

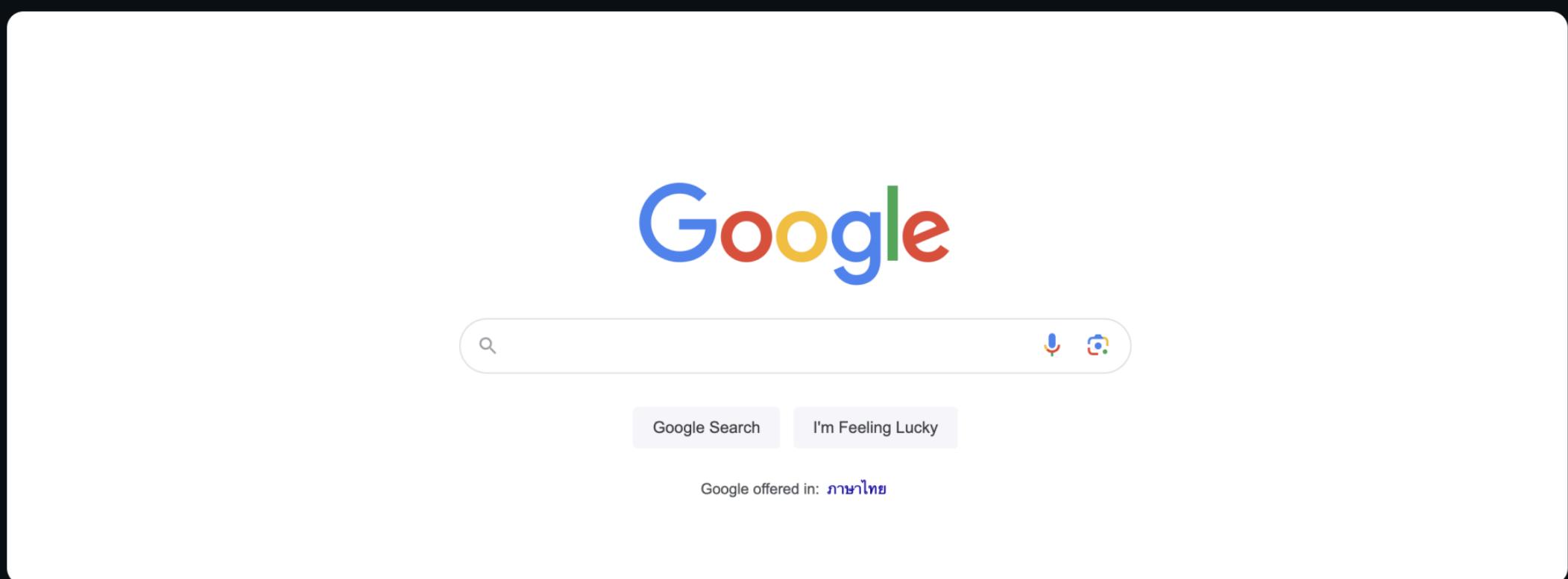
Search Engine Optimization: Next.js vs React



What is SEO?

SEO stands for **Search Engine Optimization**. Have you ever heard a business owner excitedly exclaim that they are '**on the first page of Google**'?

This means that they likely have great SEO practices and are **more likely to be found by their ideal customers!**



React.js

React.js is a JavaScript library that focuses on building UI's of web applications. It is designed for **client-side rendering**, meaning the application runs in the browser and updates the UI dynamically as users interact with it.

Next.js

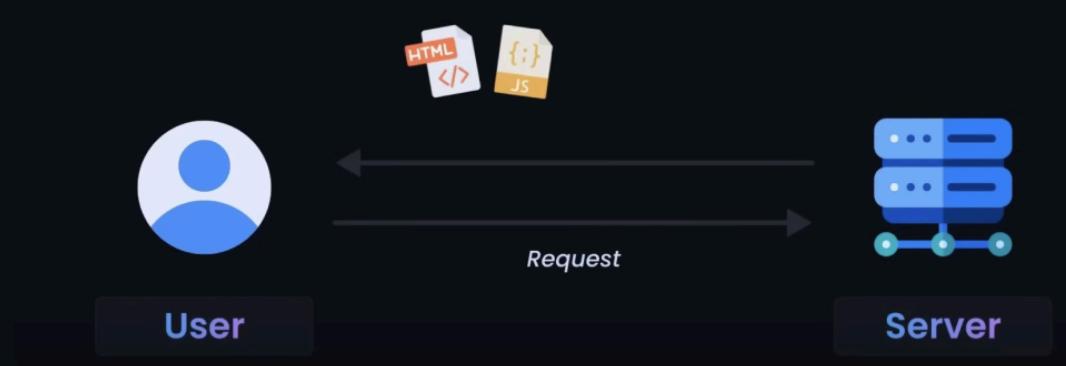
Next.js is a framework that enhances the capabilities of React. It offers flexibility with regards to rendering - your code can be rendered on the **client-side OR server-side** depending on your needs.

Client-Side Rendering

Where your code is rendered is important for SEO performance. So let's dive in and learn more about it.

Client side rendering (or browser rendering) happens **on the client's device or the browser**. When a user requests a web page, the server sends a basic HTML document and some JavaScript code.

The browser then **downloads and executes** the JavaScript code, which leads to the rendering of components and finally the **display of the website**.



Server-Side Rendering

On the other hand, server-side rendering involves rendering the web page **on the server before transmitting it to the client's device.**

When a user requests a page, the server processes the request and renders the components on the server side. The server then sends back the **fully rendered HTML** to the client's browser, enabling immediate display.

Rendering a website on the **server-side** is what makes a big difference when it comes to SEO!



Next.js vs React.js

React is designed for **client-side rendering**. This isn't optimal as search engine crawlers face difficulties when indexing pages that are dynamically rendered on the client side.

Next.js however, offers the option of **server-side rendering**. It sends pre-rendered code directly to the client which **improves SEO performance**.

Server-side rendering allows search engines to better index and understand the content of a webpage, improves loading times (and crawlability) and reduces content duplication.

So, is React 100% bad for SEO?

Not necessarily! While React can pose **SEO challenges due to aspects such as client-side rendering**, these challenges can be mitigated with proper **optimization techniques**.

Developers can leverage **SSR, SSG and other SEO best practices (such as image optimization and meaningful URLs)** to build React applications that are both user-friendly and **search engine friendly**.

Libraries like **React Helmet** can also help to manage meta tags and other **SEO-related aspects** of a React application.

What about React Server Components?

React Server Components (RSCs) offer **several advantages that can significantly enhance SEO.**

While it's **technically possible** to use React Server Components without Next.js, doing so requires staying on the cutting edge of React development and being prepared for the **potential complexities** of managing server-side rendering manually.

Next.js remains the **most straightforward path** to leveraging RSCs due to its official support and integration with React Canary releases.

Final Thoughts

By focusing on SEO, you can achieve **several benefits** for your website or business, including:

- increased organic traffic
- credibility and trustworthiness
- competitive advantage

For these reasons, **taking SEO into consideration** should ideally be an important part of the **software development process.** ✓

If you'd like to give sever-side rendering a try, head over to our **YouTube channel** to learn more about Next.js or sign up for our **Ultimate Next.js Course.** 🔥

Ultimate Next.js 14!

Join the course and become the
Next.js Developer you know you can be.

FOR DEVELOPERS WHO WANT TO STAND OUT AND BECOME IRREPLACEABLE

**Become a top 1% Next.js 14
developer in only one course**



jsmastery.pro/next14

