Awesome work! You’ve learned a lot about Node.js including:

* Node.js is a JavaScript *runtime*, an environment that allows us to execute our JavaScript code by converting it into something a computer can understand.
* REPLs are processes that **r**ead, **e**valuate, **p**rint, and repeat (**l**oop), and Node.js comes with its own REPL we can access in our terminal with the node command.
* We run JavaScript programs with Node in the terminal by typing node followed by the file name (if we’re in the same directory) or the absolute path of the file.
* Code can be organized into separate files, modules, and combined through *requiring* them where needed using the require() function.
* In addition to *core modules*, modules included within the environment to efficiently perform common tasks, we can also create our own modules using module.exports and the require() function.
* We can access NPM, a registry of hundreds of thousands of packages of re-usable code from other developers, directly through our terminal.
* Node has an event-driven architecture.
* We can make our own instances of the EventEmitter class and we can subscribe to listen for named events with the .on() method and emit events with the .emit() method.
* Node uses an event loop which enables asynchronous actions to be handled in a non-blocking way by adding callback functions to a queue of tasks to be executed when the callstack is empty.
* In order to handle errors during asynchronous operations, provided callback functions are expected to have an error as their first parameter.
* Node allows for both *output*, data/feedback to a user provided by a computer, and *input* data/feedback to the computer provided by the user.
* The Node fs core module is an API for interacting with the **f**ile **s**ystem.
* *Streams* allow us to read or write data piece by piece instead of all at once.
* The Node http core module allows for easy creation of *web servers*, computer processes that listen for requests from clients and return responses.

Woah, that was a lot… And there’s even more to Node that we didn’t cover in this lesson, but don’t panic! Learning Node isn’t about memorizing every aspect of the environment. The best way to get comfortable with Node is just to practice making things in it. Your imagination is the limit! If you haven’t already, [download Node on your local machine](https://www.codecademy.com/articles/setting-up-node-locally). You can start by recreating some of the programs you built in this lesson— put your own spin on a guessing game, for example. If you’re eager to build web application back-ends, we recommend you start [learning the awesome Express.js](https://www.codecademy.com/learn/learn-express) web framework.

Great work! We’re excited to see what you build!