Wat is Svelte?

Svelte is een component-based JavaScript user interface framework.

Het is vergelijkbaar met JavaScript-frameworks zoals React en Vue, die als doel hebben om eenvoudig interactieve gebruikersinterfaces te bouwen via compactere en leesbaardere code.

Maar er is een cruciaal verschil: Svelte maakt geen gebruik van de virtual dom, waar de andere frameworks dat wel doen. Svelte voegt tijdens het compileren code toe om de DOM te gaan updaten, ipv gebruik te maken van runtime technieken (virtual dom). Zo zijn de bundle sizes veel kleiner in Svelte applicaties wat de performance ten goede komt.

Svelte maakt ook gebruik van een Store module (gelijkaardig aan VueX), waardoor je heel gemakkelijk de state kan delen in verschillende componenten.

Wat zijn de mogelijkheden van Svelte?

* Truly reactive framework -> two way data binding,
* Easy to learn -> Svelte components lijken op html, maar met script en style tags. Makkelijke en leesbare code
* Super fast and small bundle sizes -> only the features that you use in the app that will be included in the bundle.

Verschil tussen svelte en andere traditionele frameworks?

Svelte maakt geen gebruik van de virtual DOM, omdat de code op voorhand wordt geïnterpreteerd. De code wordt dus maar 1 keer gerund en niet elke keer als er een state veranderd, wat een serieuze performance boost geeft.

De code wordt maar 1 keer gerund, maar Svelte heeft wel een oplossing gevonden voor bepaalde dingen reactive te houden. Zo kan je gebruik maken van het reactive commando “$:”. Reactive statements worden uitgevoerd voor de component update, wanneer de waarden (variabelen) waarvan ze afhankelijk zijn, zijn gewijzigd.

Scoped css -> elk component heeft zijn eigen style tag, waardoor er geen name collisions bestaan tussen de verschillende components.

Wat zijn de voor- en nadelen van Svelte t.o.v. andere traditionele frameworks?

**Pros:**

Simple Component Model -> Je moet vaak minder lijnen code schrijven om het zelfde te bekomen, in vergelijking met bv: vue, react of angular.

Solid Performance Benchmarks -> Svelte overklast de andere frameworks in performance benchmarks.

Small Application Bundle Sizes -> Zelfde als het vorige scenario, maar komt er nog bij dat optionele features, zoals stores en transitions niet in de bundle worden geïnclude als ze niet gebruikt worden.

Shallow Learning Curve -> Zeer gemakkelijk om te starten met Svelte, zeker als je al gewend bent om met component based frameworks te werken. Je kan ook een interactieve tutorial volgen op de Svelte website en er zijn ook tal van voorbeelden.

**Cons:**

Limited Adoption -> Er zijn nog maar weinig bedrijven die Svelte gebruiken. React, Vue en Angular hebben nog altijd de bovenhand.

Immature Developer Tooling -> Er is geen DevTools equivalent voor Svelte om applicaties te gaan debuggen na het compileren.

Small Selection of Third-Party Component Libraries

Doesn’t Play Well With Others -> werkt niet goed samen met andere soorten dan javascript, zoals Typescript wordt niet officieel ondersteunt.

Wat is de leercurve/moeilijkheidsgraad van Svelte?

De leercurve is zeker en vast niet zo steil. Doordat Svelte gebruik maakt van makkelijke en leesbare code, met scoped css.

Welke ontwikkelomgeving past het beste voor een svelte WebApp te maken?

Om code te schrijven, heb je een goede editor nodig. De meest populaire keuze is Visual Studio Code of VSCode, en terecht - het is goed ontworpen en volledig uitgerust, en heeft een schat aan extensies (waaronder een voor Svelte, die syntax highlighting en diagnostische berichten biedt wanneer je componenten schrijft) .

Wat zijn de unieke elementen van Svelte, waar proberen ze zich te onderscheiden van hun concurrenten?

Gebruiksvriendelijkheid en gemakkelijke leercurve: Svelte is zeer makkelijk te leren, zeker als je al ervaring hebt met component based frameworks. Svelte maakt gebruik van zeer gemakkelijke en leesbare code.

Performance: Svelte overklast alle andere frameworks in performance, doordat ze de code gaan compilen tijdens de compile time, ipv gebruik te maken van runtime technieken (bv: virtual DOM diffing). Zo zijn de bundle sizes veel kleiner in Svelte applicaties wat de performance ten goede komt.

Roadmap?

* Svelte native community project, voor het schrijven van IOS en Android apps in Svelte.
* Typescript support

Bronnen:

<https://dev.to/mstamstrom/sveltejs-the-next-big-ui-framework-4n0e>

<https://svelte.dev/blog/svelte-3-rethinking-reactivity>

<https://svelte.dev/docs>

<https://www.credera.com/blog/technology-solutions/svelte-is-this-next-generation-javascript-framework-production-ready/>

Credera is a full-service management and IT consulting firm