

Next.js and Svelte - Why should you use them?

Victor Pivtoran, 17/12/21

Firstly, what are Next.js and Svelte?

Next.js and Svelte are not exactly similar. Next.js is a framework built on top of Node.js. It's purpose is to support React-based web applications, enabling new features and easing the use of Node.js and React. Svelte, on the other hand, is a front-end compiler. A svelte application does not include a framework script.



Important features from Next.js

As mentioned at the beginning of the article, Next.js enables new features available for use within a React application. What exactly are these features, and why are they useful? Well, Next.js provides a common structure that allows you to easily build a frontend React application, and transparently handles server-side rendering for you. Handling server-side rendering is extremely important in a React-based application, as it makes the user experience much smoother and faster. With Next.js, you can render React components on the server side, before sending the HTML to the client.

Additional features of Next.js

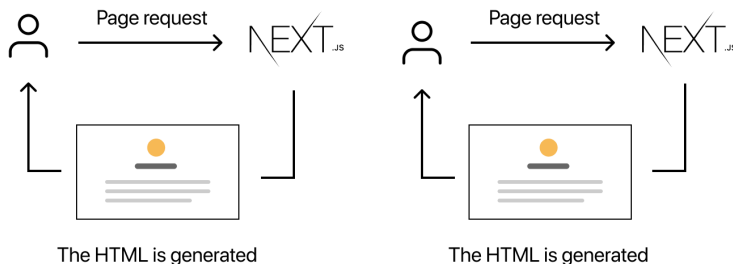
Although its main purpose serves to be the server-side rendering, Next.js also provides the user with many more additional features. These include:

- Hot Code Reloading (Next.js reloads the page when it detects any change saved to disk.)
- Automatic Code Splitting (Pages are rendered with just the libraries and JavaScript that they need, no more. Instead of generating one single JavaScript file containing all the app code, the app is broken up automatically by Next.js in several different resources.)
- Prefetching (The Link component, used to link together different pages, supports a prefetch prop which automatically prefetches page resources in the background.)

And many more additional features, which can be read about on the [Next.js site](#).

Server-side Rendering

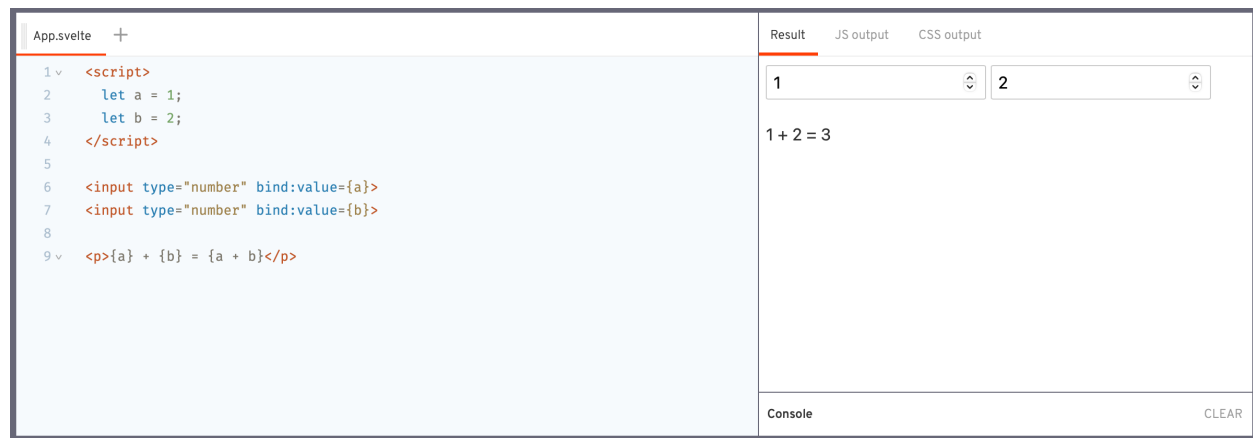
The HTML is generated on **each request**.



Important features from Svelte

"Svelte is a radical new approach to building user interfaces."

As written on the official Svelte site, Svelte is a completely new way to present your code. Instead of doing the work in the browser, like React applications, Svelte creates a new compile step. Instead of using techniques like virtual DOM diffing, Svelte writes code that surgically updates the DOM when the state of your app changes.



Svelte also uses a different style of writing code, which is much more simple to type, understand, and it takes up less space.

Additional features of Svelte

Svelte is very vast and complex, with an absurd amount of features, and the small ones cannot be left out. Below are a list of some important and interesting features that can be executed using Svelte:

- **State** (you update local component state with an assignment operator)
- **Bindings** (instead of React's system of responding to input events ourselves, Svelte has two tightly-coupled but physically separate chunks of code)
- **Top-level elements** (In Svelte, a component can have as many top-level elements as you like)

More features of Svelte can be explored on the [official Svelte website](#).

Why should you use Svelte or Next.js?

After reviewing the features of both Next.js and Svelte, it's easy to come to the conclusion that they both simply simplify the process of creating a web application, and improve the user experience on both ends of the web-app.

Sources:

Copes, Flavio. "The Next.js Handbook." *FreeCodeCamp.org*, FreeCodeCamp.org, 19 Nov. 2019, <https://www.freecodecamp.org/news/the-next-js-handbook/>.

"What Is Svelte?" *Educative*, <https://www.educative.io/edpresso/what-is-svelte>.

2019, Rich Harris Sat Apr 20. "Write Less Code." *Svelte*, <https://svelte.dev/blog/write-less-code>.

"Next.js." *Basic Features: Pages*, <https://nextjs.org/docs/basic-features/pages>.