

Learn what *hydration* means, and how Gatsby makes use of React's hydration features to build blazing fast websites and applications.

What is hydration?

Hydration is the process of using client-side JavaScript to add application state and interactivity to server-rendered HTML. It's a feature of [React](#), one of the underlying tools that make the Gatsby framework. Gatsby uses hydration to transform the static HTML created at [build time](#) into a React application.

A typical React application relies on client-side rendering. Instead of parsing HTML to create the [DOM](#), client-side rendering uses JavaScript to create it. A minimal HTML document serves as the application container, and only contains links to the JavaScript and CSS necessary to render the application.

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width,user-scalable=yes" />
    <title>ExampleApp</title>
    <style>
      body {
        background: "images/loader.svg" center no-repeat;
      }
    </style>
  </head>
  <body>
    <!-- This empty element becomes the application's container -->
    <main id="root"></main>
    <script type="text/javascript" src="./js/main.0932cd15.js"></script>
  </body>
</html>
```

With client-side rendering, most actions trigger local DOM updates instead of network requests. Clicking a navigation link builds the requested page on the client instead of requesting it from the server. Because they make fewer network requests, applications rendered in the browser provide a blazing-fast user experience — after the initial download.

That's the drawback to client-side rendering: none of your site's content is visible or interactive until the client downloads JavaScript and builds the DOM. However, not all clients can construct a DOM. For example, client-side rendering can prevent search engines and social media crawlers from consuming and indexing your site's URLs. Browser users, on the other hand, may see a blank page or loading image while your JavaScript bundle downloads and executes.

[Server-side rendering](#) makes HTML available to the client *before* JavaScript loads. Your site visitors can see and read your content even if it is not fully interactive. Server rendering eliminates the blank page problem. Rendered HTML also makes it easier for search engines and social media crawlers to consume your site. Server-side rendering also has a drawback: every URL request requires another round trip to the server.

Hydration lets us take a hybrid approach.

Note: You'll sometimes see developers use *re-hydration* used instead of *hydration*. They're interchangeable.

Gatsby's build process uses [Node.js](#) and [ReactDOMServer](#) to create two different versions of your site. Each URL is available as both a static HTML page, and as a JavaScript component.

When a visitor requests their first URL from your site, the response contains static HTML along with linked JavaScript, CSS, and images. React then takes over and *hydrates* that HTML. React adds event listeners to the DOM created during HTML parsing, and turns your site into a full React application. Subsequent page requests are DOM updates managed by React.

Learn More

- [Understanding React Hydration](#) from the Gatsby docs
- [ReactDOM.hydrate\(\)](#) from the React API Reference
- [Rendering on the Web](#) from Google