REFERENCES: https://opensourceforu.com/2016/01/an-introduction-to-node-js-the-server-side-javascript/

**An Introduction to Node.js, the Server Side JavaScript**

Node.js, often referred to as just Node, is a powerful tool that can run JavaScript applications on both the server side as well as the client side. Node.js can be used to write static file servers, Web application frameworks, messaging middleware, and servers for HTML5 multiplayer games. This article is just an introduction to Node.js.

Node.js is a server side JavaScript built on Google’s V8 JavaScript engine. It is an open source programming language that was developed by Ryan Dahl in 2009. It allows us to build scalable network applications, and is very fast when compared with other server side programming languages because it is written in C and the non-blocking I/O model. It is a software company specialising in high performance container native infrastructure. Node.js can run on Linux, Sun OS, Mac OS X and Windows platforms.

The steps below explain the difference between blocking code and non-blocking code models.  
In the blocking code model, the steps are:  
1. Read a file  
2. Process the file  
3. Print the result  
4. Perform the next function

The steps in the non-blocking code model are:  
1. Read a file  
1.1. When file reading is completed, process it  
1.1.1. When the processing is completed, print the result  
2. Perform the next function

**Internal architecture of Node.js**  
Node.js can be downloaded from [https://nodejs.org/](http://https/nodejs.org/%20)and is easy to install. Once installed in your system, you will see two of its main parts: the main node executable and the node package manager (NPM) executable. Node executable is simple, and you will pass the name of the main source file – for example, node sample.js.  
The node executable will interpret the source code and execute it. Once it finishes, it will exit back to the shell. The NPM (node package manager) will add new packages to the node executable.  
To install new modules, enter the command below:

|  |
| --- |
| npm install name-of-the-module |

**An HTTP server**  
If you are using PHP, a PHP file represents an HTML page. A Web server will accept a request and execute the code. But in Node.js, the main JavaScript file represents the entire Web server. It does not run inside a Web server.  
Now lets save the code below as main.js and run it.

|  |
| --- |
| var http = require(http);  var static = require(node-static);  var file = new static.Server();  http.createServer (function (request, response) {  file.serve (request, response);  response.writeHead(200, {Content-Type: text/plain});  response.end(Hello World!);  }).listen(1337, 127.0.0.1);  console.log(Server running at http://127.0.0.1:1337/); |

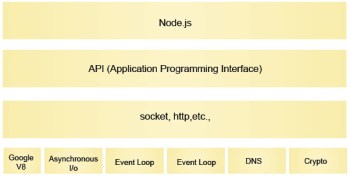
**How does the code given above work?**  
The global function require() makes a module available.  
HTTP is a built-in module, whereas node-static is an external module. So you need to install the external module with the NPM, by using the following command:

|  |
| --- |
| npm install node-static |

The keyword new is used to create an object.  
After installing node-static, you will have a new folder called node\_modules in your working directory. Now run the following command:

|  |
| --- |
| node main.js |

After running the node, you need to open the Web browser and type http://127.0.0.1:1337 into the address bar. You will see a simple Web page which says, Hello World!



**Creating your own module**  
It is very easy to create your own module in the Node.js environment.  
Step 1: Save the code below as custom\_module1.js.

|  |
| --- |
| exports.sayHello = function() {  console.log(Hello, Welcome to Node.js!);  } |

Step 2: Save the code below as main.js.

|  |
| --- |
| var hello = require(./custom\_module1.js);  hello.sayHello(); |

Step 3: Run the command node main.js.  
This is only an introductory article on Node.js. Interested readers are advised to explore further to find out about other features at [https://nodejs.org.](http://https/nodejs.org.)