1.Slide : Start

This is a approx 20 minute talk on JavaScript. My brief is to keep it fairly non technical, so please don't switch off if you are not a developer. The aim of this talk is to introduce you to Javascript and convince you that its crucial for developers to master the JavaScript language.

Its also hopefully got enough detail to interest developers but there are no live code demos.

The talk shows that JavaScript can run with very high performance, that its spreading outside of web-browsers , that new JavaScript frameworks have arrived changing how teams develop web applications and that even bigger changes such as web components are on the near horizon.

Theres a speed war amongst the big companies - constantly striving to increase the speed that JavaScript runs on your PC (or Server or database).

(Questions at end)

2.Slide : Me

chris.brooksbank@rm.com

3.Slide : ImpressJS

This presentation doesnt use PowerPoint, it uses a free javascript framework called Impress.JS. The slide transitions are using CSS3. The slides are defined in static HTML. The presentation is just a static website thats hosted on my google drive and uses nothing but the standard webstack.

4.Slide : WhatIsJS

Javascript is a computer language which not only runs inside nearly every web browser on the planet but is rapidly spreading outside of web browsers.

If you want code that runs inside web-browsers then you will be writing JavaScript code. This talk concentrates on JavaScript running inside the browser because thats the most common usage. However you should be aware that its no longer constrained by the browser.

One hot, relatively new technology is called Node.Js. Node.JS lets you run JavaScript code on your web-server, not just inside peoples web browsers. So this is code that can access your database and construct dynamic web-pages to send back to the web browser.

JavaScript has also spread into databases. You might be aware that the world of databases is changing fast. Transactional databases such as MS SQL Server are not always the best option especially for BIG data, so we have the rise of NOSQL databases. One of the most important NOSQL databases is MongoDB. It uses JavaScript (JSON JavaScript Object Notation) to query

for data, or to insert, update or delete data.

So its very feasible to have a website that uses no other programming language than JavaScript. JavaScript in the database, on your web servers and of course in your web browsers. Im not proposing we throw out C sharp, JavaScript on the server is not as fast at some things, such as computationally heavy tasks, CSharp or other server side technology remains faster for CPU intensive processing work.

5.Slide : WhatIsAWebsite

Stepping back to basics. Websites are in some ways simple to understand. They are a HTTP server which respond to small simple text commands. Your web browser sends requests to a web server. When the HTTP server (web server) receives a request it sends data back. Thats it.

But HTTP servers have a very limited command set. e.g. GET, POST, PUT, DELETE.

One thing it sends back is HTML. HTML is a bunch of angle brackets describing something it wants your web browser to display.

Another thing it sends back is CSS. CSS3 is a text file describing how you want your web browser to display things. It might have a rule saying to display headers in Big Red letters. The latest version CSS3 allows for box shadows, animations and transitions plus a lot more.

Also it might send back some text which it describes as being JavaScript. You browser will interpret and run this program code inside itself.

Of course when you send a simple request to the web-server for a RM website, it may cause the web-server to run some Csharp code that we wrote, that code runs on the web-server, it may well issues request to a SQL database, either directly or via some web service, when it gets the data back it might process it in some way, before constructing a web-page (HTML and javascript) to send back to your web-browser.

But when you strip that away you have a simple HTTP Server responding to small text requests issued by your web browser understands some standardised technologies : HTML, CSS, JavaScript.

6.Slide : SampleHTTPGet

7.Slide: Fiddler2

you can watch those requests being sent and the responses come back by using the free tool called Fiddler2.

8.Slide : ChromeDevTools1

If you use Google Chrome as your web-browser it comes with an amazingly robust set of developer tools. You can view the requests and precise timings of responses,

, you can view the JavaScript thats sent back, run it one line at a time and watch what it does, you can profile and analyse its performance and memory usage.

9.Slide : DebugJavaScript

10.Slide : EmulateTouch

11.Slide : DebugJavaScriptOnTablet

12.Slide : ProfileJavaScript

13.Slide : JSFiddleAngular

14.Slide : GoodJS

When a developer is writing JavaScript there are some important differences from, say, CSharp. Its not worse, just different. Its a very dynamic language, you can change objects on the fly if you want, functions can even rewrite their own logic as they run, if you really think thats a good idea. You need to understand and embrace the differences.

The way scope works is completely different to CSharp (function scoped). And that yields some powerful ways of working. It allows global variables but you should not use those.

Functions are first class citizens in JavaScript. You can pass functions into other functions, for example you could have a generic sort routine where you pass in the comparison logic.

Functions can return functions, which in combination with the function scoping of JavaScript allows for a powerful tool called closures.Currying.

There are a lot of free tools to help you work with JavaScript. JSLint analyses your code as you type and highlights areas of concern in your code. There are tools to allow construction of and running of automated unit tests of your JavaScript. To compress your JavaScript to allow for lower bandwidth.

And as mentioned tools such as Chrome allow you to stepthru, debug and analyse your JavaScript performance and memory profile.

15.Slide : EvilJS

Global variables

Implicit type co-ercian

16.Slide : JSRealEngine

17.Slide : WhyBotherWIthJS

You don't need JavaScript code on your website. If you are happy that whenever the user presses a button, it sends a request to the WebServer, and the WebServer sends back a web-page which is rendered from scratch in the users browser, thats OK. Its probably a cheaper development option and we don't need JavaScript expertise to develop the website.

However the user will probably notice their page flickers. We will be using a lot of bandwidth to send back the entire web-page. Our web-servers and databases will be under increased load as everything the user does results in work on the server. So we need to spend more money on server equipment to allow the user t o experience the website.

The user might wonder why our website works this way when other websites he uses are more interactive, snappier and seem a lot more modern, to work a lot more like desktop applications rather than dead web pages. We are also wasting all that processing power built into the users web-browser and instead giving all the work to our web server.

18.Slide : HowToUseJavascript

JQuery used to be THE only JavaScript library you needed. A lot of JavaScript developers are actually JQuery developers. JQuery lets you write JavaScript that works on all web-browsers. It lets you write clean code that can precisely select pieces of your web-page and change them. It lets you send AJAX requests.

Your JavaScript can make a call to the web-server. Perhaps an area on the web-page is displaying some information on the currently selected school, like the headmasters name. When the user selects a different school the JavaScript can ask the web server for the information on the newly selected school. The web server finds out the information and sends it back, the same JavaScript gets this information and uses JQuery to update just that one small area of the screen.

So all that flows from the browser to the server is a small request to get some information on a school. The web server no longer has to construct and send back the entire web page. It just sends back the headmasters name. The javascript is notified when the information comes back, and JavaScript calls update that small area of the screen.

Much less data flows over the network, the server has much less to do, the user sees no page flicker, maybe the JavaScript animates the update of the new information smoothly.

So instead of having to POST back the entire page and for our server to send back the entire page we send a tiny piece of data. The format of that data will probably be JSON.

JSON is a universal data format which is displacing XML in many places. JSON = JavaScript Object Notation. Its a really simple notation, easy to read, easy to construct and JavaScript code understands it.

You used to use JavaScript for animations but you should now probably be looking at the new features in CSS3.

19.Slide : JQuery

Query is a massively well known and mature JavaScript library.

But it has a big problem. It ties the layout of your web-page and your business logic very tightly together. So much so that changing the layout of your web page may cause you to have to rewrite and retest your JavaScript.

20.Slide : AngularJS

Theres a better way, a way in which you can reduce the amount of JavaScript you have to write - massively and which also allows you to change the layout of your web page without rewriting your code.

There are new libraries and new frameworks where you embed tokens in your web-page, the web-page contains JSON describing your business data. The JavaScript framework then automatically constructs your web-page and two way binds it.

So maybe we develop a website using Microsoft Visual Studio, we create a Microsoft MVC website which gives us total control on the markup. No spagetti markup generated by Microsoft.

We send the browser a clean, standards compliant web-page which contains a JSON object and a template for displaying the web-page. The JavaScript framework renders the web-page using the template and JSON object. If the user edits a persons name, the framework automatically updates the JSON object, maybe automatically updates other parts of the page. If our JavaScript changes then name, then the text box automatically updates. When the user hits save we can send the JSON object back to our server, in the background, not the whole page. We didn't have to write our own JavaScript to do this, the framework gave it to us for free.

MVVM

Google Angular.Js is the framework that Im most interested in at moment along with Polymer, lots happening in this area, MVVM.

21.Slide : AngularControllers

22.Slide : AngularLook

23.Slide : BestPractices

24.Slide : TheFuture

There is a new technology coming called Web Components, Google are doing a lot of work on this. The aim is to make HTML a much better way of describing web applications. Instead of being restricted to tags pre-ordained by W3C such as H1, DIV, SPAN, TITLE you can define your own tags, for your own applications and have them understood by the browser and rendered as you want. for example a tag such as <RM-School DFES="12345">.

The Shadow-Dom is a technique browsers use to encapsulate these components, to allow you to encapsulate styling and behaviour without the general web-pages CSS messing with your component.

25.Slide : WebComponents

26.Slide : End