NODE js

1.Save file into your folder-

2.Initiate "myfirst.js":

```
C:\Users\Your Name>node myfirst.js
```

```
Port 8080
```

3.What is a Module in Node.js?

Consider modules to be the same as JavaScript libraries.

A set of functions you want to include in your application.

http	To make Node.js act as an HTTP server
https	To make Node.js act as an HTTPS server.
path	To handle file paths
	url

Create Your Own Modules

You can create your own modules, and easily include them in your applications

The Built-in HTTP Module

Node.js has a built-in module called HTTP, which allows Node.js to transfer data over the Hyper Text Transfer Protocol (HTTP).

To include the HTTP module, use the require() method:

```
var http = require('http');
```

Node.js as a Web Server

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:

Example

```
var http = require('http');

//create a server object:
http.createServer(function (req, res) {
  res.write('Hello World!'); //write a response to the client
  res.end(); //end the response
}).listen(8080); //the server object listens on port 8080
```

The function passed into the http.createServer() method, will be executed when someone tries to access the computer on port 8080.

Save the code above in a file called "demo_http.js", and initiate the file:

Initiate demo_http.js:

```
C:\Users\Your Name>node demo_http.js
```

Add an HTTP Header

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:

Example

```
var http = require('http');
http.createServer(function (req, res) {
  res.writeHead(200, {'Content-Type': 'text/html'});
  res.write('Hello World!');
  res.end();
}).listen(8080);
```

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.

Read the Query String

The function passed into the http.createServer() has a req argument that represents the request from the client, as an object (http.IncomingMessage object).

This object has a property called "url" which holds the part of the url that comes after the domain name:

demo_http_url.js

```
var http = require('http');
http.createServer(function (req, res) {
  res.writeHead(200, {'Content-Type': 'text/html'});
  res.write(req.url);
  res.end();
}).listen(8080);
```

Save the code above in a file called "demo_http_url.js" and initiate the file:

Initiate demo_http_url.js:

```
C:\Users\Your Name>node demo_http_url.js
```

If you have followed the same steps on your computer, you should see two different results when opening these two addresses:

http://localhost:8080/summer

Will produce this result:

/summer

http://localhost:8080/winter

Will produce this result:

/winter

Split the Query String

There are built-in modules to easily split the query string into readable parts, such as the URL module.

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:

1]Read files:The fs.readFile() method is used to read files on your computer.

2]Create files:The File System module has methods for creating new files:

fs.open()

fs.writeFile()

3]Update files:The fs.appendFile() method appends the specified content at the end of the specified file:

4] Delete files:To delete a file with the File System module, use the fs.unlink() method.

The fs.unlink() method deletes the specified file:

5] Rename files:o rename a file with the File System module, use the fs.rename() method.

The Built-in URL Module

The URL module splits up a web address into readable parts.Parse an address with the url.parse() method

1.create two html file summer.html & winter.html

2.Create a Node.js file that opens the requested file and returns the content to the client. If anything goes wrong, throw a 404 error

Initiate myfirst.js

3.http://localhost:8080/summer.html

http://localhost:8080/winter.html

What is NPM?

NPM is a package manager for Node.js packages, or modules if you like.

What is a Package?

A package in Node.js contains all the files you need for a module.

Modules are JavaScript libraries you can include in your project.

```
C:\Users\Your Name>npm install upper-case
```

Create a Node.js file that will convert the output "Hello World!" into upper-case letters

Node.js Events

Node.js is perfect for event-driven applications.

Events in Node.js

Every action on a computer is an event. Like when a connection is made or a file is opened.

Objects in Node.js can fire events, like the readStream object fires events when opening and closing a file:

where you can create-, fire-, and listen for- your own events.

You can assign event handlers to your own events with the EventEmitter object

Node.js Upload Files

The Formidable Module

There is a very good module for working with file uploads, called "Formidable".

The Formidable module can be downloaded and installed using NPM:

```
C:\Users\Your Name>npm install formidable
```

Node.js Send an Email

The Nodemailer Module

The Nodemailer module makes it easy to send emails from your computer.

The Nodemailer module can be downloaded and installed using npm:

```
C:\Users\Your Name>npm install nodemailer
```