

SVELTE

Pregled i praktična implementacija Svelte framework-a u modernim web projektima

Viši programski jezici i RAD alati



SVELTE FRAMEWORK

Svelte je kompajler za kreiranje reaktivnih web aplikacija i interfejsa.

Svelte je besplatan, open-source okvir za front-end razvoj koji je kreirao Rich Harris, a održava ga Svelte core tim. Za razliku od klasičnih JavaScript biblioteka, Svelte kompajlira HTML šablone u specifičan kod koji direktno manipuliše DOM-om, smanjujući veličinu fajlova i poboljšavajući performanse. Kompajler automatski ažurira podatke i UI elemente kad god dođe do promjena.

Za razliku od okvira kao što su React i Vue, Svelte izbjegava virtuelni DOM i obradu u runtime-u, čime smanjuje opterećenje pretraživača.

```
1
2 <script lang="ts">
3   import { store } from "../store";
4   let edit: boolean = false;
5 </script>
6
7 <main>
8   {#if edit}
9     <label for="name">Name:</label>
10    <input id="name" type="text" bind:value={$store.name} /><br />
11    <label for="age">Age:</label>
12    <input id="age" type="number" bind:value={$store.age} /><br />
13    <label for="admin">Admin:</label>
14    <input id="admin" type="checkbox" bind:checked={$store.admin} />
15  {:else}
16    Name: {$store.name}<br />
17    Age: {$store.age}<br />
18    Admin: {$store.admin ? "Yes" : "No"}
19  {/if}
20 </main>
21
22 <button on:click={() => (edit = !edit)}>
23   {edit ? "Close" : "Edit"}
24 </button>
```

Prikaz jednostavne Svelte sintakse:

<script> - JavaScript

<main> - HTML

<style> - CSS

KORIST SVELTE FRAMEWORK-a

Svelte se može koristiti za razvoj dijelova interfejsa ili cijelih aplikacija. Može se postepeno integrisati u postojeće projekte ili koristiti od nule. Posebno je pogodan za:

1. Web aplikacije za uređaje sa slabijim performansama zbog manjih paketa i manje potrošnje resursa.
2. Visoko interaktivne stranice i složene vizualizacije zahvaljujući performansama bez runtime opterećenja.
3. Razvojne timove sa osnovnim znanjem web tehnologija, jer ima plitku krivu učenja.
4. Progresivne web aplikacije, koje se brzo učitavaju i funkcionišu offline.
5. SSR aplikacije, poput blogova i e-commerce sajtova, zbog bolje SEO optimizacije i brzine.
6. IoT kontrolne ploče, jer je efikasan u radu sa osvježavanjem podataka.
7. Aplikacije sa animacijama i tranzicijama, jer Svelte ima ugrađenu podršku za njih.

SvelteKit, dodatni okvir, donosi moderne funkcionalnosti poput SSR-a, rutiranja i offline podrške.



Svelte vs. React, Angular, Vue

Svelte je **KOMPAJLER** i nije framework, jer Svelte kompajluje naš kod za production u trenutnom vremenu pravljenja aplikacije tj. tokom kucanja samog koda i sav naš kod premješta u JavaScript Bundle za daljnu optimizaciju.

Svelte ne ubacuje dodatne skripte i biblioteke u naše aplikacije, što ih čini znatno bržima.



U poređenju sa drugim framework-ovima poput React-a, Angular-a i Vue-a, Svelte se ističe kao kompajler, a ne tradicionalni framework zasnovan na biblioteci. Dok React ima veliku podršku zajednice, Angular nudi stabilnost i moćne alatke, a Vue je svestran, Svelte koristi kompajliranje za optimizaciju performansi.

Svelte KOMPONENTE I SINTAKSA

Svelte komponente kombinuju HTML, CSS i JavaScript unutar jednog fajla sa ekstenzijom .svelte. Struktura se sastoji od tri dijela:

HTML: Definiše UI, kao što su elementi koji će biti prikazani korisnicima.

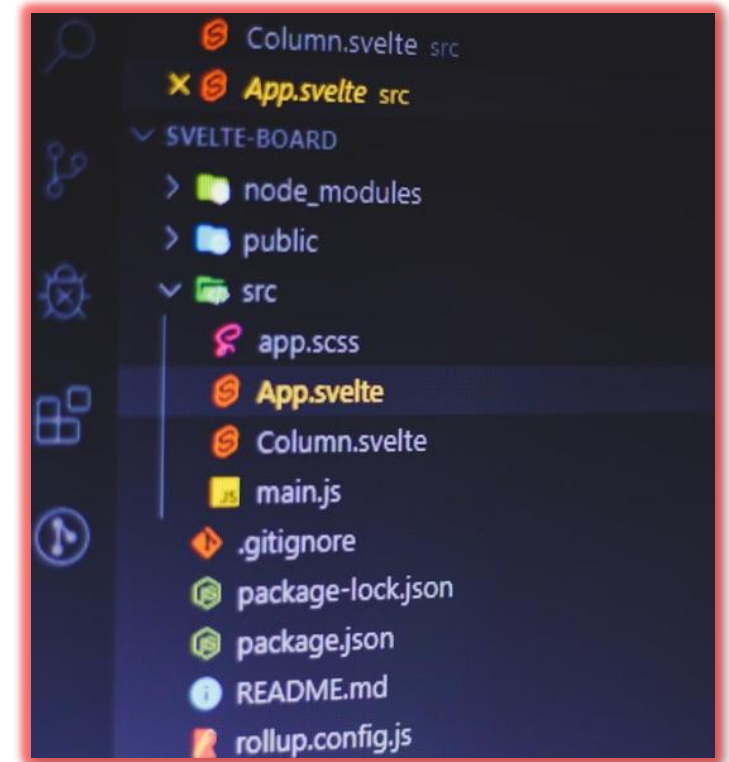
```
trenkic.svelte > ...
1
2 <main>
3   <h1>Hello, {name}!</h1>
4 </main>
```

CSS: Stilovi su lokalni za komponentu i ne utiču na ostale dijelove aplikacije.

```
trenkic.svelte > ...
1 <style>
2   h1 {
3     color: blue;
4   }
5 </style>
```

JavaScript: Definiše logiku komponente i koristi "reactive" promjenljive koje automatski ažuriraju UI.

```
trenkic.svelte > </script>
1 <script>
2   export let name = 'World';
3 </script>
```



BINDOVANJE PODATAKA

Svelte omogućava bindovanje podataka i reagovanje na događaje uz direktive poput **bind:value** i **on:event**.

bind:value : Promijena vrijednosti promjenljive automatski ažurira UI.

```
trenkic.svelte > ...
1  <input bind:value={name}>
2  <p>Your name is: {name}</p>
```

on:event : Upravljanje korisničkim akcijama poput klika.

```
trenkic.svelte > ...
1  <button on:click={handleClick}>Click me</button>
2  <script>
3    function handleClick() {
4      alert('Button clicked!');
5    }
6  </script>
```

Stilizacija je lokalna, a dinamičke klase omogućavaju promjene stila u zavisnosti od stanja.

```
trenkic.svelte > ...
1  <style>
2    .highlight {
3      background-color: yellow;
4    }
5  </style>
6  <p class:highlight={isHighlighted}>This is a highlighted paragraph.</p>
7  <script>
8    let isHighlighted = true;
9  </script>
```

PREDNOSTI I MANE Svelte FRAMEWORK-a



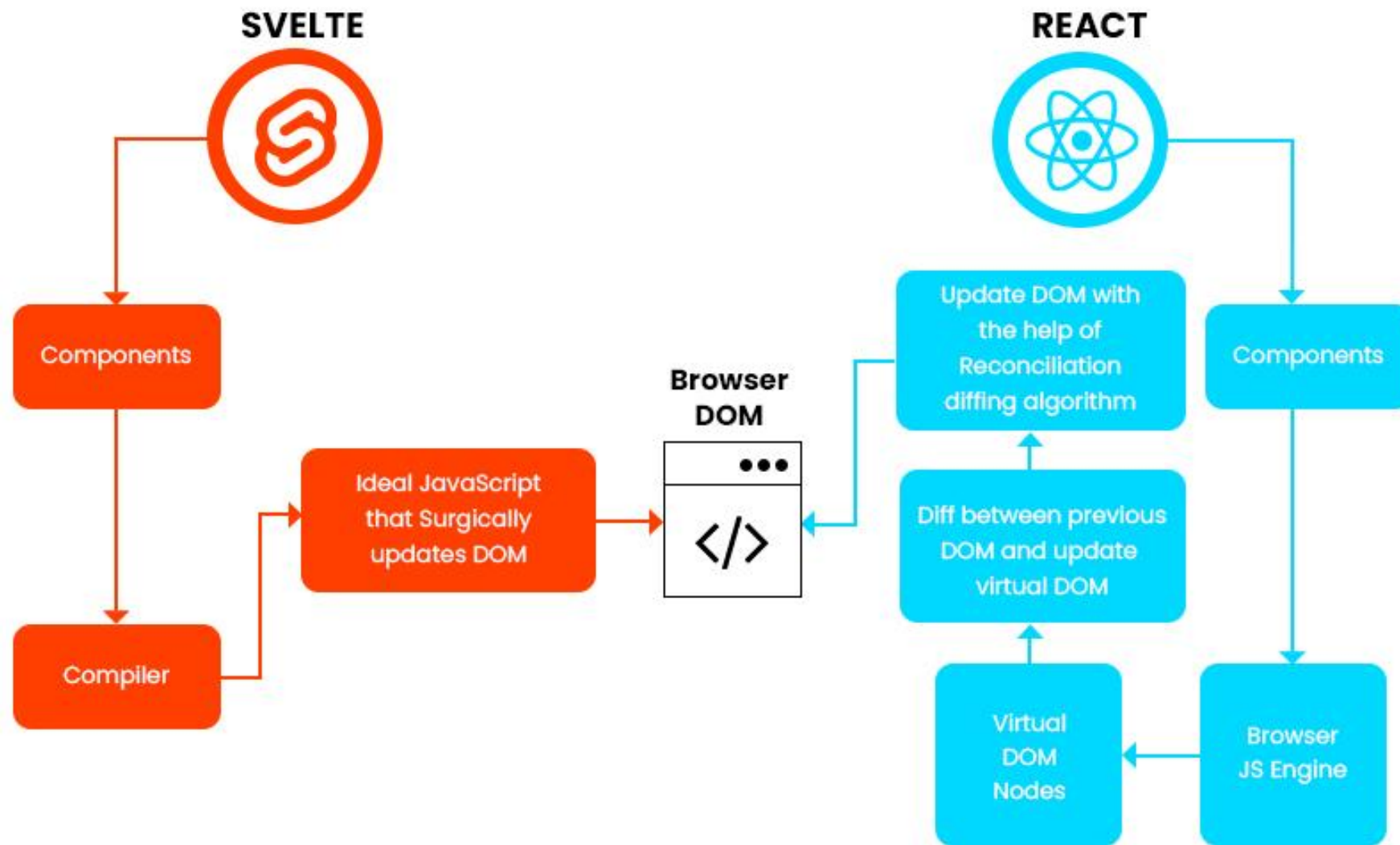
PREDNOSTI

Svelte pruža visoke performanse jer kompajlira komponente u optimizovani JavaScript koji direktno manipuliše DOM-om, zaobilazeći potrebu za virtuelnim DOM-om, što ubrzava ažuriranje interfejsa. Generiše manji izlazni kod jer uključuje samo potrebne funkcionalnosti, što smanjuje veličinu fajlova i ubrzava učitavanje stranica. Sintaksa je jednostavna i omogućava kombinovanje HTML-a, CSS-a i JavaScript-a u jednom fajlu, olakšavajući održavanje i učenje.

MANE

Zbog svoje novine, Svelte ima manji ekosistem biblioteka u poređenju sa React-om i Vue-om, što otežava pronalazak specifičnih dodataka. Kompajlerski pristup može dodati kompleksnost u razvoju, zahtijevajući rad sa build alatima i razumijevanje kompajliranja koda. Manje rasprostranjenost otežava pronalazak resursa i podrške. Također, prelazak na Svelte može zahtevati velike promijene u postojećim projektima, dok je njegova dugoročna podrška još uvek neizvjesna.

Svelte i React MANIPULACIJA DOM-a



HVALA VAM PA PAŽŇJI !!!