Programming with NodeJS

Lesson 2: Callbacks, buffers, streams, APIs and Express

Febuary 14th, 2017

Introduction
JS basics
Web Servers
Hello, World!
Summary

Setting up and basics

Summary of last week

Last week we:

Introduction
JS basics
Web Servers
Hello, World!
Summary

Setting up and basics

Summary of last week

Last week we:

- We have installed NodeJS and NPM

Introduction
JS basics
Web Servers
Hello, World!
Summary

Setting up and basics

Summary of last week

Last week we:

- We have installed NodeJS and NPM
- We have learnt about JS syntax

Introduction
JS basics
Web Servers
Hello, World!
Summary

Setting up and basics

Summary of last week

Last week we:

- We have installed NodeJS and NPM
- We have learnt about JS syntax
- We have learnt about NodeJS and what it's useful for

Introduction
JS basics
Web Servers
Hello, World!
Summary

Setting up and basics

Summary of last week

Last week we:

- 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

Callbacks? Huh?

NodeJS is a language developed around the idea of everything being asynchronous.

Callbacks? Huh?

NodeJS is a language developed around the idea of everything being asynchronous.
Asynchronous programming is programming where an operation is executed after the previous one has been terminated.

Callbacks? Huh?

NodeJS is a language developed around the idea of everything being asynchronous.

Asynchronous programming is programming where an operation is executed after the previous one has been terminated.

How does asynchronous programming differ from normal programming?

Callbacks? Huh?

NodeJS is a language developed around the idea of everything being asynchronous.

Asynchronous programming is programming where an operation is executed after the previous one has been terminated.

How does asynchronous programming differ from normal programming?

Consider this example:

Asynchronous programming example

Say you really want to have a chat with your mate Paul about the weather. You have two ways to get in touch with him. You can either:

Asynchronous programming example

Say you really want to have a chat with your mate Paul about the weather. You have two ways to get in touch with him. You can either:

- Phone him
- Send him a letter

Both have distinct advantages and disadvantages.

Both have distinct advantages and disadvantages.
If you phone him and ask him what the weather is like, you have to sit there and wait for him to look outside and reply.

Both have distinct advantages and disadvantages.

If you phone him and ask him what the weather is like, you have to sit there and wait for him to look outside and reply.

This is considered synchronous. You have a guarantee that he will reply as soon as possible, but you have to wait the entire time he's looking outside.

Both have distinct advantages and disadvantages.

If you phone him and ask him what the weather is like, you have to sit there and wait for him to look outside and reply.

This is considered synchronous. You have a guarantee that he will reply as soon as possible, but you have to wait the entire time he's looking outside.

Alternatively, if you send him a letter asking what the weather is like, you can do other things while you wait for the postman to come with his response. However, you have no idea how long until you will recieve that response is.

Both have distinct advantages and disadvantages.

If you phone him and ask him what the weather is like, you have to sit there and wait for him to look outside and reply.

This is considered synchronous. You have a guarantee that he will reply as soon as possible, but you have to wait the entire time he's looking outside.

Alternatively, if you send him a letter asking what the weather is like, you can do other things while you wait for the postman to come with his response. However, you have no idea how long until you will recieve that response is.

This is considered asynchronous.

NodeJS is asynchronous. The reason for this is just the same as with Paul and the weather. If instead of talking to just Paul, you want to talk to 100s of people about the weather they are all experiencing, an asynchronous approach will allow you to keep conversations open with all of these people at the same time.

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.