

How has JavaScript evolved on the server side over the years?

Daniel Quinn

So, JavaScript in the past had to mainly respect the separation between the back end and the front end. And by that I mean, the process of content flow or even, like, the workflow of an application on the web begins in the back end when the back-end developer writes in a language like PHP or dotnet in order to render the application itself, so the markup and the content, and then it's served over to the browser. And at that point, it's been rendered.

So, there's nothing more we can do to the page except through manipulating it through JavaScript. And there were ways to communicate with the back end to get more information, but those ways were rudimentary because JavaScript was inconsistent between browsers.

Nowadays, though, the language has become so robust that JavaScript is able to bridge the gap between the back and the front end. It can not only communicate with the server directly in a way that's structured, but it can also bypass the need for the back end entirely. So, for example, in React and in Facebook, the browser is able to use JavaScript to go back and get the HTML, the CSS, and the content all at once and then render it into the page without refreshing the page.

So, essentially, the evolution of JavaScript has allowed us to obviate the need for a separation between the back and front end, and more and more, we're seeing the possibility that we're able to do everything in JavaScript, which can be a scary thing, but also a liberating thing.