



Formation Svelte et SvelteKit | en ligne

I. Formation Svelte et SvelteKit | en ligne: Introduction et premiers pas

- Préparation de l'environnement de développement
- Utilisation des extensions utiles dans VS Code
- Exploration du site officiel de Svelte
- Premiers pas avec Svelte via le REPL
- Ajout d'un composant à une application Svelte
- Installation ou mise à jour de Node.js
- Création d'un premier projet Svelte avec Vite

II. Syntaxe de base de Svelte

- Création d'un squelette Svelte standard avec Vite
- Faire des interpolations dans les templates Svelte
- Exercice pratique pour mettre en pratique les interpolations
- Introduction aux variables réactives et au data binding
- Création et utilisation de plusieurs composants dans une application Svelte
- Utilisation d'un objet JSON pour les props des composants
- Passage de fonctions d'un composant à un autre
- Interpolations en texte brut ou en HTML
- Classes conditionnelles pour le rendu conditionnel

III. Conditions, boucles et gestion événementielle

- Exécution de code HTML conditionnellement avec les instructions conditionnelles
- Bucles each pour itérer sur des éléments
- Faire une boucle each avec deux paramètres
- Parcours d'objets JSON avec each pour afficher des listes dynamiques
- Gestion événementielle de la souris avec les écouteurs d'événements
- Gestion événementielle du clavier avec les écouteurs d'événements
- Utilisation des modificateurs d'événements pour modifier le comportement par défaut

IV. Les composants

- Première approche des composants et de leur utilisation
- Exploration approfondie de l'utilisation des composants dans Svelte
- Passage de données d'un composant enfant à son parent avec des Custom Events
- Transmission de données au parent avec un Custom Event
- Chaînage d'événements de composant à composant pour une communication indirecte
- Chaînage d'événements du DOM pour interagir avec des éléments HTML

V. Slots et Lifecycle hooks

- Utilisation des slots simples pour insérer du contenu dynamique dans un composant
- Utilisation des slots nommés pour avoir plusieurs emplacements définis dans un composant
- Définition d'un slot par défaut lorsque aucun contenu n'est fourni
- Affichage conditionnel de slots en fonction des conditions définies
- Utilisation conjointe de slots et de props pour une plus grande flexibilité
- Exploration des Lifecycle Hooks pour interagir avec le cycle de vie d'un composant
- Interfaçage d'une API REST dans le hook onMount pour récupérer des données externes
- Bloc `{#await}` pour gérer les opérations asynchrones
- Interfaçage de la base de données The Movie Database pour récupérer et afficher des informations de films
- Suite de l'interfaçage de la base de données The Movie Database pour explorer d'autres fonctionnalités

VI. Gestion des formulaires

- Utilisation des inputs de type text et number pour récupérer des valeurs
- Inputs de type radio et checkbox pour les sélections multiples
- Balises `Select` et `Option` pour les listes déroulantes
- Exploration des autres balises `input` pour gérer différents types de données
- Binding avec un composant pour mettre à jour les données en temps réel
- Validation d'un formulaire côté front pour assurer des données valides

VII. Les stores de Svelte

- Introduction aux stores de Svelte pour gérer l'état global de l'application
- Utilisation d'un premier exemple de store pour stocker des données partagées
- Éviter les fuites mémoire avec `unsubscribe()` pour libérer les ressources
- Faciliter l'accès au store en utilisant des méthodes utilitaires
- Ajout d'une fonctionnalité pour supprimer un livre à l'aide d'un store
- Déplacement de la logique programmatique dans le store pour une meilleure organisation
- Exploration des `Readable` stores pour une gestion avancée de l'état

VIII. Animations et transitions

- Animation de nombres avec le module `motion` pour des transitions fluides
- Action sur une propriété CSS inline pour créer des animations personnalisées
- Utilisation du store `Spring` pour des animations réactives
- Animations d'éléments HTML avec le module `transition` pour des transitions élégantes
- Suite d'animations d'éléments HTML avec le module `transition` pour des

effets avancés

- Capturer le début et la fin d'une animation pour effectuer des actions spécifiques
- Utiliser des animations d'apparition et de disparition différentes pour un effet visuel raffiné

IX. Une application SPA avec Svelte

- Création d'une application SPA (Single Page Application) pour une expérience utilisateur fluide
- Utilisation d'un routeur basé sur le hashing pour gérer les différentes vues
- Passage de paramètres dans l'URL pour des pages dynamiques
- Création de boutons et de liens pour naviguer vers la page précédente dans les pseudo-pages

X. SvelteKit

- Exploration d'une application de démonstration de SvelteKit pour une utilisation avancée
- Compréhension de l'architecture d'une application SvelteKit pour une meilleure organisation
- Création d'une application SvelteKit from scratch pour une personnalisation complète
- Utilisation de props dans les pages pour la flexibilité des données
- Création d'un composant de layout pour une mise en page cohérente
- Définition de règles de styles globales pour une apparence uniforme
- Utilisation d'un layout spécifique à une ou plusieurs pages pour des mises en page différentes
- Chargement des données de façon asynchrone pour une performance optimale
- Gestion de plusieurs sources de données pour une intégration avancée
- Création de routes dynamiques pour une navigation flexible
- Gestion des erreurs dans une route dynamique pour une meilleure expérience utilisateur

Conclusion sur la Formation Svelte et SvelteKit

Nous contacter:
Doussou Formation
Email: info@doussou-formation.com
<http://doussou-formation.com>

