

# Introducing Node.js

In this chapter, you'll discover how to create JavaScript applications outside the browser thanks to a technology called Node.js.

## We'll cover the following ^

- A Bit of History
- A First Example

## A Bit of History #

To understand what [Node.js](#) (or Node for short) is, we have to travel back in time to the 2000's. As JavaScript was becoming increasingly important for improving the user experience on the web, web browser designers spent a considerable amount of resources on executing JS code as fast as possible. In particular, the Chrome JavaScript engine, codenamed V8, became open source in 2008 and was a huge step forward in general performance and optimization.



The core idea behind Node.js was simple yet visionary: since the V8 engine is so good at executing code, why not leverage its power to create efficient JavaScript applications outside the browser? And thus Node.js was born in 2009, originally written by Ryan Dahl. Its project quickly became very popular and Node is now one of the top technologies for building apps and creating APIs with JavaScript



Node also made it easier for developers to publish, share and reuse code. Today, hundreds of thousands of ready-to-use JavaScript libraries, called packages, are available and easy to integrate in any Node-based project (more on that later). This rich ecosystem is one of Node's greatest strengths.

## A First Example #

The rest of this chapter assumes a working Node environment. Refer to the appendix for setting one up.

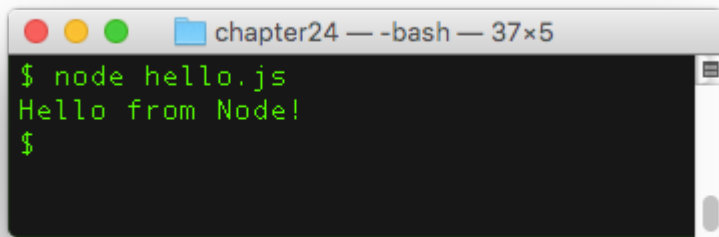
The simplest possible Node program is as follows.

```
console.log("Hello from Node!");
```



As you see, the `console.log()` command is also available in Node. Just like in a web browser, it outputs the value passed as parameter to the console. Assuming this code is saved into a file named `hello.js`, here's how to execute it through Node in a terminal.

```
node hello.js
```

A screenshot of a macOS terminal window. The title bar at the top shows three colored window control buttons (red, yellow, green) followed by a folder icon and the text "chapter24 — -bash — 37x5". The terminal has a black background with green text. It shows a prompt "\$" followed by the command "node hello.js", the output "Hello from Node!", and another prompt "\$".

```
chapter24 — -bash — 37x5
$ node hello.js
Hello from Node!
$
```

An in-depth study of the Node platform is out of this book's scope. Let's focus on two of its defining features: *modules* and *packages*.