**Node**.**js** is an open-source, cross-platform JavaScript run-time environment for executing JavaScript code server-side. ... **Node**.**js** enables JavaScript to be used for server-side scripting, and runs scripts server-side to produce dynamic web page content before the page is sent to the user's web browser.

Node.js, often called simply "Node" in conversation, is a development platform built on top of Google's V8 JavaScript virtual machine. While [JavaScript](http://searchsoa.techtarget.com/definition/JavaScript) engines (including V8) are traditionally run in [Web browsers](http://searchwindevelopment.techtarget.com/definition/browser) to form the client side of a [client/server application](http://searchnetworking.techtarget.com/definition/client-server), the Node.js [libraries](http://searchsqlserver.techtarget.com/definition/library) are focused on building server-side applications in JavaScript.

Node.js is intended to run on a dedicated [HTTP](http://searchwindevelopment.techtarget.com/definition/HTTP) server and to employ a single thread with one process at a time. Node.js applications are [events-based](http://searchsoa.techtarget.com/definition/event-driven-architecture) and run [asynchronously](http://searchnetworking.techtarget.com/definition/asynchronous). Code built on the Node platform does not follow the traditional model of receive, process, send, wait, receive. Instead, Node processes incoming requests in a constant event stack and sends small requests one after the other without waiting for responses

One of the major advantages of Node.js, according to its creator Ryan Dahl, is that it does not block input/output ([I/O](http://searchcio-midmarket.techtarget.com/definition/input-output)). Some developers are highly critical of Node.js and point out that if a single process requires a significant number of [CPU cycles](http://searchwinit.techtarget.com/definition/clock-cycle), the application will block and that the blocking can crash the application. Proponents of the Node.js model claim that CPU processing time is less of a concern because of the high number of small processes that Node code is based on.

Node.js is a server-side platform wrapped around the JavaScript language for building scalable, event-driven applications. This is confusing for even experienced programmers because the traditional JavaScript environment has always been client-side - in a user's browser or in an application that is talking to a server. JavaScript has not been considered when it comes to the server responding to client requests, but that is exactly what Node.js provides.

Node.js is not written in JavaScript (it is written in C++) but it uses the JavaScript language as an interpretive language for server-side request/response processing. In other words, Node.js runs stand-alone JavaScript programs. The advantage is that programmers can use their current, albeit client-side, programming knowledge and begin coding with Node.js much more easily

Node.js has several attributes that make it particularly attractive for network or over-the-Internet programming. The first has to do with all the overhead and packaging that existing technologies use to talk back and forth over the Internet

The second Node.js attribute that is attractive has to do with the Web programming event model. Most existing technologies are written to take "big gulps" of data for every request and response. In other words, a whole page of data might be sent to a server - even if there are only smallish changes. These technologies are optimized to use larger chunks of data with fewer events. Node.js does the opposite; it is designed to work with more interactivity - smaller chunks of data responding to many more events

Node.js also provides a rich library of various JavaScript modules which simplifies the development of web applications using Node.js to a great extent.

Node It’s the latest in a long line of “Are you cool enough to use me?” programming languages, APIs, and toolkits. In that sense, it lands squarely in the tradition of [Rails](http://rubyonrails.org/), and [Ajax](http://www.adaptivepath.com/ideas/ajax-new-approach-web-applications), and [Hadoop](http://radar.oreilly.com/2011/01/what-is-hadoop.html), and even to some degree [iPhone programming](http://oreilly.com/catalog/9780596806446) and [HTML5](http://radar.oreilly.com/2010/03/why-html5-is-worth-your-time.html). Go to a big technical conference, and you’ll almost certainly find a few talks on Node.js, although most will fly far over the head of the common mortal programmer