

# Formation : Campus Atlas - Vue.js - Développement d'applications web

*Cours pratique - 3j - 21h00 - Réf. AJC*

*Prix : 1830 € H.T.*

NEW

Dans un contexte de développement web moderne où les Single Page Applications (SPA) sont devenues incontournables, Vue.js s'impose comme un framework progressif et performant. Sa courbe d'apprentissage douce et son écosystème riche en font un choix privilégié pour le développement d'applications web modernes. Cette formation vise à doter les développeurs web, intégrateurs et architectes des compétences nécessaires pour développer et déployer des applications Vue.js professionnelles. Ce programme de formation est destiné aux salariés des branches professionnelles relevant de l'OPCO Atlas.

## Objectifs pédagogiques

**À l'issue de la formation, le participant sera en mesure de :**

- Maîtriser l'environnement du développeur JavaScript
- Comprendre les principes clés et les bonnes pratiques de Vue.js
- Connaître les outils indispensables au développement d'applications web dynamiques avec Vue.js
- Savoir développer une SPA (application monopage) avec Vue.js
- Réaliser des tests

## Public concerné

Pour les adhérents à l'OPCO Atlas : développeurs web, intégrateurs, architectes logiciel, chefs de projet...

## Prérequis

Avoir une bonne connaissance pratique de HTML et la maîtrise de JavaScript.

### PARTICIPANTS

Pour les adhérents à l'OPCO Atlas : développeurs web, intégrateurs, architectes logiciel, chefs de projet...

### PRÉREQUIS

Avoir une bonne connaissance pratique de HTML et la maîtrise de JavaScript.

### COMPÉTENCES DU FORMATEUR

Les experts qui animent la formation sont des spécialistes des matières abordées. Ils ont été validés par nos équipes pédagogiques tant sur le plan des connaissances métiers que sur celui de la pédagogie, et ce pour chaque cours qu'ils enseignent. Ils ont au minimum cinq à dix années d'expérience dans leur domaine et occupent ou ont occupé des postes à responsabilité en entreprise.

### MODALITÉS D'ÉVALUATION

Le formateur évalue la progression pédagogique du participant tout au long de la formation au moyen de QCM, mises en situation, travaux pratiques...

Le participant complète également un test de positionnement en amont et en aval pour valider les compétences acquises.

## Méthodes et moyens pédagogiques

### Méthodes pédagogiques

Pour optimiser le parcours d'apprentissage, des modules e-learning peuvent être fournis avant et après la session présentielle ou la classe virtuelle, sur simple demande du participant.

### Modalités d'évaluation

Le formateur évalue la progression pédagogique du participant tout au long de la formation au moyen de QCM, mises en situation, travaux pratiques...

Le participant complète également un test de positionnement en amont et en aval pour valider les compétences acquises.

## Programme de la formation

### 1 JavaScript, apprendre les bases du langage - Contenu digital learning

#### préformation

- Introduction.
- Basique du langage.
- Exécution de scripts.
- Objets JavaScript et callback.
- Manipulation du DOM.

#### Activités digitales

Cette formation en ligne apprend les bases essentielles de JavaScript telles que les fonctions, les conditions et les boucles, afin de développer de premières applications en toute autonomie. Les participants mettront leurs connaissances en pratique à travers la création d'une application de gestion de tâches (ToDoList) et seront prêts à aborder des frameworks comme AngularJS ou Node.js.

### 2 L'écosystème du développeur JS modern

- Features modernes d'ECMAScript.
- Intégration de TypeScript.
- Tooling moderne JavaScript.
- DevTools spécifiques Vue.js.

#### Travaux pratiques

JavaScript Moderne, pratiquer TypeScript, DevTools et debugging.

### 3 Utilisation de Vue.js avec Vite

- Fondamentaux de Vite et ses avantages.
- Architecture d'un projet Vue.js moderne.
- Gestion des dépendances et scripts.
- Bonnes pratiques de structuration.

#### Travaux pratiques

Installation et architecture du projet. Scripts et optimisation.

## MOYENS PÉDAGOGIQUES ET TECHNIQUES

• Les moyens pédagogiques et les méthodes d'enseignement utilisés sont principalement : aides audiovisuelles, documentation et support de cours, exercices pratiques d'application et corrigés des exercices pour les formations pratiques, études de cas ou présentation de cas réels pour les séminaires de formation.

• À l'issue de chaque formation ou séminaire, ORSYS fournit aux participants un questionnaire d'évaluation du cours qui est ensuite analysé par nos équipes pédagogiques.

• Une feuille d'émargement par demi-journée de présence est fournie en fin de formation ainsi qu'une attestation de fin de formation si le participant a bien assisté à la totalité de la session.

## MODALITÉS ET DÉLAIS D'ACCÈS

L'inscription doit être finalisée 24 heures avant le début de la formation.

## ACCESSIBILITÉ AUX PERSONNES HANDICAPÉES

Pour toute question ou besoin relatif à l'accessibilité, vous pouvez joindre notre équipe PSH par e-mail à l'adresse psh-accueil@orsys.fr.

## 4 Les composants

- Syntaxe et templating Vue.js.
- Architecture des composants.
- Gestion des styles et scoping.
- Communication entre composants.
- Patterns de composition avancée.
- Hooks de cycle de vie.

### Travaux pratiques

Créer des composants, composition avancée, cycle de vie et optimisation.

## 5 Répartir son code dans des composants

- Architecture des composants.
- Patterns de composition avancés.
- Réutilisation de la logique.
- Tests et validation.
- Bonnes pratiques de structuration.

### Travaux pratiques

Décomposer en composants et patterns de composants. Valider et optimiser.

## 6 Gestion de l'état avec Composition API

- Reactive et Ref en profondeur.
- Gestion des références complexes.
- Computed properties avancées.
- Watchers et leurs options.
- Pattern Provide/Inject.
- State management patterns.

### Travaux pratiques

État local et réactivité, patterns de communication, debugging et optimisation.

## 7 Vue Router et navigation

- Configuration avancée du router.
- Navigation programmatique.
- Guards et middleware.
- Route meta fields.
- Lazy loading de routes.
- Nested routes et views.
- Gestion des paramètres.

### Travaux pratiques

Configurer le routing. Routes avancées. Tester et valider.

## 8 Gestion d'état avec Pinia

- Architecture Pinia.
- Set up et configuration.
- Stores et state.
- Getters et actions.
- Mutations et state changes.
- Modules et composition.
- Plug-ins et extensions.
- DevTools integration.

### Travaux pratiques

Installer et configurer Pinia. Actions avancées. Tester et optimiser.

## 9 Tests et qualité du code

- Testing framework Vitest.
- Vue Test Utils.
- Component testing.
- E2E testing avec Cypress.
- Code coverage.
- CI/CD pour Vue.js.
- Code quality tools.

### Travaux pratiques

Tests unitaires, tests E2E, CI/CD et qualité.

## 10 Composants avancés

- Slots patterns avancés.
- Composants dynamiques.
- Transitions et animations.
- Rendu conditionnel optimisé.
- Portails et téléportation.
- Patterns de performance.

### Travaux pratiques

Composants dynamiques, transitions et animations, portails et optimisation.

## 11 Intégration API et gestion des données

- HTTP clients (Axios/Fetch).
- Gestion des requêtes.
- Error handling.
- Caching et optimisation.
- Interceptors.
- Data validation.
- State synchronization.

### Travaux pratiques

Installation de la couche API, gérer les données, valider et tester.

## 12 Performance et optimisation

- Code splitting avancé.
- Lazy loading intelligent.
- Optimisation du bundle.
- Performance du runtime.
- Caching strategies.
- Server-side rendering.
- Build optimisation.

### Travaux pratiques

Optimiser le bundle, performance runtime, optimiser le déploiement.

## 13 Production et déploiement

- Configuration de production.
- Sécurité application.
- Monitoring et logging.
- SEO et métadonnées.
- Maintenance continue.
- Stratégies de mise à jour.
- Documentation.

### Travaux pratiques

Configurer la production. Monitoring et SEO. Documenter le projet.

## 14 Vue.js, le framework JavaScript pour développer le front-end de ses

### applications web - Contenu digital learning post-formation

- Introduction.
- Utiliser Vue.js pour interagir avec le DOM.
- Mettre en œuvre la réactivité de Vue.js.
- Structurer une page en composants.

### Activités digitales

Cette formation en ligne d'2h33 apprend à créer des interfaces utilisateur riches et dynamiques avec le framework JavaScript progressif Vue.js. Afin d'illustrer et de mettre en pratique les différentes fonctionnalités, la vidéo présente en fil rouge la conception et la réalisation d'un jeu de rapidité et d'adresse. Pour commencer, les participants découvriront Vue.js et les cas dans lesquels il est à privilégier par rapport à d'autres frameworks tels que Angular ou React. Puis, ils étudieront la création d'une première instance de type Vue et son cycle de vie. Les participants verront ensuite comment utiliser Vue.js pour interagir avec le DOM. Ils manipuleront ainsi différentes propriétés, méthodes, attributs ou directives pour initialiser la grille du jeu, modifier l'affichage des cases, compter le nombre d'essais restants ou encore afficher un message de félicitations.

## Dates et lieux

### CLASSE À DISTANCE

2026 : 24 mars, 19 mai, 13 oct., 15 déc.

### METZ

2026 : 19 mai, 13 oct.

### PARIS LA DÉFENSE

2026 : 17 mars, 5 mai, 6 oct., 8 déc.

### NANCY

2026 : 19 mai, 13 oct.