

Formation Nuxt.js : Initiation + Approfondissement

■ Durée :	5 jours (35 heures)
■ Tarif inter-entreprise :	3 775,00 CHF HT (standard) 3 020,00 CHF HT (remisé)
■ Public :	Développeurs Javascript
■ Pré-requis :	Bonnes connaissances de Javascript
■ Objectifs :	Réaliser des applications frontend en Nuxt.js
■ Modalités pédagogiques, techniques et d'encadrement :	• Formation synchrone en présentiel et distanciel. • Méthodologie basée sur l'Active Learning : 75 % de pratique minimum. • Un PC par participant en présentiel, possibilité de mettre à disposition en bureau à distance un PC et l'environnement adéquat. • Un formateur expert.
■ Modalité d'évaluation :	• Définition des besoins et attentes des apprenants en amont de la formation. • Auto-positionnement à l'entrée et la sortie de la formation. • Suivi continu par les formateurs durant les ateliers pratiques. • Évaluation à chaud de l'adéquation au besoin professionnel des apprenants le dernier jour de formation.
■ Sanction :	Attestation de fin de formation mentionnant le résultat des acquis
■ Référence :	SIT102899-F
■ Note de satisfaction des participants :	Pas de données disponibles
■ Contacts :	commercial@dawan.fr - 09 72 37 73 73
■ Modalités d'accès :	Possibilité de faire un devis en ligne (www.dawan.fr, moncompteformation.gouv.fr, maformation.fr, etc.) ou en appelant au standard.
■ Délais d'accès :	Variable selon le type de financement.

■ Accessibilité :

Si vous êtes en situation de handicap, nous sommes en mesure de vous accueillir, n'hésitez pas à nous contacter à referenthandicap@dawan.fr, nous étudierons ensemble vos besoins

Introduction

Présentation et comparaison des différents framework Javascript : Angular, React, Next, Vue, Nuxt...

Présentation de Node.JS et npm

Historique des versions de Vue et Nuxt

Présentation de la notion de Single Page Application et de Composant

Présentation de Vite et Webpack

Créer une application avec npm et la CLI

Découvrir ECMAScript

Spécification ECMAScript

Fonctions fléchées,

Design patterns

Définition de modules

Système natif de gestion des modules, chargement

Gestion des dépendances

La communication asynchrone (Promesses, async await)

Créer une application Nuxt JS

Les différentes manières de créer une application

Choisir son template

Comprendre l'architecture d'un projet

Concevoir des Single Files Components

Présentation des composants monofichier

Comprendre la syntaxe de template : interpolation et data binding

Afficher du rendu conditionnel

Afficher des listes

Comprendre la Composition API de Vue 3

Composition vs Option API

La fonction setup

Surveiller l'état des données

Gérer les événements

Créer des méthodes

Utiliser les propriétés calculées

Surveiller ses données avec les watchers

Communiquer entre les composants

Créer une architecture de composants parents / enfants

Passer des informations entre les composants : les props et les slots

Créer des événements personnalisés

Concevoir ses pages

Présentation de Vue Router

Créer ses pages et configurer ses routes

Créer des pages imbriquées

Passer des informations avec les paramètres

Protéger ses routes avec des middlewares

Gérer les formulaires :

Récupérer les données d'un formulaires

Accéder aux éléments du DOM avec Template Ref

Découvrir les outils de validation externes : Vee-validate, Vuelidate

Atelier: Créer un formulaire avec validation

Communiquer avec une API

Le helper \$fetch,

useFetch, useLazyFetch,

useAsyncData, useLazyAsyncData,

Rafraichir les données

Atelier: Afficher des données depuis une API externe

Gérer les données avec les stores

Présentation du concept de store, et des différents outils existants

Mise en place de store : Vuex, Pinia

Gérer les données dans le store : état, modifications, actions asynchrones

Atelier: Mettre en place un système de panier avec le store

Mettre en place le Server Side Rendering et améliorer le SEO

Qu'est-ce que le SRR,

Configurer son application,

Gérer les entêtes et les méta informations