# Department of Computing

**CS-213: Advanced Programming**

**Class: BSCS 7AB**

# Lab 4: Introduction to Node.js

**Date: 26rd September, 2019**

**Time: 10:00-01:00pm & 02:00-05:00pm**

## **Name:** Amna Muqeem

## **Class:** BSCS 7B

## **CMS ID:** 216259

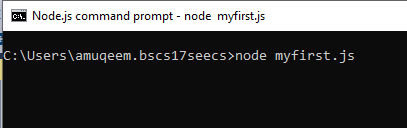
# Instructor: Dr. Sidra Sultana

**Lab Engineer: Ms. Ayesha Asif**

# 

**Lab Tasks Solution**

**Task 1:** Download Node.js from the official Node.js web site: [https://nodejs.org](https://nodejs.org/).



**Task 2:** Once you have downloaded and installed Node.js on your computer, let's try to display "Hello World" in a web browser.

Hint:

Create a Node.js file named "myfirst.js", and add the code.

Save the file on your computer: C:\Users\Your Name\myfirst.js

The file you have just created must be initiated by Node.js before any action can take place.

Start your node.js command line interface, write node myfirst.js and hit enter

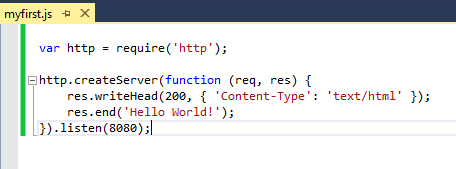
var http = require('http');

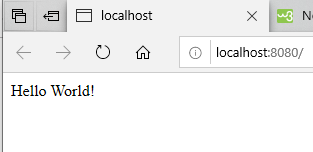
http.createServer(function (req, res) {

res.writeHead(200, { 'Content-Type': 'text/html' });

res.end('Hello World!');

}).listen(8080);



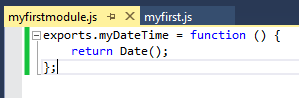


**Task 3:** Create a module that returns the current date and time. Save the code in a file called "myfirstmodule.js".

exports.myDateTime = function () {

return Date();

};



**Task 4:** Use the module "myfirstmodule" of date and time in a Node.js file.

var http = require('http');

var datemod = require('./myfirstmodule');

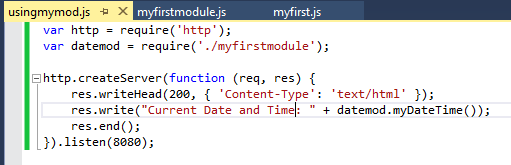
http.createServer(function (req, res) {

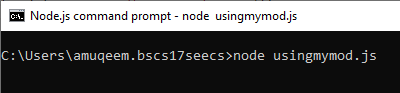
res.writeHead(200, { 'Content-Type': 'text/html' });

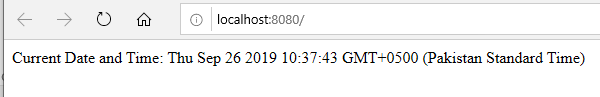
res.write("Current Date and Time: " + datemod.myDateTime());

res.end();

}).listen(8080);







**Node.js as a Web Server**

**Task 5:** The HTTP module can create an HTTP server that listens to server ports and gives a response back to the client. Use the createServer() method to create an HTTP server.

var http = require('http'); //using the pre-exiting http module in this file

http.createServer(function (req, res) { // using the createServer() method with the function having parameters request and response.

//more code..

**Task 6:** Add an HTTP Header

Hint: If the response from the HTTP server is supposed to be displayed as HTML, you should include an HTTP header with the correct content type like:

**res.writeHead(200, {'Content-Type': 'text/html'});**

The first argument of the res.writeHead() method is the status code, 200 means that all is OK, the second argument is an object containing the response headers.

var http = require('http');

http.createServer(function (req, res) {

res.writeHead(200, { 'Content-Type': 'text/html' }); // using the writeHead() method

res.end('My code'); // printing “My Code”

}).listen(8080); // the port on which we are performing these actions, it can be accessed through local host

**Node.js as a File Server**

The Node.js file system module allows you to work with the file system on your computer. To include the File System module, use the require() method:

var fs = require('fs');

Common use for the File System module:

* Read files
* Create files
* Update files
* Delete files
* Rename files

**Task 7:** Create a Node.js file that reads the HTML file, and return the content.

var http = require('http');

var fs = require('fs');

http.createServer(function (req, res) {

fs.readFile('myfile.html', function (err, data) {

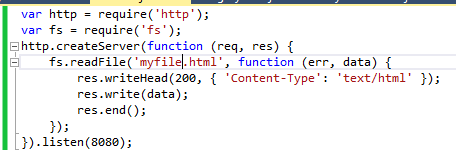
res.writeHead(200, { 'Content-Type': 'text/html' });

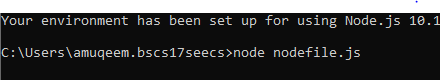
res.write(data);

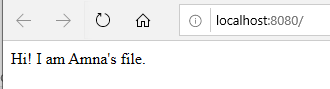
res.end();

});

}).listen(8080);







**Task 8:** Create a new file using

* appendFile() method

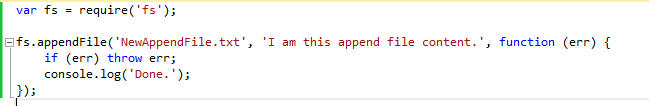
var fs = require('fs');

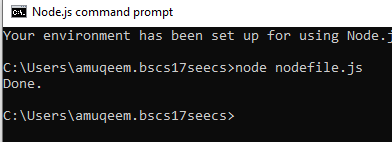
fs.appendFile('NewAppendFile.txt', 'I am this append file content.', function (err) {

if (err) throw err;

console.log('Done.');

});







* open() method

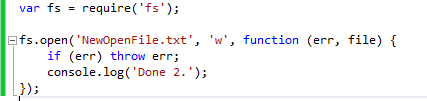
var fs = require('fs');

fs.open('NewOpenFile.txt', 'w', function (err, file) {

if (err) throw err;

console.log('Done 2.');

});







* writeFile() method

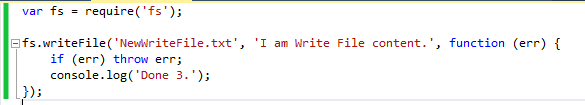
var fs = require('fs');

fs.writeFile('NewWriteFile.txt', 'I am Write File content.', function (err) {

if (err) throw err;

console.log('Done 3.');

});







**Task 9:** Append "This is my text." to the end of the file "mynewfile1.txt". – NewAppendFile.txt

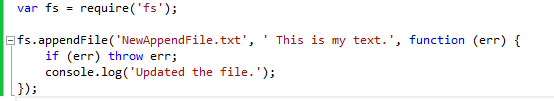
var fs = require('fs');

fs.appendFile('NewAppendFile.txt', ' This is my text.', function (err) {

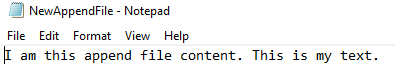
if (err) throw err;

console.log('Updated the file.');

});







**Task 10:** Replace the content of the file "mynewfile3.txt". – NewWriteFile.txt

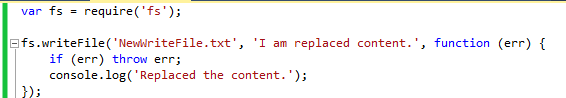
var fs = require('fs');

fs.writeFile('NewWriteFile.txt', 'I am replaced content.', function (err) {

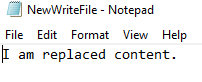
if (err) throw err;

console.log('Replaced the content.');

});







**Task11:** Delete "mynewfile2.txt". – NewOpenFile.txt

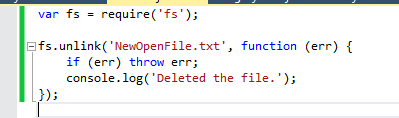
var fs = require('fs');

fs.unlink('NewOpenFile.txt', function (err) {

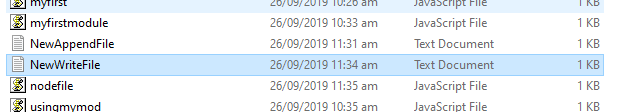
if (err) throw err;

console.log('Deleted the file.');

});







**Task 12:** Rename "mynewfile1.txt" to "myrenamedfile.txt".

var fs = require('fs');

fs.rename('NewAppendFile.txt', 'RenamedFile.txt', function (err) {

if (err) throw err;

console.log('Renamed the file.');

});

