**REPORT** - [Node.js Tutorial for Beginners: Learn Node in 1 Hour](https://www.youtube.com/watch?v=TlB_eWDSMt4&amp=&t=1s)

Node is not a programming language, it is an open source, cross platform environment for executing JavaScript code outside of a web browser - it’s often used for backend services such as APIs. It is ideal for highly scalable, data intensive services, and sets itself apart because it is: great for prototyping, is superfast and scalable, it uses JavaScript making it easy to do full stack development, it’s cleaner and more consistent, and has the largest ecosystem of open source libraries available. Similar to a browser, Node is a runtime environment with JavaScript code that can execute the code, but different from browsers there are additional objects that offer more capabilities otherwise not available in a browser.

Node applications are asynchronous by default allowing them to work more efficiently. This makes Node ideal for building applications that are data intensive and include a lot of disc or network access. It should not be used for CPU-intensive applications that require a lot of calculations. It is important to incorporate modularity as well. Every file within a Node application is considered a module, and every Node application has at least one called the “main module.” Node also does not execute code directly, it always wraps it within a function.

There are a handful of modules built into Node. With these we can work with files, the operating system and others. Some of the most useful are file system, HTTP, OS, Path, Process, Query Strings and Stream. One of the core concepts of Node is events. An event is essentially a signal that something has happened in the application. Another powerful building block of Node is the HTTP module. With this we can easily create backend services for a client application. In the real world, however, and as code gets more complex we use a different framework for building a server.