

Mis à jour le 17/03/2026

S'inscrire

Formation Svelte

3 jours (21 heures)

Présentation

La formation [Svelte](#) vous présentera ce framework JavaScript open source qui permet de développer une application avec une syntaxe minimaliste pour ensuite la déployer.

Montez en compétences, en maîtrisant les concepts et la syntaxe de Svelte. Vous découvrirez la raison pour laquelle Svelte est le framework à utiliser aux dépens de ses concurrents.

Pendant cet enseignement, vous serez apte à contrôler les états et les différents stores. Cela vous permettra de mettre en pratique des moyens pour l'élaboration de votre propre store personnalisé.

Apprenez à mettre en place des animations et des transitions dans l'objectif d'améliorer l'ergonomie de l'application.

Comme toutes nos formations, celle-ci vous présentera **la dernière version stable** de la technologie et ses nouveautés.

Objectifs

- Comprendre la préparation de l'environnement de développement et l'installation de Node.js
- Acquérir une compréhension approfondie de la syntaxe de base de Svelte 5 et la création de projets avec Vite
- Maîtriser le système de réactivité
- Appliquer les concepts de conditions, boucles et gestion événementielle
- Explorer l'utilisation des composants, des snippets, de la communication parent/enfant par callback props
- Comprendre la gestion d'état avec le contexte, les stores et les runes
- Maîtriser les animations, transitions et la gestion des formulaires
- Développer des applications full-stack avec SvelteKit

Public visé

- Développeurs
- Lead Developer

Pré-requis

- Connaissance du langage JavaScript
- Connaissance en HTML et CSS

Programme de notre formation Svelte

Introduction à Svelte

- Préparation de l'environnement de développement
- Installation ou mise à jour de Node.js
- Les extensions utiles dans VS Code
- Le site officiel de Svelte et la documentation interactive
- Premiers pas avec Svelte via le Playground
- Création d'un premier projet Svelte avec Vite
- Structure d'un projet Svelte : fichiers .svelte, scripts et styles
- Ajout d'un premier composant

La syntaxe de base de Svelte

- Création d'un squelette Svelte standard avec Vite
- Les interpolations (expressions dans le markup)
- Interpolations en texte brut ou en HTML
- Les attributs dynamiques
- Les raccourcis d'attributs
- Classes conditionnelles
- Styles conditionnels

Les composants

- Première approche des composants
- Déclaration de props
- Valeurs par défaut et props optionnelles
- Props bindables
- Communication enfant vers parent avec des callback props (fonctions passées en props) •
Remplacement de createEventDispatcher par des callback props
- Chaînage de la communication entre composants
- Exercice

Conditions, boucles et gestion événementielle

- Exécution de code HTML conditionnellement
- Boucles each
- Deux paramètres dans une boucle each
- Parcours d'objets JSON avec each
- Blocs clés pour forcer le re-rendu
- Gestion de promesses dans le template
- Gestion événementielle
- Les modificateurs d'événements

Les Runes et la réactivité

- Présentation du système de runes dans Svelte 5
- Déclaration d'état réactif avec \$state
- Réactivité profonde vs superficielle : \$state vs \$state.raw
- Valeurs dérivées avec \$derived
- Calculs dérivés complexes avec \$derived.by
- Effets de bord avec \$effect
- Effets de pré-rendu avec \$effect.pre
- Débogage de la réactivité avec \$inspect
- Comparaison avec l'ancienne syntaxe réactive (\$:)

Snippets et composition

- Introduction aux snippets (remplacement des "slots")
- Déclaration de snippets et rendu de snippets
- Le snippet spécial children
- Snippets nommés
- Snippets optionnels et rendu conditionnel
- Passage de données aux snippets
- Gestion des erreurs avec <svelte:boundary>
- La directive {@attach} pour les actions sur éléments

Contexte propagé

- Partage de données avec le contexte (setContext / getContext / createContext) • Portée du contexte et limites

Les stores de Svelte

- Concept fondamental d'un store
- Création d'un store writable
- Stores en lecture seule (readable)
- Stores dérivés (derived)
- Auto-souscriptions avec le préfixe \$
- Contrôle des abonnements et prévention des fuites mémoire
- Déplacement de la logique métier dans le store

Gestion des formulaires

- Data binding avec bind:value
- Input de type text et input de type number
- Input de type radio et de type checkbox
- Select et option
- Les autres balises input (textarea, range, etc.)
- Binding avec un composant (\$bindable)
- Validation d'un formulaire côté front

Animations et transitions

- Transitions d'éléments HTML (fade, fly, slide, scale, blur, draw)
- Personnalisation des paramètres de transition
- Capturer le début et la fin d'une transition
- Utiliser des transitions d'apparition et de disparition différentes (in: / out:)
- Animation de nombres et de valeurs avec les classes Spring et Tween
- Action sur une propriété CSS inline avec Spring
- Animation de listes avec la directive animate:

Introduction et architecture

- Présentation et architecture d'une application SvelteKit
- Création d'une application SvelteKit from scratch

Routing et navigation

- Système de routing basé sur le système de fichiers
- Pages et layouts (+page.svelte, +layout.svelte)
- Layouts imbriqués et layouts spécifiques à un groupe de pages
- Définition de règles de styles globales
- Routes dynamiques et paramètres de route

Chargement de données

- Charger des données côté serveur avec les fonctions load (+page.server.ts, +layout.server.ts)
- Charger des données côté client avec les fonctions load (+page.ts, +layout.ts)
- Multiples sources de données
- Routes API avec +server.ts

Formulaires et gestion des erreurs

- Gestion des erreurs (+error.svelte, error())
- Form actions pour le traitement des formulaires côté serveur

Configuration et déploiement

- Hooks (handle, handleError, handleFetch dans hooks.server.ts)
- Options de page (ssr, csr, prerender, trailingSlash)
- Adapters de déploiement (adapter-auto, adapter-node, adapter-static, adapter-vercel, etc.)
- Mode SPA avec adapter-static et ssr=false

Sociétés concernées

Cette formation s'adresse à la fois aux particuliers ainsi qu'aux entreprises, petites ou grandes, souhaitant former ses équipes à une nouvelle technologie informatique avancée ou bien à acquérir des connaissances métiers spécifiques ou des méthodes modernes.

Positionnement à l'entrée en formation

Le positionnement à l'entrée en formation respecte les critères qualité Qualiopi. Dès son inscription définitive, l'apprenant reçoit un questionnaire d'auto-évaluation nous permettant d'apprécier son niveau estimé sur différents types de technologies, ses attentes et objectifs personnels quant à la formation à venir, dans les limites imposées par le format sélectionné. Ce questionnaire nous permet également d'anticiper certaines difficultés de connexion ou de sécurité interne en entreprise (intraentreprise ou classe virtuelle) qui pourraient être problématiques pour le suivi et le bon déroulement de la session de formation.

Méthodes pédagogiques

Stage Pratique : 60% Pratique, 40% Théorie. Support de la formation distribué au format numérique à tous les participants.

Organisation

Le cours alterne les apports théoriques du formateur soutenus par des exemples et des séances de réflexions, et de travail en groupe.

Validation

À la fin de la session, un questionnaire à choix multiples permet de vérifier l'acquisition correcte des compétences.

Sanction

Une attestation sera remise à chaque stagiaire qui aura suivi la totalité de la formation.