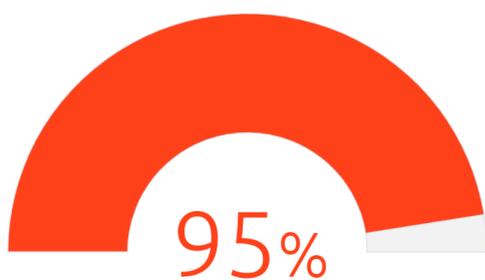


PROGRAMME DE FORMATION

Introduction à Vue.js



ÉVALUATION ÉQUIPE PÉDAGOGIQUE



CONTENU & ORGANISATION



OBJECTIFS

Vue.js est un framework JS permettant de créer des interfaces utilisateurs dynamiques. Il est facile à prendre en main, flexible et modulaire : il peut être utilisé de façon très minimale, ou dans une application complète de type Single Page Application ou Progressive Web App. Sa communauté et son écosystème en font un outil de choix pour construire des applications robustes et évolutives.

Avec cette formation vous comprendrez comment Vue fonctionne, prendrez en main toutes ses fonctionnalités ainsi que ses plugins Vue-Router et Vuex. Nous étudierons ensemble les bonnes pratiques, les pièges à éviter pour gagner du temps ainsi que les axes d'optimisation et d'amélioration dans le cadre d'un projet construit sur ce framework.

PRÉ-REQUIS

- Connaissances en HTML
- Connaissances en JavaScript
- Ordinateur portable à apporter

PUBLIC VISÉ

Développeuses et développeurs JavaScript ou PHP.

DURÉE

2 jours.



MOYENS ET MÉTHODES PÉDAGOGIQUES

- Présentation des concepts théoriques.
- Ateliers avec cas pratiques.

MODALITÉS D'ÉVALUATION

- Diagnostic préalable à l'inscription
- Evaluation en contrôle continu (ateliers et exercices pendant la formation)
- QCM en fin de formation



PROGRAMME

Introduction

Le Javascript moderne

- Déclaration de variables
- Déstructuration
- Fonctions fléchées
- Valeurs par défaut
- Les classes
- Interpolation
- Utilisation et création de promesses
- Manipulation d'objets
- Les modules

Première rencontre avec le framework Vue.js

- La réactivité
- Les limites de la réactivité
- Le DOM virtuel

L'instance Vue, le cœur du système

- Intégration minimale dans une page HTML
- Le fichier main.js
- Le composant principal

Les Single File Components

- Structure d'un fichier
- Philosophie des composants Vue

La création d'un projet Vue.js

- Les outils pour développer
- Découverte de l'outil vue-cli
- Typescript
- Class components



- Vue UI
- Architecture de dossiers et fichiers

Les plugins

- Utilisation d'un plugin tiers
- Création de plugin

Focus sur les composants Vue

Le template

- Structure d'un template
- Interpolation et data binding
- Attributs
- Les directives
- Les modificateurs
- Les filtres
- Data binding sur class et style

Les données réactives

Les propriétés calculées

Les méthodes

Rendu conditionnel

Les listes

Créer des directives

Les observateurs

Les styles



Les événements

Le cycle de vie d'un composant Vue

Communication entre les composants

Importer et utiliser des composants

Les props

Les attributs non props

Émettre et écouter un événement

Le bus d'événement global

Écouter des événements natifs

La modification de props

Accès à l'instance Vue principale

Accès à l'instance Vue parente

Références aux composants enfants

Injection de dépendances

Fonctionnalités plus avancées

Quelques modules usuels

Les slots

- Slots par défaut
- Slots nommés
- Slots avec portée



Les liaisons de formulaire

- La directive v-model
- Utilisation sur les types de champs
- Les modificateurs
- Créer un composant input personnalisé

Les Mixins

- Abstraire du code JS
- Ordre et priorité d'exécution

Composants dynamiques

Vue Router

- Utilisation du plugin de router
- Les composants du router
- Routes simples
- Routes nommées
- Routes dynamiques
- Informations meta
- Accès au routeur et à la route courante depuis les composants
- Redirection et alias
- Catch-all
- Les hooks de navigation par composant
- Les hooks de navigation par route
- Gérer les permissions avec vue-router

Axios

- Présentation et installation
- Création de l'instance Axios
- Création des services
- Les intercepteurs de requête et de réponse

Vuex

- Présentation et installation
- Description du store vuex
- Réactivité du store



- Bonnes pratiques
- Les modules Vuex
- Accès depuis les composants
- Les helpers Vuex
- Garder un store "à plat"
- Créer une factory de store

Tests unitaires

- Présentation de Jest
- Monter un composant
- Test des entrées/sorties de composant
- Exemples concrets de tests unitaires
- Les espions
- Les mocks
- Les stubs
- Les snapshots

Transitions et animations

- Les composants transition et transition-group
- Les états et classes de transition
- Réutilisation de transitions

Aller plus loin

Organiser et réorganiser son code quand l'application grossit

- Structure de dossiers et fichiers
- Garder des composants concis et réutilisables
- Utilisation de Typescript
- Découpage du store Vuex
- Le fichier main.js
- Abstraction avec les Mixins
- Composants renderless
- Nouvelle API composition



Optimisation des performances

- Rendu unique d'un composant
- Propriétés non réactives
- Composants fonctionnels
- Lazy loading
- Composants asynchrones
- Conserver l'état d'un composant

Créer une PWA avec Vue

NOTES

Nous pouvons ajuster la formation à vos besoins, en mettant l'accent sur un aspect plus particulier du développement.

PSH

Les personnes en situation de handicap sont invitées à contacter en toute confidentialité notre équipe pour définir les modalités d'accueil de la formation PSH@les-tilleuls.coop.

