



Empire
Training

Knowledge forges Empires



PLAN FORMATION

DÉVELOPPEMENT WEB



VUE JS



www.empire-training.tn



DESCRIPTION DE LA FORMATION VUE.JS

- ✓ **Vue.js est un framework Javascript basé sur le principe d'une application à page unique (SPA) qui consiste à réécrire une page Web existante avec de nouvelles données provenant du serveur sans recharger entièrement la page. L'interaction avec les utilisateurs permet donc d'atteindre d'excellentes performances et les applications SPA sont connues pour offrir également une bonne expérience utilisateur (UX).**
- ✓ **La simplicité de prise en main de Vue.js et sa facilité d'intégration avec d'autres bibliothèques ou projets existants font également partie de ses atouts pour une adoption par les équipes de développement Front.**





OBJECTIFS :

OBJECTIF OPÉRATIONNEL :

- ✓ **Savoir créer une interface utilisateur dynamique ou une application web monopage à l'aide du framework Vue.js.**

OBJECTIFS PÉDAGOGIQUES :

- ✓ **À l'issue de cette formation Vue.js, vous aurez acquis les connaissances et compétences nécessaires pour :**
 - **Améliorer une page existante de façon non obstrusive, comme le permettrait jQuery**
 - **Construire un templating dynamique et automatique de toute la page, comme le ferait Angular.JS 1 .**





OBJECTIFS :

OBJECTIFS PÉDAGOGIQUES :

- ✓ • **Créer vos propres widgets réutilisables et intégrer ceux de la communauté pour aboutir à une architecture complètement basée sur des composants réutilisables, comme avec React.JS.**
- **Comprendre la philosophie et les apports de Vue.js par rapport à d'autres frameworks (jQuery, Angular, React...)**
- **Savoir implémenter Vue.js au sein d'un environnement de développement JavaScript cohérent**
- **Savoir développer une Single Page App (SPA) à l'aide de Vue.js**
- **Être capable de construire un templating dynamique et automatique d'applications web**
- **Savoir créer et intégrer des widgets réutilisables**
- **Connaître et mettre en œuvre les bonnes pratiques de développement Vue.js**





CONTENU DU COURS VUE.JS

- ✓ **Avantages et inconvénients de Vue.js** ✨
 - Qu'est-ce qu'une application SPA (Single Page Application) ?
 - Les origines du framework Vue.js
 - Pourquoi choisir Vue.js?
 - Comparaison avec les frameworks (jQuery, Angular, ReactJS).
- ✓ **Utilisation simple de Vue** ✨
 - Utilisation simple, sans outils de build
 - Présentation de "hello world"
 - Savoir intégrer Vue.js incrémentalement
- ✓ **Utilisation sous forme de templates** ✨
 - Le one-way data binding
 - Connaître les méthodes d'utilisation de Vue.js
 - Les computed properties
 - Directives de contrôle de flux
 - Directives de gestion d'évènements
 - Présentation des principaux filtres
 - Les composants fournis avec Vue.js
 - Connaître les pièges les plus courants



- ✓ **Utilisation sous forme de composant** ✨
 - Découpler les widgets du HTML
 - Présentation du cycle de vie des composants
 - Modules complets "stand alone"
 - Connaitre la hiérarchie et la composition
 - Maîtriser les états et la communication entre composants
- ✓ **Routing avec Vue-router** ✨
 - La déclaration des vues
 - La manipulation de l'historique
 - La navigation
 - Routes imbriquées / Nested routes
 - Redirection
 - Gestion d'erreur
 - Les messages flash
- ✓ **Data store avec Vuex (introduction)** ✨
 - Les apports d'un gestionnaire d'état (état, vue, actions)
 - Zone de stockage centralisée pour tous les composants
 - Règles pour contrôler les mutations de l'état de l'application
 - Créer et appeler des actions
 - Cache et performance
 - La gestion des erreurs
 - Définir des accesseurs (getters)

- ✔ **Vue.js et les données extérieures (Call API)** ✨
Utiliser Axios pour consommer des API
Consommation d'une API REST avec Vue 3
- ✔ **Vue.js dans une stack JS moderne : npm, webpack et Babel** ✨
Installation via le package manager
La construction via webpack
Pre-processing avec Babel
- ✔ **Bonnes pratiques avec Vue.js** ✨
Emplacements et composants
Organisation du store Vuex
Appels d'API et validation des données
Comment simplifier notre base de code (mapState, mapGetters, mapMutations, mapActions)
Les factory API
La gestion des variables d'environnements (\$config)
Bonnes pratiques dans le nommage des commit (git)
Optimiser l'utilisation des bibliothèques (coût d'importation)





SCAN ME



N'hésitez pas à venir visiter notre site web.

www.empire-training.tn

