**Node.js Console**

Node.js comes with virtual environment called REPL (aka Node Shell). REPL Read Eval Print Loop. It is a quick and easy way to test simple Node.js/JavaScript code.

To launch the REPL (Node Shell), open command prompt (in window) to terminal and type node as show below.

|  |  |
| --- | --- |
| REPL Command | Description |
| .help | Display help of all the command |
| Tab keys | Display the list of all commands |
| UP/Down keys | See previous commands applied in REPL |
| .save fileName | Save current Node REPL session to a file |
| .load fileName | Load the specified file in the current node REPL session |
| Cntr + c | Exit from the REPL |
| Cntr + c (Twice) | Exit from the REPL |
| .break | Exit from multiple expression |
| .clear | Exit from multiple expression |

**Node.js Basic**

Node.js supports JavaScript. So, JavaScript syntax on Node.js is similar to the browser's JavaScript syntax.

**Primitive Data types**

String

Number

boolean

undefined

Null

RegExp

**Core Modules**

**file module**

**fileModule.js**

console.log(\_\_filename)

console.log(\_\_dirname)

**time.js**

function abc(){

console.log('It call only once')

}

var n=0;

setTimeout(abc,2000)

function xyz() {

console.log('It will call again and again '+n);

n++

if(n>=5) {

console.log('It stopped');

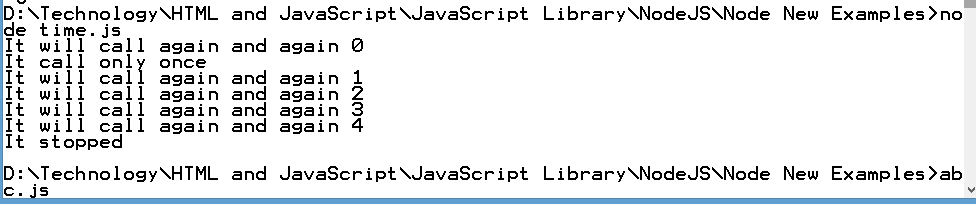
clearTimeout(holdValue)

}

}

var holdValue = setInterval(xyz,1000)

After running this program



**Global Objects**

Console: It is known as Console object. Used to print information on stdout and stderr.

**ConsoleDemo.js**

console.log('Log Details');

console.info('Info Details');

console.error('Err Details');

console.warn('Warn Details');

Process : Used to get information on current process. Provides multiple events related to process activities.

**processDemo.js**

console.log(process.pid)

console.log(process.env)

console.log(process.version)

console.log(process.arch)

console.log(process.platform)

process.stdout.write('Display msg through process object')

**OS Module**

Node.js os module provides a few basic operating - system related utility functions. This module can be imported using the following syntax

**osDemo.js**

var os = require("os")

console.log(os.hostname())

console.log(os.platform())

console.log(os.type())

console.log(os.release())

console.log(os.totalmem())

console.log(os.cpus())

**Path module**

Node.js path module is used to handling and transforming file paths. This module can be imported using the following syntax

var path = require("path")

var path = require("path")

var fileName="D:\Technology\HTML and JavaScript\JavaScript Library\NodeJS\Node New Examples\abc.js"

var file = path.basename(fileName);

var extension = path.extname(fileName);

var directories = ["dirA", "dirB", "dirC"];

var directory = directories.join(path.sep);

console.log(directory);

console.log(extension);

console.log(file);

**net module**

Node.js net module is used to create both server and clients. This module provides an asynchronous network wrapper and its can be improved using the following syntax

var net = require("net")