Programming with NodeJS

Lesson 1: Setting up, JS basics and hello world

Febuary 7th, 2017

What is NodeJS? What programming languages is it like?

What is NodeJS? What programming languages is it like?

Previous series:

What is NodeJS? What programming languages is it like?

Previous series:
https://github.com/Casper-Oakley/python-lessons

What is NodeJS? What programming languages is it like?

Previous series:
https://github.com/Casper-Oakley/python-lessons
This series:

What is NodeJS? What programming languages is it like?

Previous series:
https://github.com/Casper-Oakley/python-lessons
This series:
https://github.com/Casper-Oakley/nodejs-lessons

What is NodeJS? What programming languages is it like?

Previous series:
https://github.com/Casper-Oakley/python-lessons
This series:
https://github.com/Casper-Oakley/nodejs-lessons

Figure: Image curtosy of HackSocNotts

Setting up your development environment

We will be using version: v6.9.5 (with npm version 3.10.10)

- Windows: **https://nodejs.org/en/download/**
- Mac OSX: **https://nodejs.org/en/download/**
- Linux: Either **sudo apt-get install nodejs** or **https://nodejs.org/en/download/**

If at any time you get stuck, just stick your hand up.

Setting up your development environment

You will also need a text editor. The choice of text editor is up to what you feel most comfortable using.

Setting up your development environment

You will also need a text editor. The choice of text editor is up to what you feel most comfortable using. Some options include:

Setting up your development environment

You will also need a text editor. The choice of text editor is up to what you feel most comfortable using. Some options include:

- notepad++

Setting up your development environment

You will also need a text editor. The choice of text editor is up to what you feel most comfortable using. Some options include:

- notepad++
- Vim

Setting up your development environment

You will also need a text editor. The choice of text editor is up to what you feel most comfortable using. Some options include:

- notepad++
- Vim
- GNU Emacs

Setting up your development environment

You will also need a text editor. The choice of text editor is up to what you feel most comfortable using. Some options include:

- notepad++
- Vim
- GNU Emacs
- Sublime

Setting up your development environment

You will also need a text editor. The choice of text editor is up to what you feel most comfortable using. Some options include:

- notepad++
- Vim
- GNU Emacs
- Sublime
- atom

Setting up your development environment

You will also need a text editor. The choice of text editor is up to what you feel most comfortable using. Some options include:

- notepad++
- Vim
- GNU Emacs
- Sublime
- atom - Fun fact, this was actually written in part in NodeJS!

JS basics

Javascript types

As with most common languages, the javascript engine contains primitive types:

Javascript types

As with most common languages, the javascript engine contains primitive types:

- Number

Javascript types

As with most common languages, the javascript engine contains primitive types:

- Number
- String

Javascript types

As with most common languages, the javascript engine contains primitive types:

- Number
- String
- Boolean

Javascript types

As with most common languages, the javascript engine contains primitive types:

- Number
- String
- Boolean
- Undefined

Javascript types

As with most common languages, the javascript engine contains primitive types:

- Number
- String
- Boolean
- Undefined
- Null

Javascript types

As with most common languages, the javascript engine contains primitive types:

- Number
- String
- Boolean
- Undefined
- Null

Javascript also has implicit objects, which contain all other types

Javascript types

Other types:

Javascript types

Other types:

- Arrays

Javascript types

Other types:

- Arrays
- JSON

Javascript types

Other types:

- Arrays
- JSON
- Buffers

Javascript types

Other types:

- Arrays
- JSON
- Buffers
- Streams

Javascript types

Other types:

- Arrays
- JSON
- Buffers
- Streams
- Higher order functions

Javascript syntax

Javascript syntax is C-like, with a soft type system.

Javascript syntax

Javascript syntax is C-like, with a soft type system.
Variables are initialised with *var*, though if not instantiated via *var* will be assumed to be global.

Javascript syntax

Javascript syntax is C-like, with a soft type system.

Variables are initialised with *var*, though if not instantiated via *var* will be assumed to be global.

Lines are not required to be delimited with a , though most standards strongly recommend it (and bugs can creep in without it).

Javascript syntax

Javascript syntax is C-like, with a soft type system.

Variables are initialised with *var*, though if not instantiated via *var* will be assumed to be global.

Lines are not required to be delimited with a , though most standards strongly recommend it (and bugs can creep in without it).

As with C, javascript contains for loops, while loops, conditional statements (both single expression ternary and expanded) and switches.

JavaScript Object Notation (JSON)

JavaScript Object Notation (JSON) is a standard originally built for javascript which provides simple, logical packaging of data into a data structure, which can in turn be converted into a string and back again.

JavaScript Object Notation (JSON)

JavaScript Object Notation (JSON) is a standard originally built for javascript which provides simple, logical packaging of data into a data structure, which can in turn be converted into a string and back again.

Elements in JSON are key-value pairs, where the key must be a string and the value can be any of the primitive types.

JavaScript Object Notation (JSON)

JavaScript Object Notation (JSON) is a standard originally built for javascript which provides simple, logical packaging of data into a data structure, which can in turn be converted into a string and back again.

Elements in JSON are key-value pairs, where the key must be a string and the value can be any of the primitive types.

JSONs are defined through curly braces {}.

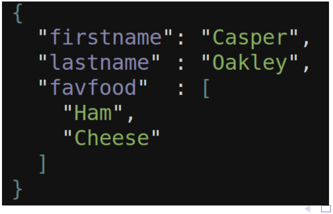
JavaScript Object Notation (JSON)

JavaScript Object Notation (JSON) is a standard originally built for javascript which provides simple, logical packaging of data into a data structure, which can in turn be converted into a string and back again.

Elements in JSON are key-value pairs, where the key must be a string and the value can be any of the primitive types.

JSONs are defined through curly braces {}.

For example:



```
{
  "firstname": "Casper",
  "lastname" : "Oakley",
  "favfood"  : [
    "Ham",
    "Cheese"
  ]
}
```

What is a web server?

What is a web server?

A web server is

What is a web server?

A web server is something which **serves** you things.

What is a web server?

A web server is something which **serves** you things.

When you go on a website, you, the client, ask a server for something.

What is a web server?

A web server is something which **serves** you things.
When you go on a website, you, the client, ask a server for something.This file could be an html document, an image or something completely different.

HTTP requests

HTTP requests

Specifically, you send the server an HTTP request. What is HTTP?

HTTP requests

Specifically, you send the server an HTTP request. What is HTTP?
HTTP - HyperTextTransferProtocol

HTTP requests

Specifically, you send the server an HTTP request. What is HTTP? HTTP - HyperTextTransferProtocol - The protocol which the world wide web relies on

HTTP requests

Specifically, you send the server an HTTP request. What is HTTP?

HTTP - HyperTextTransferProtocol - The protocol which the world wide web relies on

There are many types of HTTP request. The most common two used are:

- GET

HTTP requests

Specifically, you send the server an HTTP request. What is HTTP?
HTTP - HyperTextTransferProtocol - The protocol which the world wide web relies on
There are many types of HTTP request. The most common two used are:

GET	gets you data from a location/service

HTTP requests

Specifically, you send the server an HTTP request. What is HTTP? HTTP - HyperTextTransferProtocol - The protocol which the world wide web relies on

There are many types of HTTP request. The most common two used are:

GET	gets you data from a location/service
POST	

HTTP requests

Specifically, you send the server an HTTP request. What is HTTP?

HTTP - HyperTextTransferProtocol - The protocol which the world wide web relies on

There are many types of HTTP request. The most common two used are:

GET	gets you data from a location/service
POST	sends your data to a location/service

A little more on HTTP

A little more on HTTP

Status code responses

A little more on HTTP

Status code responses

- 200

A little more on HTTP

Status code responses

- 200
- 304

A little more on HTTP

Status code responses

- 200
- 304
- 400

A little more on HTTP

Status code responses

- 200
- 304
- 400
- 401

A little more on HTTP

Status code responses

- 200
- 304
- 400
- 401
- 403

A little more on HTTP

Status code responses

- 200
- 304
- 400
- 401
- 403
- 404

A little more on HTTP

Status code responses

- 200
- 304
- 400
- 401
- 403
- 404
- 500

The basics of NodeJS

NodeJS is an extremely powerful language developed on top of the V8 javascript engine, written by google.

The basics of NodeJS

NodeJS is an extremely powerful language developed on top of the V8 javascript engine, written by google. It operates on a single thread, using an event loop to manage concurrency.

The basics of NodeJS

NodeJS is an extremely powerful language developed on top of the V8 javascript engine, written by google. It operates on a single thread, using an event loop to manage concurrency. It has been found to be particularly strong in certain areas:

The basics of NodeJS

NodeJS is an extremely powerful language developed on top of the V8 javascript engine, written by google. It operates on a single thread, using an event loop to manage concurrency. It has been found to be particularly strong in certain areas:

- I/O bound applications

The basics of NodeJS

NodeJS is an extremely powerful language developed on top of the V8 javascript engine, written by google. It operates on a single thread, using an event loop to manage concurrency. It has been found to be particularly strong in certain areas:

- I/O bound applications
- Data streaming applications

The basics of NodeJS

NodeJS is an extremely powerful language developed on top of the V8 javascript engine, written by google. It operates on a single thread, using an event loop to manage concurrency. It has been found to be particularly strong in certain areas:

- I/O bound applications
- Data streaming applications
- JSON APIs

The basics of NodeJS

NodeJS is an extremely powerful language developed on top of the V8 javascript engine, written by google. It operates on a single thread, using an event loop to manage concurrency. It has been found to be particularly strong in certain areas:

- I/O bound applications
- Data streaming applications
- JSON APIs
- As the backend in a MEAN stack web app

The basics of NodeJS

NodeJS is an extremely powerful language developed on top of the V8 javascript engine, written by google. It operates on a single thread, using an event loop to manage concurrency. It has been found to be particularly strong in certain areas:

- I/O bound applications
- Data streaming applications
- JSON APIs
- As the backend in a MEAN stack web app - This is the one we will focus on

Running NodeJS code

Two ways to run your NodeJS code:

- Through the command line interpreter (the REPL)
- By putting your code into a file and executing the file

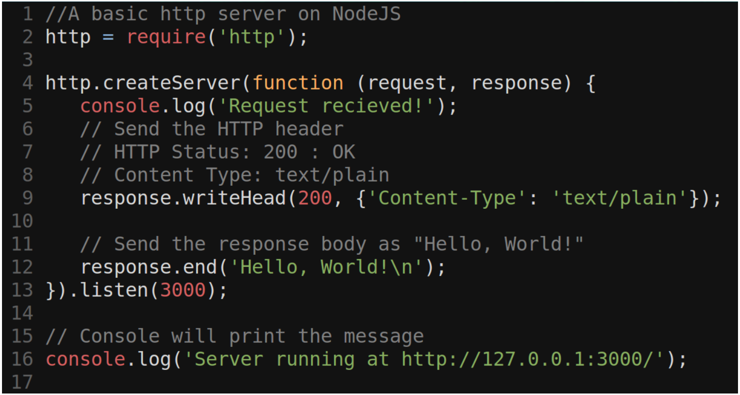
Running NodeJS code

Two ways to run your NodeJS code:

- Through the command line interpreter (the REPL)
- By putting your code into a file and executing the file

demo

A hello world server

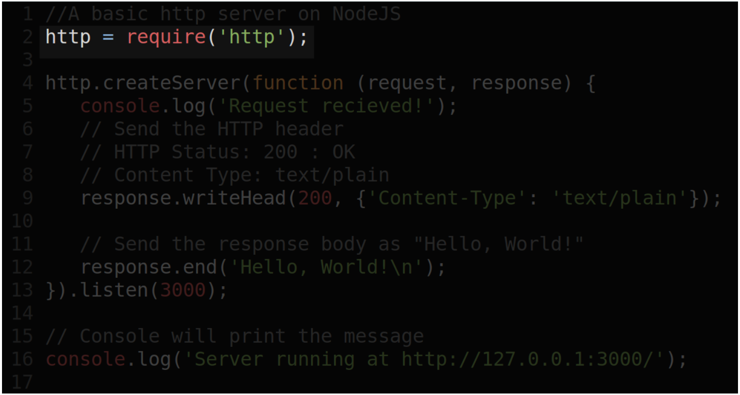


```
//A basic http server on NodeJS
http = require('http');

http.createServer(function (request, response) {
   console.log('Request recieved!');
   // 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(3000);

// Console will print the message
console.log('Server running at http://127.0.0.1:3000/');

```

A hello world server

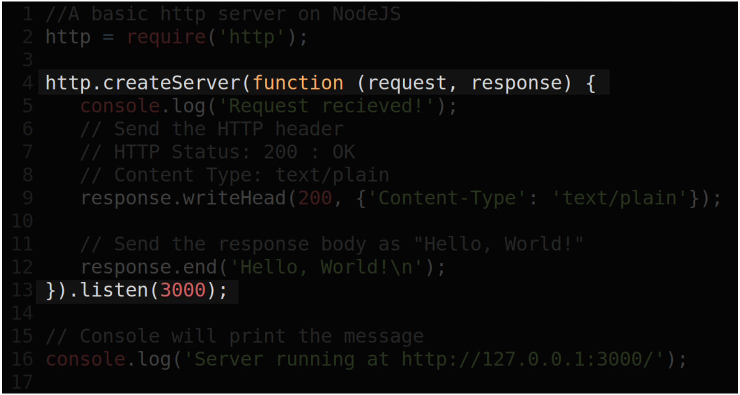
Our basic hello world server consists of three primary parts

A hello world server



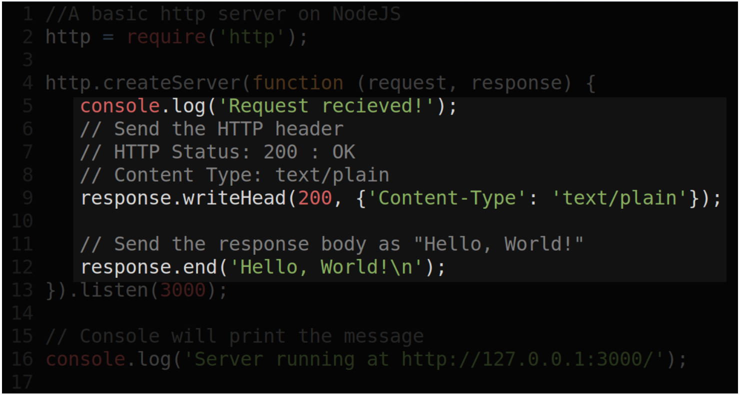
```
//A basic http server on NodeJS
http = require('http');

http.createServer(function (request, response) {
   console.log('Request recieved!');
   // 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(3000);

// Console will print the message
console.log('Server running at http://127.0.0.1:3000/');
```

A hello world server



```
//A basic http server on NodeJS
http = require('http');

http.createServer(function (request, response) {
   console.log('Request recieved!');
   // 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(3000);

// Console will print the message
console.log('Server running at http://127.0.0.1:3000/');

```

A hello world server



```
//A basic http server on NodeJS
http = require('http');

http.createServer(function (request, response) {
    console.log('Request recieved!');
    // 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(3000);

// Console will print the message
console.log('Server running at http://127.0.0.1:3000/');
```

NPM: node package manager

NodeJS's strengths lie mainly in it's ability to route information from places to other places, not to do heavy computation/store data/render stuff.

NPM: node package manager

NodeJS's strengths lie mainly in it's ability to route information from places to other places, not to do heavy computation/store data/render stuff. The Node Package Manager (NPM) sources all kinds of packages for NodeJS, each of which can be installed in one simple command!

NPM: node package manager

NodeJS's strengths lie mainly in it's ability to route information from places to other places, not to do heavy computation/store data/render stuff. The Node Package Manager (NPM) sources all kinds of packages for NodeJS, each of which can be installed in one simple command! *https://www.npmjs.com/*

Emoji demo

That's all for tonight!

To summarise:

- We have installed NodeJS and NPM
- We have learnt about JS syntax
- We have learnt about NodeJS and what it's useful for
- We have made a very simple http server

For next week

Source code plus lecture slides will be available online soon after the lesson.
If you are new to HackSocNotts, please join us on *http://hacksocnotts.slack.com*.
If you have any questions, feel free to ask now or over slack.