

What is nodejs?

- JavaScript has existed since 1995 and has since taken over as the dominant language for web development. For much of its life, JavaScript was used mainly for client-side scripting inside `<script>` tags executing in web browsers. This limitation meant that developers were often working in many different languages and frameworks between the front-end (client-side) and backend (server-side) aspects of a web application.

Modules

Node has many built-in modules to aid in interactions with the command line, the computer file system, and the Internet. These include

- ⑩ HTTP and HTTPS for creating web servers.
- ⑩ File System, OS, and Path for interacting with the file system, operating system, and file/directory paths.

You can view the full [docs](#) to see more of Node's built-in features.

Why Node?

Per the [Node.js homepage](#), Node “uses an event-driven, non-blocking I/O model.” In practice, this means that Node is built well to handle asynchronous JavaScript code to perform many asynchronous activities such as reading and writing to the file system, handling connections to database servers, or handling requests as a web server.

To handle asynchronous code, Node uses a callback-based system. Node functions and methods that will implement some asynchronous activity take a callback function. This callback will be called whenever the asynchronous operation has resolved. By convention, the first argument of this callback is an error placeholder. If an error occurred in the asynchronous operation occurred (trying to read a non-existent file, for example), the error argument will be an Error object, but it will be null if no error occurs.