COMPLETE NOTES ON

Node.Js

INDEX

NODE.JS

Node.js is a cross-platform runtime environment and library for running JavaScript applications outside the browser. It is used for creating Server-side and networking Web applications. It is open source and free to use.
Many of the basic modules of Node.js are written in JavaScript. Node.js is mostly use to run real-time server applications.
The definition given by its official documentation is as follows:
?Node.js is a platform built on chorme's JavaScript runtime for easily building fast and scalable network applications. Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient, perfect for data-intensive real-time applications that run across distributed devices.?
Node.js also provides a rich library of various JavaScript modules to simplify the development of Web applications.

Node.js = Runtime Environment + JavaScript Library

Features of Node.js

Following is a list of some important features of Node.js that makes it the first choice of software architects.

1. Extermely fast:
Node.js is built on Google Chrome's V8 JavaScript Engine, so its library is very fast in code execution.

2. I/O is Asynchronous and Event Driven:
All APIS of Node.js library are asynchronous ie, non-blocking. so a Node.js based server never Waits for an API to return data. The Server moves to the next API after calling it and a notifaction mechanism of Events of Node.js helps the server to get a response from the previous API call. It is also a reason that it is very fast.

3. Single threaded:
Node.js follows a single threaded model with event looping.

4. Highly scalable:
Node.js is highly scalable because event mechanism helps the server to respond in a non-blocking way.

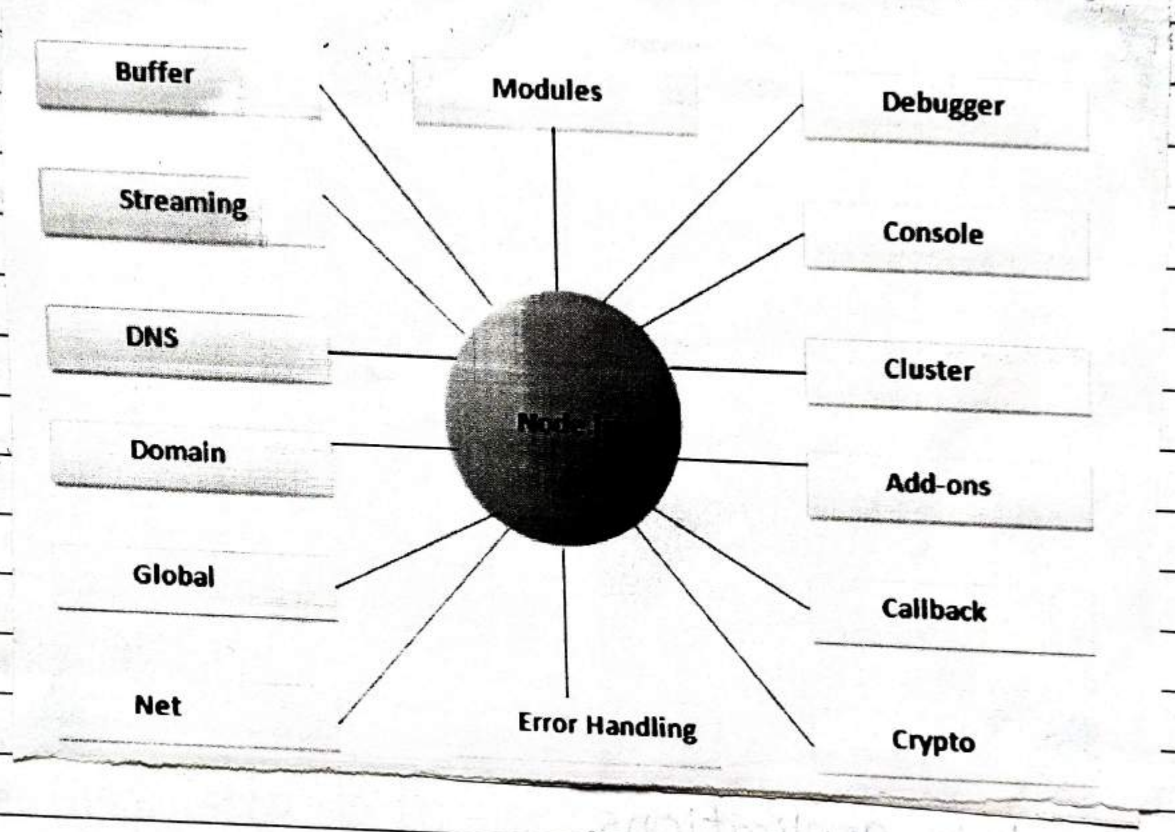
5. No buffering:
Node.js cuts down the overall processing time while uploading audio and video files. Node.js applcations never buffer any data. These applications simply output the data in chunks.

6. Open source:
Node.js has an open source community which has produced many excellent modules to add additional capabilities to Node.js applications.

7. License:
Node.js is released under the MIT license.

Different parts of Node.js

The following diagram specifies some important parts of Node.js:



Install Node.js on Windows

To install and setup an environment for Node.js, you need the following two softwares available on your Computer:

1. Text Editor
2. Node.js Binary installable

Text Editor:

The text editor is used to type your program. For example: Notepad is used in Windows, vim or vi can be used on Windows as well as Linux or UNIX. The name and version of the text editor can be different from operating system to operating system. The files created with text editor are called Source files and contain program source code. The source files for Node.js programs are typically named with the extension ".js".

The Node.js Runtime:

The source code written in source file is simply Javascript. It is interpreted and executed by the Node.js interpreter.

How to download Node.js:

You can download the latest version of Node.js installable archive file from https://nodejs.org/en/

Node.js First Example

There can be Console-based and Web-based node.js applications.

Node.js console-based Example

File: Console_example1.js

```
Console.log ("Hello JavaTpoint");
```

Open.log Node.js Command prompt and run the following code:

```
node console_example1.js
```

Node.js Web-based Example

A node.js web application contains the following three parts:

1. Import required modules:
The "require" directive is used to load a Node.jsmodule

2. Create server:
You have to establish a server which will listen to client's request similar to Apache HTTP Server.

3. Read request and return response:
Server created in the second step will read HTTP request made by client which can be a browser or console and return the response.

How to create node.js web applications.

Follow these steps:

1. Import required module:
The first step is to use ?require? directive to load http module and store returned HTTP instance into http variable. For example:

```
Var http = require("http");
```

2. Create server:
In the second step, you have to use created http instance and call http.createServer() method to create server instance and then bind it at port 8081 using listen method associated with server instance. Pass it a function with request and responce parameters and write the sample implementation to return "Hello World". For example:

```
http.createServer(function(request, response) {
   //Send the HTTP header
   // HTTP Status: 200 : ok
   // Content Type: text/plain
   response.writeHead(200,{'Content-type': 'text/plain'});
   //send the response body as "Hello World"
   response.end('Hello World\n');
}).listen(8081);
 // Console will print the message
 console.log('Server running at http://127.0.0.1:8081/');
```

3. Combine step 1 and step2 together in a file named "main.js".

Node.js Console

The Node.js Console module provides a Simple debugging console similar to JavaScript console mechanism provided by web browsers.
There are three console methods that are used to write and node.js stream:

1. console.log()
2. Console. error()
3. Console. warn()

Nodejs console.log()

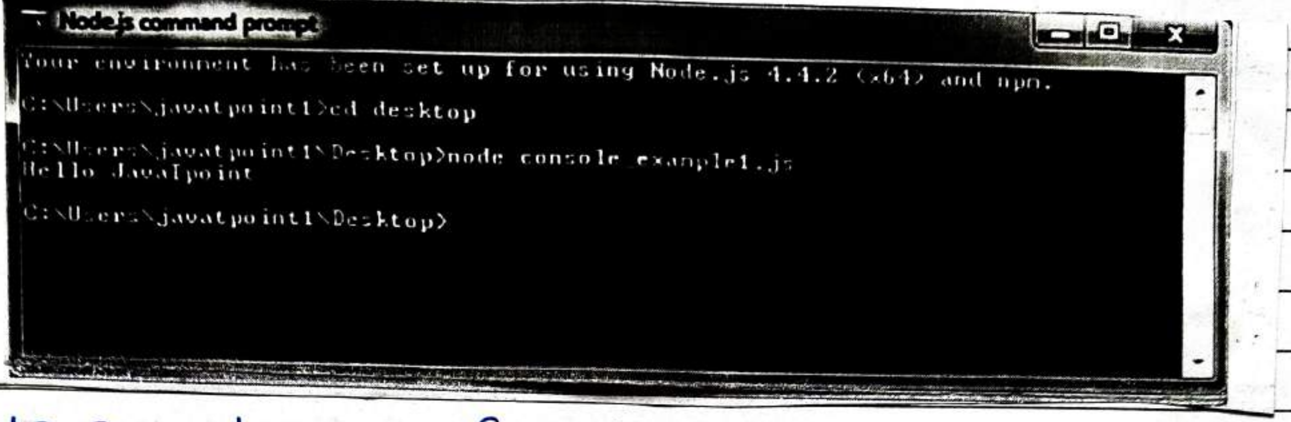
The Console.log() function is used to display Simple message on console.

```
Console.log('Hello JavaTpoint');
```

Open Node.js Command prompt and run the following

Code:

node console_example1.js



```
Your environment has been set up for using Node.js 4.4.2 (x64) and npm.

C:\Users\javatpoint1>cd desktop

C:\Users\javatpoint1\Desktop>node console_example1.js
Hello JavaTpoint

C:\Users\javatpoint1\Desktop>
```

We can also use format specifier in console.log() function.

File: Console_example2.js

```
Console.log('Hello %s','JavaTpoint');
```

Node.js console.error()

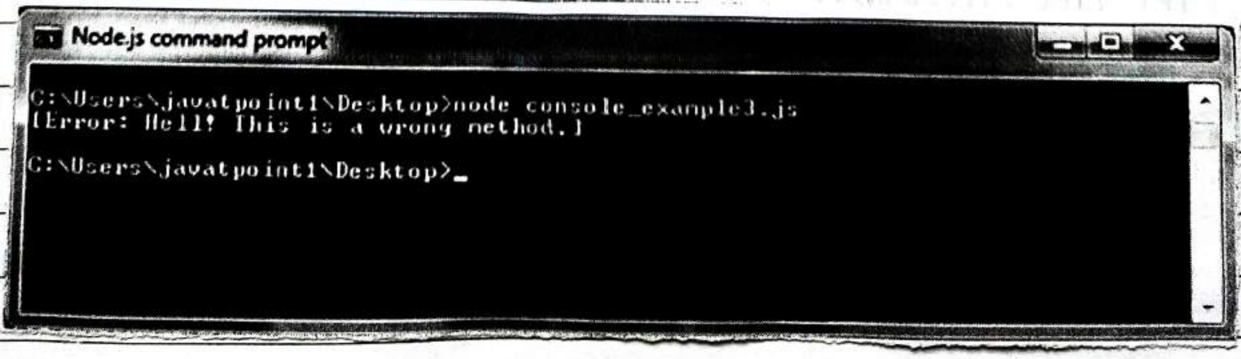
The console.error() function is used to render error message on console.

File: console_example3.js

```
Console.error(newError('Hell! This is a wrong method.'));
```

Open Node.js command prompt and run the following Code:

node console_example3.js



```
C:\Users\javatpoint1\Desktop>node console_example3.js
[Error: Hell! This is a wrong method.]

C:\Users\javatpoint1\Desktop>_
```