'adresse cette formation ?



#### Pour qui

- Développeurs web
- Intégrateurs
- Architectes logiciels
- Chefs de projet...



#### Prérequis

- Disposer d'une bonne connaissance pratique de HTML et maîtriser JavaScript (ES5 minimum)
- **Disposez-vous des connaissances nécessaires pour suivre cette formation ? Testez-vous !**

### Programme

#### 1 - Présentation de Vue.js

- Principes fondamentaux de Vue.js
- Comparaison avec d'autres frameworks (React, Angular)
- Utilisation de vue-cdn, vue-cli et de Vite
- Créer et tester une première application
- Architecture de projet  
Atelier

Installation de Vue.js, configuration et mise en place d'un projet Vue.js avec Vue CLI et Vite

#### 2 - Les fichiers de composants

- Les Single File Components (fichier.vue)
- Créer un composant : API Option vs API composition
- Lier un composant à l'application Le template

- Structuration et modularisation du code avec les modules ES (ECMAScript)
  - L'encapsulation de style
- Atelier

Création de composants SFC et les lier à une application

### **3 - Les syntaxes d'interpolation**

- Les données du composants (propriété data de l'API Option)
  - Introduction au Reactive API de l'API
  - Composition : présentation de ref et reactive
  - Opérateur moustache (interpolation) et expression JS
  - Injecter du texte (v-text) et du HTML (v-html)
  - Injecter sans prendre en compte les changements (v-once)
  - Insérer un composant (balise de composant)
- Atelier

Interpoler différents types de variables (tableaux, objets, chaînes de caractères contenant une balise HTML...)

### **4 - Rendu conditionnel**

- Les directives v-if, v-else et v-else-if
  - Affichage avec existence dans le DOM permanente avec v-show
  - Utiliser v-if avec la balise template
- Atelier

Affichage conditionnel en fonction de la valeur d'une variable

### **5 - Affichage de liste**

- La directive v-for et ses syntaxes (in / of)
  - L'attribut key
  - Immutabilité des éléments de tableau
  - Créer un range
  - Utiliser v-for avec la balise template
  - Priorité entre v-if et v-for
- Atelier

Créer une liste d'éléments avec v-for

### **6 - Lier des attributs**

- Lier un attribut avec v-bind
  - Class-binding avec v-bind:class
  - Style-binding avec v-bind:style (prefixer automatique)
  - Les arguments dynamiques (v-bind[...])
  - Syntaxe raccourcis (:attribut)
- Atelier

Créer un composant avec des classes conditionnelles qui changent en fonction de l'état du composant

### **7 - Gérer les événements**

- La directive v-on, Les modificateurs d'événement (stop, prevent, ...)
  - Les modificateurs de touches (enter, space, ...)
  - Créer des méthodes pour gérer vos événements
  - Les arguments dynamiques (v-on[...])
  - Syntaxe raccourcis (@event)
  - La gestion des événements avec l'API Composition
  - Hooks : événements liés au cycle de vie d'un composant (created, mounted, updated...)
- Atelier

Créer un composant qui affiche un message différent à chaque étape de son cycle de vie

### **8 - Les formulaires**

- Les directives v-model
- Liaison de valeur (checkbox, radio, select)
- Les modificateurs (lazy, number, ...)
- Les librairies de validation de formulaires tels que : vuelidate, vee-validate, yup...

Atelier

Lier des valeurs de formulaire avec les nouvelles directives

## 9 - Les composants avancés

- Passer des données à un composant avec v-bind
- Mise en cache de méthodes avec les propriétés calculées : methods VS computed, usage avec le Composition API
- Les watchers (watch) : utilisation des hooks de watch dans le Composition API
- Emettre des événements depuis un composant enfant : \$emit (API Option), defineEmits (API Composition), utilisation avec v-model
- Les slots nommés et les slots dynamiques
- Passer des données à un composant enfant : props (API Option), defineProps (API Composition)
- Passer des données aux composants enfants : provide, inject
- Les différentes solutions pour communiquer avec le backend (fetch, axios, websockets...)

Atelier

Utilisation des slots, props et \$emits pour la communication entre un composant parent et enfant

## 10 - Routing et navigation

- Installation et mise en place de vue-router
- Configuration du router
- Les routes
- Les liens (router-link)
- Les vues (router-view)
- Gérer les paramètres (utiliser les props)
- Redirections et alias
- Les navigations imbriquées

Atelier

Créer un menu dynamique

## 11 - Gérer l'état de son application avec Pinia

- Introduction à Pinia
- Installation et mise en place de Pinia
- Création du store
- Définition et accès à l'état
- Définition et accès aux actions
- Mutualiser les accès avec les getters
- Gérer des mutations asynchrones avec les actions

Atelier

Créer un store avec Pinia pour gérer l'état global d'une application

### Après la session

- Des vidéocasts pour revenir sur les points clés de la formation
- Des vidéos-tutos pour vous accompagner dans la mise en oeuvre de vos acquis



### Les objectifs de la formation

- Configurer et tester un projet Vue.js
- Définir et structurer les composants avec les API Composition et Option
- Utiliser efficacement les syntaxes d'interpolation
- Appliquer les directives de rendu conditionnel
- Naviguer entre différentes pages ou vues dans une application Vue.js avec Vue Router
- Gérer l'état d'une application avec Pinia



## Evaluation

- Pendant la formation, le formateur évalue la progression pédagogique des participants via des QCM, des mises en situation et des travaux pratiques. Les participants passent un test de positionnement avant et après la formation pour valider leurs compétences acquises.



## Les points forts de la formation

- La formation permet aux participants de configurer, tester et structurer des projets Vue.js avec les API Composition et Option, en utilisant des Single File Components.
- Une formation pratique : l'alternance de cours théoriques et de nombreux ateliers pratiques pour un ancrage durable des acquis.
- Toutes nos formations sont animées par des consultants-formateurs expérimentés et reconnus par leurs pairs.
- 85% des participants à cette formation se sont déclarés satisfaits ou très satisfaits au cours des 12 derniers mois.



## Dates et villes 2026 - Référence LI270



Dernières places disponibles



Session garantie

### Rouen

du 12 janv. au 14 janv.

du 4 mai au 6 mai

du 29 juin au 1 juil.

du 21 sept. au 23 sept.

### A distance

du 12 janv. au 14 janv.

du 16 févr. au 18 févr.

du 4 mai au 6 mai

du 29 juin au 1 juil.

du 21 sept. au 23 sept.

du 23 nov. au 25 nov.

### Paris

du 12 janv. au 14 janv.

du 16 févr. au 18 févr.

du 4 mai au 6 mai

du 29 juin au 1 juil.

du 21 sept. au 23 sept.

du 23 nov. au 25 nov.

### Aix-en-Provence

du 12 janv. au 14 janv.

du 4 mai au 6 mai

du 21 sept. au 23 sept.

du 23 nov. au 25 nov.

### Sophia Antipolis

du 12 janv. au 14 janv.

du 4 mai au 6 mai

du 29 juin au 1 juil.

du 21 sept. au 23 sept.

### Toulouse

du 12 janv. au 14 janv.

du 16 févr. au 18 févr.

du 29 juin au 1 juil.

du 21 sept. au 23 sept.

## Marseille

du 12 janv. au 14 janv.

du 4 mai au 6 mai

du 21 sept. au 23 sept.

du 23 nov. au 25 nov.

## Strasbourg

du 12 janv. au 14 janv.

du 4 mai au 6 mai

du 29 juin au 1 juil.

du 21 sept. au 23 sept.

## Lille

du 12 janv. au 14 janv.

du 4 mai au 6 mai

du 21 sept. au 23 sept.

du 23 nov. au 25 nov.

## Rennes

du 16 févr. au 18 févr.

du 29 juin au 1 juil.

du 21 sept. au 23 sept.

du 23 nov. au 25 nov.

## Nantes

du 16 févr. au 18 févr.

du 29 juin au 1 juil.

du 21 sept. au 23 sept.

du 23 nov. au 25 nov.

## Lyon

du 16 févr. au 18 févr.

du 29 juin au 1 juil.

du 4 mai au 6 mai

du 23 nov. au 25 nov.

## Bordeaux

du 16 févr. au 18 févr.

du 29 juin au 1 juil.

du 4 mai au 6 mai

du 23 nov. au 25 nov.