Commitment to JS

Frame the entire discussion around Microsoft committing to JS in a big way. Goal is to show MS is serious about this and educate on the various options which exist

# Introduction

Standard introduction bit – these are the topics I plan to cover. Broken into three stories, platform looks at our platform support for JS. IDEs looks at the improvement in VS & introduction to Code. Tools covers our extras we are shipping to make everyone’s life easier.

Less than 1min.

* Platform
	+ Windows 10
	+ Edge
	+ Project Westminster
* IDEs
	+ Visual Studio 2015
	+ Visual Studio Code
* Tools
	+ TypeScript
	+ WinJS
	+ Vorlon

# Windows 8

Brief recap of Windows 8 and what we did in Windows 8 for JS. Less than a min. Goal is to setup for compare to Windows 10. Key touch points:

* Store app platform
* Build once for Windows Phone 8.1 & Windows 8
* Build in JavaScript with CSS & HTML
* Rendering engine is IE 11

# Windows 10

* Universal app platform
* Build once for Windows 10 Mobile, Windows 10, Xbox & Hololens
* Build in JavaScript
* Rendering engine is Edge
* Free upgrade from Windows 7, 8 & 8.1

# IE

* Old browser with backwards compat
* Poor performance
* Constantly updates at a slow pace
* Plug-in model requires native code (& some C++)

# Edge

* New browser with no backwards compat
* Great Performance
* Constantly updates with fast performance
* New plug-in model uses JavaScript
* Aligned to Chrome & FireFox so easy to port over

# Project Westminster

* Part of our four bridge plan
* iOS, Android, Classic Windows Programs, Web
* <DEMO Creating a project Westminster app for JSinSA>

# Visual Studio 2015

* Community, Express, Professional, Enterprise
* Community & Express are free
* Express: has functionality limits and low usage limits
* Community: no functionality limits and some usage limits
* Major enhancements for JavaScript, HTML & CSS
* <DEMO VS Enhancements>

# Visual Studio Code

* Free IDE & Cross Platform
* MacOS and Linux
* BETA BETA BETA
* Current goal is a code first experience
* Build in JS, CSS, HTML & Chrome
* Amazing support for JavaScript
* <DEMO: VS Code on Linux>

# TypeScript

* JavaScript transpiler
* Allows you to use JavaScript 2015 & 2016 functionality today
* TypeScript gives extra info to IDEs
* All JavaScript is already TypeScript
* <DEMO: TypeScript in Sublime>

# WinJS

* Cross platform JavaScript control library
* Brings modern design controls to apps
* Useful with everyone going flat
* <DEMO – WinJS demo app>

# Vorlon

* Cross platform JavaScript library to allow remote debugging
* Think browser dev tools without the browser and over the internet
* <DEMO – Vorlon>

JSiNSA Demo Script

# Setup

* Win 10
* VS 2015 < start it up
* Sublime + typescript plugin < start it up
* Vorlon server running + website
* VS Code (on Linux VM) < start it up
* Open try.build**winjs**.com

# Project Westminster DEMO

StartPage="http://www.findfreewifi.co.za/">

 <uap:ApplicationContentUriRules>

 <uap:Rule Match="http://\*.findfreewifi.co.za/" Type="include" WindowsRuntimeAccess="all"/>

 </uap:ApplicationContentUriRules>

 <DeviceCapability Name="location"/>

# VS 2015 DEMO

New Angular + WinJS project

Right click > add grunt + bower

Show grunt + bower config files

Open packages.json

Add grunt-copy to it and hover over it to show the details

HTML:

 <ng-view/>

 {{Dog}}

 <horse></horse>

App.js > F12

Class snippet – method: awesome(){}

JSDoc example /\*\* @function \*/

Navbar

Commit to repro

# VS Code DEMO

Pull from repro

Bower.json

“private”: true 🡪 “private”:”true”

Open html – show ng-app

To app.js code everything is there too

Show JSDoc tooltip working

# TypeScript DEMO

Rename app.js to app.ts

Open in sublime

Show intellisense dog = “” vs dog = 1 demo

Ctrl+B > show JS

# WINJS DEMO

Open website – show semantic zoom hub, angular

# Vorlon DEMO

VORLON DEMO