Unit 13.1 Highlights

http (core node module) - http module can create an HTTP server that listens to server ports and gives a response back to the client over the Hyper Text Transfer Protocol
http.createServer() - http module method to create an HTTP server
http.listen() - Starts the HTTP server listening for connections
response.write() - Method writes some data to the stream, aka response
response.end() - Method signals no more data will be written to the stream, with the option to add one final chunk to the response

Ex. 1

```
var http = require("http"); //load the http module (npm install is NOT needed)
var PORT = 8080

//callback directly in the method
http.createServer(function (req, res) {
  res.write('Hello World!'); //write a response to the client
  res.end(); //end the response
}).listen(PORT);
```

Ex. 2

```
var http = require("http"); //load the http module (npm install is NOT needed)
var PORT = 8080

function handler(req, res){
  res.write('Hello World!');
  res.end() //end the response
}

//callback function as a reference
http.createServer(handler).listen(PORT);
```

response.writeHead() - Method adds HTTP header values to the response, which are the name/value pairs displayed in the request and response messages of message headers for HTTP
request.url - Property containing the value of the URL path

Ex. 3

```
var http = require("http"); //load the http module (npm install is NOT needed)
var PORT = 8080

http.createServer(function (req, res) {
  res.writeHead(200, {'Content-Type': 'text/html'});
  res.write(req.url); //add the request path to the response
  res.end();
}).listen(PORT);
```

__dirname - Node global variable containing the directory of the file currently executing

Ex. 4

```
var http = require("http");
var fs = require("fs"); //load the fs module (npm install is NOT needed)
var PORT = 8080;

function handler(req,res){
  fs.readFile(__dirname + "/index.html", function(err, data) { //loads file from the
    res.writeHead(200, { "Content-Type": "text/html" });
    res.end(data);
  });
}

http.createServer(handler).listen(PORT);
```

req.method - Property containing the HTTP method of the request

Ex. 5

```
var http = require("http"); //load the http module (npm install is NOT needed)
var PORT = 8080;

function handler(req, res){
    res.writeHead(200, { "Content-Type": "text/html" });
    res.end(req.method); //adds the request method to the response
}

http.createServer(handler).listen(PORT);
```

.on() - Method which triggers when events occur in request, response, or server object

Ex. 6

```
var http = require("http");
var PORT = 8080;
var myServer = http.createServer(handler).listen(PORT)

function handler(req,res){
  res.end("YAY!")
}
myServer.on('request', (request, response) => {
  console.log('request is received');
});
```

(Source: https://nodemon.io/)

Nodemon - Nodemon is a utility that will monitor for any changes in your source and automatically restart your server

Ex. 7 (from cli, make sure to run a npm version 5.2+)

```
npx nodemon <filename>
```

Ex. 8 (from cli, using global npm package)

```
npm install -g nodemon
nodeman <filename>
```