MTA HTML5 APP DEVELOPMENT FUNDAMENTALS

11/03/2019

K-275-01

Your fastest way to learn. Guaranteed.

FIREBRAND

INTRODUCTION

- Welcome

- You are going to ENJOY THIS COURSE and LEARN and CLARIFY anything you didn't know about HTML, CSS and JAVASCRIPT to a beginner/medium level

- Prerequisites
  - Basic knowledge already of CSS, HTML and Javascript
  - That's it!

CRITICAL READING

- FIREBRAND FAST TRACK

- If you want the FAST TRACK INFO FOR THIS COURSE THEN CHECK OUT THE FOLLOWING LINKS

CRITICAL READING

- AJAX http://www.w3schools.com/xml/ajax_intro.asp
- CSS Grid Model http://www.w3schools.com/w3css/w3css_grid.asp
- CSS Box Model http://www.w3schools.com/css/css_boxmodel.asp
- CSS Flex Model http://www.w3schools.com/css/css3_flexbox.asp
- HTML File API http://www.w3schools.com/jsref/dom_obj_fileupload.asp
- JSON http://www.w3schools.com/js/js_json_intro.asp
- Javascript Web Workers http://www.w3schools.com/html/html5_webworkers.asp
- Javascript Web Sockets https://www.html5rocks.com/en/tutorials/websockets/basics/
- Geolocation API http://www.w3schools.com/html/html5_geolocation.asp
- Canvas http://www.w3schools.com/html/html5_canvas.asp
- SVG http://www.w3schools.com/html/html5_svg.asp
- CSS Transformations http://www.w3schools.com/css/css3_2dtransforms.asp
- W3C Official HTML Specification https://www.w3.org/TR/html5/
- List of HTML tags http://www.w3schools.com/tags/default.asp
- Mozilla HTML Reference https://developer.mozilla.org/en-US/docs/Web/HTML/Element

RESOURCES USED IN THIS COURSE

- Resources used in the course

- Code Editor
  - Visual Studio for Web
  - Notepad++
  - Sublime Text

- Firebrand Courseware

- Official Microsoft PDF courseware

OFFICIAL EXAM LINK

- Official Microsoft Exam Link

  - https://www.microsoft.com/en-us/learning/exam-98-375.aspx

1 - LIFECYCLE

- We will be looking at the following topics:

1. Intro
2. Building an application: looking at the Software Development Lifecycle and other factors involved with building a modern application from scratch
   - Containers
   - Package Files
   - Manifest Files
     - Capabilities of an App
     - Publisher of an App
     - Prerequisites
     - Dependencies

1 - LIFECYCLE

- Security eg. Permissions
- Authentication
- Licensing and EULA
- IDE Platforms
- SDK Software Dev Kit
- Debugging
- Testing
- Validating
- Certifying
- Publishing

2 – SOFTWARE DEVELOPMENT LIFECYCLE

- Building an application in Windows

- Building In Windows
  - Windows 8: Mobile operating system was called Windows RT or Windows RunTime
  - Windows 10: Platform is now called UWP, Universal Windows Platform

- Concept of an application

- Main components of an application

2 – SOFTWARE DEVELOPMENT LIFECYCLE

- Containers for your application
  - Application Container
  - Application Package

- Package File
  - All of the files inside the application, zipped up and made available for distribution
  - Examples include:
    - Android.APK
    - IOS.APP
    - Windows.EXE.MSI

2 – SOFTWARE DEVELOPMENT LIFECYCLE

Application Manifest File

- Contains instructions about the capability of the application
  - Eg. Windows Store App – File is called Package.appxmanifest
- Basic template example: https://msdn.microsoft.com/en-us/library/windows/apps/br211475.aspx

```
<?xml version="1.0" encoding="utf-8"?>
<Package xmlns="http://schemas.microsoft.com/appx/2010/manifest">
  <Identity Name=""
            Version=""
            Publisher="" />
  <Properties>
    <DisplayName></DisplayName>
    <PublisherDisplayName></PublisherDisplayName>
    <Logo></Logo>
  </Properties>
  <Prerequisites>
    <OSMinVersion></OSMinVersion>
    <OSMaxVersionTested></OSMaxVersionTested>
  </Prerequisites>
  <Resources>
    <Resource Language="" />
  </Resources>
  <Applications>
    <Application Id="" StartPage="">
      <VisualElements DisplayName="" Description=""
          Logo="" SmallLogo=""
          ForegroundText="" BackgroundColor="">
        <SplashScreen Image="" />
      </VisualElements>
    </Application>
  </Applications>
</Package>
```

2 – SOFTWARE DEVELOPMENT LIFECYCLE

Application Manifest File

- Main purpose is to:
  - Trigger the screen which says “This application requests access to these features….Do you wish to grant the application permission to use these features”
- Contains
  - IDENTITY OF THE APPLICATION (WHAT IT IS CALLED)
    - Identify Name=“MyCompany.MySuite.MyApp” Version=“1.0.0.0”
  - PUBLISHER OF THE APP
    - Publisher=“CN=MyCompany, O=MyCompany, L=MyCity, S=MyState, C=MyCountry”>

2 – SOFTWARE DEVELOPMENT LIFECYCLE

- Application Manifest File

- Contains…
  - DISPLAY NAME
    - <DisplayName>MyApp</DisplayName>
  - PUBLISHER DISPLAY NAME
    - <PublisherDisplayName>MyCompany</PublisherDisplayName>
  - PATH TO LOGO IMAGE
    - <Logo>images\icon.png</Logo>
  - PREREQUISITES (Minimum OS levels required before the application will be permitted to run)
    - <Prerequisites><OSMinVersion>6.2.1</OSMinVersion><OSMaxVersionTested>6.2.1</OSMaxVersionTested></Prerequisites>
  - DEPENDENCIES OF THE APP

2 – SOFTWARE DEVELOPMENT LIFECYCLE

Application Manifest File

- Contains…
  - CAPABILITIES OF THE APP (WHAT IS IT GOING TO NEED PERMISSION TO ACCESS Eg. NETWORK. GEOLOCATION, PHONE CONTACTS, CAMERA ETC.)
- <Capabilities> <Capability Name="internetClient"/> <Capability Name="musicLibrary"/> <Capability Name="videosLibrary"/></Capabilities><DeviceCapability Name="microphone"/> <DeviceCapability Name="webcam"/><DeviceCapability Name="usb"> <Device Id="vidpid:xxxx xxxx"> <Function Type="classId:xx xx xx"/> <Function Type="name:xxxxx"/> <Function Type="winUsbId:xxxxx"/> </Device></DeviceCapability><DeviceCapability Name="humaninterfacedevice"><DeviceCapability Name="pointOfService"/>

2 – SOFTWARE DEVELOPMENT LIFECYCLE

- Security: Permissions
  - Permissions grant the ability for some code to execute other code : we are not talking about user permissions here but code permissions!
    - .NET : Full Trust : Unrestricted access
    - .NET : Execution : Run code

2 – SOFTWARE DEVELOPMENT LIFECYCLE

- Security: User Authentication and Access
  - Users can log in via a variety of methods determined by the programmer
    - OAuth : very popular today (not in this exam!) : connects your website to any other method of authentication such as:
    - Google
    - Facebook
    - Windows Authentication : handled by Windows
    - Database Authentication : handled by the programmer
    - Anonymous : zero authentication

2 – SOFTWARE DEVELOPMENT LIFECYCLE

- Licensing
  - EULA End User License Agreement
    - Is what you agree to before you start using an application (I agree to these terms…)
  - Open Source
    - MIT License
    - SourceForge.Net
  - IDE Development Platforms
    - Class discussion on different types of IDE Environments and their pros and cons

2 – SOFTWARE DEVELOPMENT LIFECYCLE

- Overview of Software Dev Lifecycle
  - Analysis
  - Design
  - Build
  - Test
  - Release
  - Ongoing Maintenance
  - Documentation

2 – SOFTWARE DEVELOPMENT LIFECYCLE

- Building Your App : Using A SDK
  - Windows SDK
    - https://developer.microsoft.com/en-us/windows/downloads/windows-10-sdk
  - Android SDK
    - https://developer.android.com/studio/index.html
  - IOS SDK
    - https://en.wikipedia.org/wiki/IOS_SDK
    - https://developers.facebook.com/docs/ios/
  - Launcher Icon

2 – SOFTWARE DEVELOPMENT LIFECYCLE

- Debugging Your App : Different Tools

- Debugging : Different Types of Errors
  - Syntax Error
  - Logical Error
  - Exception

2 – SOFTWARE DEVELOPMENT LIFECYCLE

- Testing Your App
  - Unit Testing
  - Integration Testing
  - System Testing
  - Touch Screen Testing
    - Can use mouse Touch events
    - Can use an Emulator to simulate touch events
- Validating Your App
  - W3C Page Validation
    - https://www.w3.org/developers/tools/
    - http://ready.mobi/Unit Testing/

2 – SOFTWARE DEVELOPMENT LIFECYCLE

- Certifying Your App : Windows App Certification Kit
  - Please do some research on the following links:
    - https://msdn.microsoft.com/en-us/library/mt637086%28v=vs.85%29.aspx?f=255&MSPPError=-2147217396
    - https://developer.microsoft.com/en-us/windows/develop/app-certification-kit
    - https://msdn.microsoft.com/en-us/library/mt674655(v=vs.85).aspx
    - https://msdn.microsoft.com/en-us/windows/uwp/debug-test-perf/windows-app-certification-kit
    - https://msdn.microsoft.com/en-us/library/windows/hardware/br230771(v=vs.85).aspx

2 – SOFTWARE DEVELOPMENT LIFECYCLE

- Publishing Your App
  - Licensing Issues eg. EULA
  - Publishing To The Windows Store
    - https://developer.microsoft.com/en-us/store/publish-apps
  - Publishing To The Android Store
    - https://developer.android.com/studio/publish/index.html
    - https://developer.android.com/studio/publish/app-signing.html
  - Publishing To The IOS Store
    - https://developer.apple.com/library/content/documentation/IDEs/Conceptual/AppDistributionGuide/SubmittingYourApp/SubmittingYourApp.html#//apple_ref/doc/uid/TP40012582-CH9-SW1
    - https://code.tutsplus.com/tutorials/how-to-submit-an-ios-app-to-the-app-store--mobile-16812
    - https://developer.apple.com/app-store/review/guidelines/

3 – HTML

Introduction

- What is HTML?
  - It is a language used to transmit information across the internet from one machine to another
  - Data is transmitted from one machine (client) to another (server), and vice-versa
  - HTML data is split into two parts : the header information which tends to be INVISIBLE then the body information which produces the visible page
  - HTML is PLAIN TEXT
  - HTML can be EDITED WITH ANY TEXT EDITOR
  - HTML source code can be VIEWED FOR ANY PAGE (Inspect Element on any web page)

3 – HTML

Structure

- What is the structure of HTML?
  - HTML uses something called the DOM (Document Object Model) model to produce data which can be viewed (rendered) on a page
  - Data is stored on the server in text files called .htm files and then when it is sent to the client, the browser is a program which takes this data and produces output for the viewer to see

3 – HTML

Structure of a Page

- DEMO : Go to F12 in Chrome and open up the:
  - Window Object
  - Document Object
  - Body Object
    - As a class, view and discuss the components visible within the BODY object

- DOM Model = Document Object Model
  - It is a clear hierarchy of every object laid out correctly and finding its place on the page, so we can locate every item uniquely using a simple hierarchy system
  - WINDOW ==> DOCUMENT ==> HTML ==> BODY ==> ELEMENT ==> ATTRIBUTE ==> VALUE

3 – HTML

Development

- Who's in charge of the development of HTML?
  - W3C the World Wide Web Consortium is a group of companies and organisations which steer the development of HTML and other web-related standards
    - Website at http://www.w3.org
    - Follow at https://twitter.com/w3c

3 – HTML

Learning

- Where are the best places on the web to investigate and learn HTML?
  - W3C Official Site
  - Morzilla HTML reference
  - W3Schools to learn
  - CodeCademy to learn with dynamic editor
  - Code.org

3 – HTML

Building a Basic Web Page

- <tag></tag>
  - Used to open and close all documents within an HTML page
  - Every opening <tag> must have a corresponding </tag> closing tag
  - Some tags are ‘self closing’ such as ,input />

- <!DOCTYPE html.
- <html>
- <head>

Building a Basic Web Page

- <meta>
- ‘meta’ data is DATA ABOUT DATA so the <meta> tags will hold information about the web page and some of the data within it
- <meta charset=“UTF-8”> where UTF is a 2-byte character set used to represent all of the languages in the whole world. Using it we can represent any character in any language which is quite useful!!!

3 – HTML

Full List of Tags

- HTML tags
  - See Entire List of HTML tags at these locations

- Learning Material
  - http://www.w3schools.com/tags/
- Official Reference
  - https://developer.mozilla.org/en-US/docs/Web/HTML/Element
- W3C Official Reference
  - http://w3c.github.io/html-reference/elements.html#elements

3 – HTML

Testing for Validity of Tags

- CanIUse.com
  - http://caniuse.com
  - Use to see if a particular feature is supported by a particular browswer
- Rendering engines : be aware different browsers ‘render’ or display data using different mechanisms called ‘rendering engines’
  - https://en.Wikipedia.org/wiki/Comparison_of_layout_engines_(HTML)
  - Discussion : Blink is a newer engine run by Google, Opera and Samsung
    - https://en.Wikipedia.org/wiki/Blink_(web_engine)

3 – HTML

Simple Tags

- Class Discussion on Known HTML Tags
  - Use a discussion to lay out commonly known tags, either for discussion or to actually build some interesting components together as a class

- Examples to start us off:
  -
  -
  - <small>
  - <hgroup> contains <h1>…to… <h6> tags

3 – HTML

- Forms

- HTML <form> tag

- HTML <input> tag
  - Inputting data with the <input> element
  - See overleaf for list of different input elements

3 – HTML

Input Elements

- Text Field (type="text")
- Number Filed (type="number")
- Pattern
- Date Field (type="date")
- URL (type="url")
- Email (type="email")
- Password (type="password")
- Datalist – Used to provide 'autocomplete' as users find an item within a pre-determined list <datalist><option value="x">
- url – for entering a URL. It must start with a valid URI scheme, (for example http://, ftp:// or mailto:)

3 – HTML

Input Elements

- tel For entering phone numbers. It does not enforce a particular syntax for validation, so if you want to ensure a particular format, you can use pattern
- email For entering email addresses. By default it will only take one, but if the multiple attribute is provided, a comma separated list of email addresses is valid
- search A text input field styled in a way that is consistent with the platform's search field
- number For numeric input, can be any rational integer or float value
- color For choosing colors
- datatime – For entering a date and time value where the time zone is provided as GMT
- datetime-local – For entering a data and time value where the time zone provided is the local time zone
- date – For entering a date (only) with no time zone provided
- time – For entering a time (only) with no time zone provided

Input Elements

- week – For entering a date that consists of a week-year number and a week number, but no time zone
- month – For entering a date with a year and a month, but no time zone
- range – For number input, but unlike the number input type, the value is less important. It is displayed to the user as a slide control

Forms : Input Attributes

- Input attributes affect the way data is input
  - Placeholder – Telling the user what is expected with the <placeholder> tag
  - Required – Forcing data to be entered with the <required> tag
  - autocomplete: specifies if an element may have autocomplete enabled
  - autofocus : this element will get the focus when the page loads
  - list – refers to <datalist>
  - max / min eg. of number element
  - maxlength of input element

3 – HTML

Forms : Pattern Attribute

- <input pattern="…"> can force data to comply with a set input pattern
  - used to validate the user input against a very specific arrangement of letters, numbers and characters specified inside the pattern /... / string
  - pattern="\d{3}-\d{4}" forces ddd-dddd where d stands for any alphabetic character
  - let this site teach you (https://regexone.com/)
  - then move on to more advanced playing around with regular expressions here - http://regexr.com

3 – HTML

- Forms

  - Lab : Build an HTML form and use as many input elements as you can
  - Be sure to use the ‘placeholder’ attribute to tell the user what is expected in text fields
  - Be sure to use the ‘required’ attribute on some fields eg Surname
  - Bonus : can you work out how to force data to comply to a ‘Regular Expression’ pattern eg 3 numbers : use the ‘pattern’ attribute eg \d{3}

3 – HTML

- DataList Element

- Lab : add to your form the <datalist> element where a user can choose similar to the <select> element which has an <option> tag for each element. Do some research on the <datalist. Element and include it in your form

3 – HTML

- Tables

- Lab : build a quick table using
  - <table>
  - <caption>
  - <thead>
  - <tbody>
  - <tfoot>
  - <tr>
  - <td>
  - <col> for styling
  - <colgroup> for styling

3 – HTML

- Tables

- Bonus : depending on time : can you build a quick BOOTSRAP PAGE showing a NAVBAR, BASIC HORIZONTAL FORM AND A BASIC 2-ROW TABLE???

- Reason : HTML Tables are so OLD HAT but BOOTSTRAP is the WAY TO GO!

3 – HTML

New Elements

- HTML5 brought with it many new elements and we enjoy them today in web pages
- HTML5 Structure Elements
  - Nav
  - Header
  - Section
  - Article
  - Aside
  - Footer

3 – HTML

Semantic Elements

- These elements may not actually DO ANYTHING but they convey MEANING to the BROWSER PROGRAM and possibly to the END USER also
- A good example will be <header> or <footer> which don't actually do anything but can be used to clearly mark where the page header and footer go

3 – HTML

Semantic Elements

List Of HTML5 Semantic Elements Below:

- <article> Defines an article
- <aside> Defines content aside from the page content
- <details> Defines additional details that the user can view or hide
- <figcaption> Defines a caption for a <figure> element
- <figure> Specifies self-contained content, like illustrations, diagrams, photos, code listings, etc.
- <footer> Defines a footer for a document or section
- <header> Specifies a header for a document or section
- <main> Specifies the main content of a document
- <mark> Defines marked/highlighted text
- <nav> Defines navigation links
- <section> Defines a section in a document
- <summary> Defines a visible heading for a <details> element
- <time> Defines a date/time
- http://www.w3schools.com/TAGS/tryit.asp?filename=tryhtml5_time
- Outlining : automatically splitting page up into sections eg at https://gsnedders.html5.org/outliner/

3 – HTML

Media Elements

- HTML5 now has the ability to play audio and video files without the use of 3rd party plugins, natively from within the browser. This is a great step forwards from previous versions where it was quite hit-and-miss as to what media files would work in particular browsers eg Macromedia Flash or Silverlight, both of which are now no longer required nor used so much to build great web pages

3 – HTML

Media Elements

- <audio>
  - Now comes NATIVE IN HTML5!!!
  - preload
  - controls
  - autoplay
  - autoloop
  - metadata
  - canplaytype(type)
  - currentTime
  - duration
  - play()
  - pause()
  - File types supported
  - mp3
  - m4a
  - wav
  - ogg
  - aac

3 – HTML

<audio> example

```
<!DOCTYPE html>
<html>
<head>
<title>HTML5 Audio Example</title>
<script type="text/javascript">
function playAt(seconds){
var audio = document.getElementsByTagName("audio")[0];
audio.currentTime=seconds;
audio.play();
}
function restart(){
var audio = document.getElementsByTagName("audio")[0];
audio.currentTime=0;
audio.play();
}
</script>
</head>
<body>
Page 2 98-375 HTML5 Application Development Fundamentals
<audio controls="controls">
<source src="BuffaloChickenDip.mp3" />
Your browser does not support the audio element.
</audio>
<br />
<button title="Play at 15 seconds" onclick="playAt(4.5);" >4.5
Seconds</button>
<br />
<button title="Restart Audio" onclick="restart();">Restart</button>
</body>
</html>
```

Media Elements

- <video>
  - src multiple src elements can be used to define the file to be used to play the video. The browser will use the first valid source file that it can play.
  - width, height of the video
  - controls : if present then the play/pause/forward/rewind controls are visible on the screen for the user to use
  - poster : image to display while the initial video is downloading so the user is not staring at a blank screen but has something to look at
  - autoplay : automatically begins playing once the video has buffered (annoying adverts on pages do this also!)
  - play
  - pause
  - preload
  - seeking
  - volume
  - fullscreentoggle
  - captions/subtitles
  - Track

3 – HTML

- <video> example

- Class Challenge
  - Build a web page similar to the <audio> page using code similar to that below.
  - Bonus : see if you can do advanced effects like make a button to jump half way into the video

```
<video width="320" height="240" controls="controls">
<source src="travelVideo1.mp4" type="video/mp4" />
<source src="travelVideo1.ogg" type="video/ogg" />
Your browser does not support the video tag.
</video>

<video width="320" height="240" controls="controls" poster =
“FrenchLesson1.png”>
```

3 – HTML

Media Elements

Terms To Be Aware Of

- Codec
  - Codec is short for ENCODE / DECODE and is used to mean the art of taking a video or audio file, saving it for use as a single file (ENCODING) and then playing it back again on demand (DECODING). A CODEC is a means of doing this i.e. different CODECS are used on different platforms to produce the same result ie to play the file

- Compression
  - Is the art of making a file smaller
  - Data compression uses various techniques to reduce the size of data blocks
  - Video/audio/image compression may or may not reduce the image quality
  - Lossy WILL LOSE DATA (FOREVER!) eg BMP => JPG, WMA => MP3, WMV => MP4
  - Lossless means that ALL THE DATA IS SAVED even though the file takes up less space EG WMA => M4A

3 – HTML

Drawing Elements

- <img> tag : standard images
  - Just play around with creating a web page using a few images sourced from the web.
  - A couple of tags to consider:
    - <figure>
    - <figcaption>
    - <alt>
      - Used to display alternative text when the image cannot load for some reason

3 – HTML

Drawing Elements : Canvas

- There are two main ways to draw images on a screen.

- Raster Images
  - These use PIXELS to create an image point-by-point
  - Images can be quite detailed and hard to draw, but better where that is required eg high resolution gaming

- Vector Images
  - These are drawn using mathematics and are fully scalable to large and small screens.
  - Better for static images eg architecture drawings

3 – HTML

Drawing Elements : Canvas

- <canvas>
  - Uses Javascript to draw images point-by-point
  - Better for complex scenes, real-time mathematical animations, high-performance elements (filters and ray tracers), games and video manipulation
  - Canvas is better for real time data
  - Drawn images cannot be modified by scripting and CSS
  - Image types are bitmaps and not easily scalable.
  - Must add an ID to add the element to the HTML DOM
  - Must specify which application to run when canvas is clicked

3 – HTML

Drawing Elements : Canvas

- <canvas>
  - First Ever Canvas Page
  - Add this to any standard HTML page

  - <canvas id="canvasElement" width="480" height="320"></canvas>

  - Add code to make the following effects happen:
  - Context (creates the object in Javascript)
  - moveTo (moves cursor to a point)
  - lineTo (draws a line)
  - Stroke (puts the colour on the page)

3 – HTML

Drawing Elements : Canvas

- <canvas>
  - <script> has two functions
  - Firstly : draw interface
  - Secondly : process click event

  - Canvas : Example : Game
  - https://msdn.microsoft.com/en-us/library/gg589521?f=255&MSPPError=-2147217396

3 – HTML

- Drawing Elements : Canvas

- <canvas>
  - Class Example : Test for browser compatibility

```
try{
document.createElement("canvas").getContext("2d");
document.getElementById("support").innerHTML = "HTML5 Canvas is supported in your
browser.";
}
catch (e) {
document.getElementById("support").innerHTML = "HTML5 Canvas is not supported in your
browser.";
}
```

3 – HTML

- Drawing Elements : Canvas

- Sample Canvas Application
  - Sample Canvas Application
  - See lab 2.2 for Sample Canvas Application
  - Run this code eg at
  - http://www.onlinehtmleditor.net/

3 – HTML

Drawing Elements : Canvas

```
<!DOCTYPE html>
<html>
<head>
<title>HTML5 Canvas Diagonal Line</title>
<script>
function drawDiagonal() {
var canvas = document.getElementById('diagonal'); //creates a css id
var context = canvas.getContext('2d');
// Save a copy of the current drawing state to be restored upon exit
context.save();
// Move the drawing context to the right and down using transformation
context.translate(70, 140);
// Draw the same line as before, but using the origin as a start
context.beginPath();
context.moveTo(0, 0);
context.lineTo(70, -70);
context.stroke();
// Restore the old drawing state
context.restore();
}
window.addEventListener("load", drawDiagonal, true);
</script>
</head>
<body>
<header>
<h1>Check out the new canvas element</h1>
</header>
<canvas id="diagonal" width="400" height ="400";>
You need to upgrade your browser to see the canvas
</canvas>
</body>
</html>
```

3 – HTML

Drawing Elements : SVG

- <svg>
  - Better for static images and high-fidelity documents for viewing and printing.
  - Objects are part of the Document Object Model (DOM); they can be modified by scripting and CSS at any time.
  - Vector graphics are scalable.
  - Best used to create static images which scale well to any screen size, large or small

3 – HTML

Drawing Elements : SVG

- <svg>
  - Lab : First ever SVG image
  - Use the <svg> element to create an image on the page.
  - Check out any differences with <canvas> for the following words – context, moveTo, lineTo, stroke

  - Sample Game
    - https://msdn.microsoft.com/en-us/library/gg589521?f=255&MSPPError=-2147217396

3 – HTML

- Drawing Elements

- <canvas> vs <svg>

- Which is better? Which is faster???

  - Which is better? Canvas or SVG?

  - https://msdn.microsoft.com/en-us/library/gg193983%28v=VS.85%29.aspx?f=255&MSPPError=-2147217396#Using_Canvas_AndOr_SVG

3 – HTML

- Research

- Other elements to be aware of

- New Form Elements in HTML5
  - What do the new form elements (progress, meter, datalist, keygen, and output) add to a web page?
  - Progress is a control the shows the status of a task being completed such as a file being uploaded
  - Meter is used for scalar measurement within a known range such as temperature or weight measurement
  - Datalist is used to show a list of options; it is used with the new list attribute for the input element

3 – HTML

- Research

- Keygen is a control for key-pair generation.

- The output control displays the result of a calculation; an example would be the sum of the values of two input elements.

- Alternative To Table - Definition List
  - Note : Can be called either Definition List or Description List
  - <dl> Definition List
  - <dt> Definition Tag
  - <dd> Definition Detail

3 – HTML

- Research

- <command>
- <mark>
- <time>
- <details>
  - Provides a drop-down for content
  - http://www.w3schools.com/tags/tryit.asp?filename=tryhtml5_details

3 – HTML

Multimedia Input

- HTML5 has the native ability to talk to multimedia items on the native device
  - <input> tag to connect to camera, video and microphone
  - input type="file" accept="image/*;capture=camera"
  - input type="file" accept="video/*;capture=camcorder"
  - input type="file" accept="audio/*;capture=microphone“

  - Class discussion – are these used much – Hint : No? Probably best to build a native APP rather than HTML for this

4 – CSS

Cascading Style Sheets

- What is CSS?
  - It is used to control the display of HTML elements
  - style="background-color:yellow" can be used to style just one line of HTML (inline styling)
  - <style> tag is used inside a page
  - <link rel="stylesheet" href="style.css"> is used to link to an external CSS page
  - Selecting items for display with the ID or Class attributes
    - h1{} will style all h1 elements
    - #thisID will style element marked 'id=thisID'
    - .thisClass will style element marked 'class=thisclass'

4 – CSS

Selecting Items for Styling

- CSS Lab 3.1
  - If you are not familiar with CSS then please have a look at this lab.
  - If you are familiar with CSS then build a basic web site using
    - Navbar
    - Main
    - Footer
    - - all with some content, and style the page

4 – CSS

- Displaying Items on a Page

- Items can be either BLOCK or INLINE elements

- BLOCK elements will take up the MAXIMUM WIDTH ON A PAGE
  - <h1>..<h6>

- INLINE elements will only take up the width of the natural element and no more
  -

Displaying Items on a Page

- Items can be positioned on a page relative to each other or the page
  - Position:absolute
    - Fixed on the page relative to top left (0,0) point ie absolute positioning (does not move when page moves)
  - Position:relative
    - Relative to other items ie relative positioning
  - Position:static
    - Default 'static' positioning

4 – CSS

- Displaying Items on a Page

- Floating Left and Right

- Float:Left
  - Items all flow from the left

- Float:Right
  - Items all flow from the right

- Clearing a float with the clear: both property
  - Causes the next element to be displayed below as if it were a block level element, taking up the full row

4 – CSS

Overflow (when a Text Box is Full)

- Overflow : What to do when content exceeds the size of a box
  - scroll : scroll bar becomes visible
  - hidden : extra text is hidden and truncated and unavailable to view by the user
  - visible : extra text flows outside the box (very messy)

4 – CSS

- Differences between Browsers

- CanIUse.com

- Vendor Prefixes
  - Unique to the particular browser
    - -ms- Microsoft
    - -o- Old Opera
    - -moz- Firefox
    - -webkit- Chrome, Safari, New Opera

4 – CSS

- Graphical Effects

- Rounding boxes with border-radius element
  - border-radius eg at https://msdn.microsoft.com/library/gg589503.aspx

- Creating shadows and blur with box-shadow element
  - box-shadow: 10px 10px 5px #888888;
  - h-shadow
  - v-shadow
  - Blur
  - Spread
  - colour
  - inset

4 – CSS

Tooltips

- Tooltips are handy little ‘mini-pop-ups’ which appear when a user hovers over a particular element on the screen.

- They appear in order to supply more information to the user to help them understand better what the information on the screen is trying to say
  - Tooltips are a pop-up box which appear on the screen eg http://www.w3schools.com/howto/tryit.asp?filename=tryhow_css_tooltip

4 – CSS

- Opacity
- Opacity is the OPPOSTITE OF TRANSPARENCY!
- Opacity runs from 0 (fully transparent) through to 1 (fully opaque)
- Lab : can we make an image go 0.5 opaque by default when page is loaded, but then when the user moves the mouse over the item, it becomes fully opaque

4 – CSS

- Colours

- HTML and CSS colours are specified in several different ways

- RGB stands for Red Green Blue
  - Normally each colour is represented by a number from 0 through 255 eg 255,0,0 produces red.

- RGBA stands for Red Green Blue Alpha
  - Normally each colour is represented by a number from 0 through 255 eg 255,0,0 produces red, but in addition the transparency is also represented by the alpha number between 0 (transparent) and 1(opaque)

4 – CSS

- Fonts

- CSS can be used to style the fonts on the page which gives great power in the display of text.

- What are fonts?
  - Fonts are files which are stored on a computer. When a particular letter is requested to be viewed on the screen it is 'rendered' (displayed) with the help of the font files already present on that computer.

- What is typography?
  - Typography is the art of laying out text correctly on a page to suit font, styling, colour and spacing etc. to maximise the visual appeal of the text.

4 – CSS

- Fonts

- Discussion, Introduction And Demo
  - Simplest fonts
    - Arial
    - Serif
    - Monospaced fonts like Courier and Consolas

- Some Examples Of The Latest Fonts
  - Have a look at the fonts available at
  - https://www.myfonts.com/hotnewfonts/

Fonts – File Types

- What files are used to create fonts?

- Common ones are
  - EOT Embedded Open Font
  - TTF True Type Font
  - OFF Open Font Format
  - WOFF Web Open Font Format

4 – CSS

Fonts – File Types

Support For Fonts

Support for fonts is not universal but differs from operating system to operating system.

- EOT : Microsoft only!
- TTF : very good - see http://caniuse.com/#feat=ttf
- OTF : very good - see http://caniuse.com/#search=otf
- WOFF : very good - see http://caniuse.com/#search=woff
- SVG fonts : Apple only - see http://caniuse.com/#feat=svg-fonts

4 – CSS

- Fonts : Using on a page

- Using A Font On A Web Page
  - <font> tag is now deprecated.

- Fonts are set on a page using CSS
  - font-family declaration

- eg font-family:arial
  - to include Arial font on a page.

Web-Safe Fonts

Web-safe Fonts

- These fonts are included in most operating systems so are counted as safe to use on your web page with the minimum of fuss.
  - http://www.w3schools.com/cssref/css_websafe_fonts.asp
  - https://www.tutorialspoint.com/html/html_fonts_ref.htm
  - http://www.cssfontstack.com/

4 – CSS

- Web Fonts using @font-face

- Web Fonts

- These fonts have to be installed in order to be used.

- They are installed for a web page by downloading the font then including the link to that font on the website using the CSS @font-face command

- CSS @font-face { font-family: FONT-NAME; src:url('path-to-font-file.eot'); font-weight: normal; font-style: normal; }

4 – CSS

- Font Stacks

- Font stacks are used to declare multiple fonts in case the first (preferred) ones are not supported

- font-family: Chunkfive, Georgia, Palatino, Times New Roman, serif;

- Examples of web fonts
  - http://www.cssfontstack.com/Web-Fonts
  - https://fonts.google.com/

4 – CSS

WOFF Web Open Font Format

- WOFF Web Open Font Format
  - Is an open standard for managing fonts
    - https://developer.mozilla.org/en-US/docs/Web/Guide/WOFF
- Possible Lab :
  - Create, download and use fonts at Font Squirrel
    - https://www.fontsquirrel.com/
  - Download and use ChunkFive web font (instructions at)
    - http://sixrevisions.com/css/font-face-guide/
    - Font at http://www.fontex.org/download/Chunkfive.otf#

4 – CSS

- Box, Flex, Grid Layouts
  - CSS Layouts
  - Several different layouts exist to lay out pages well for HTML
    - Standard Box Model (This is the default on web pages)
    - Flexbox Layout
    - Grid Layout

Box Model

- CSS Box Model
  - This is default for web pages
- Check out F12 to view the box details
  - Outline : Does not take up space!
  - Margin : Between items
  - Border : Around items
  - Padding : between content and border
  - Content : width of item in box

4 – CSS

- FlexBox Layout

- Enabled with display:flexbox

- Manipulating child boxes with
  - flex-wrap
  - flex-pack
  - flex-direction
  - flex-flow
  - flex-order

4 – CSS

- Grid Layout

- Enabled with display:grid

- Displaying items with
  - grid-columns
  - grid-rows
  - grid-template
  - grid-position

- Spacing with 'fr' fractional element

4 – CSS

- Multi-Column Layout

- Used for a newspaper or magazine article to lay out content neatly on a page but to ‘flow’ from one column to the next

- Displaying items with
  - Column-count
  - Column-gap
  - Column-rule

- Lab 3.3.8

4 – CSS

- Multi-Column Layout
  - https://www.w3.org/TR/css3-multicol/
  - Use instead of CSS Regions (see later)
  - Examples Of Multi Column Layout
    - http://people.opera.com/howcome/2013/02-reader/
    - http://people.opera.com/howcome/2013/reader/news/h1.html

4 – CSS

Colour Gradients

- Creating change of colour called a gradient where a colour or multiple colours change in a controlled fashion across an object or the screen
  - Linear Gradient : colours change between two points
  - Circular Gradient : colours change to produce a circular pattern
  - Repeating Gradients where colour change repeats predictably over and over.

4 – CSS

- Colour Gradients

- linear-gradient
- radial-gradient
- repeating-linear-gradient
- repeating-radial-gradient
- color stop

5 – JAVASCRIPT

Introduction

- What is Javascript?
  - Javascript is the ability to run code within your browser.
  - Unlike HTML which is a markup language and is used to display elements, and CSS which is a styling element with a job of enhancing the display of items, Javascript can do the full range of programming tasks.

5 – JAVASCRIPT

- Introduction
  - Running Javascript
  - Inline
    - <button onclick="alert('hi');" will run the javascript code when button has been clicked by the user. Not recommended to write this kind of code.
  - Internal
    - <script>function x(){}</script>
    - Script is embedded on the page
  - External
    - <script src="myScript.js" /> runs Javascript sourced from a page outside the host page.

5 – JAVASCRIPT

- Data Types – Variables

  - Number var x=1;
  - String var x = “text”
  - Boolean var x = true;
  - Array var x = [1,2,3]
  - Object var x = {“a”:”1”,”b”:”2”}
  - Null var x = NULL nil value
  - Undefined var x; x is UNDEFINED

5 – JAVASCRIPT

Communicating with User

- Alert
  - Simply produce a message on the screen
- Confirm
  - Ask for Yes/No response and test for output true or false
- Prompt
  - Ask for user to enter text. Can provide default text for user.

5 – JAVASCRIPT

- Receiving Data from the User – Events and Event Listeners

- Events

```
<button onclick="fnDoThis()">
    Alert
```

- Simply produce a message on the screen
  - Event Listener
  - Document.getElementById('x').addEventListener('click',function(){});

- Event Handler is the function which 'handles' or executes when event fires

5 – JAVASCRIPT

- Events
  - onload()

5 – JAVASCRIPT

- Setting CSS

- CSS can be set using Javascript

- Example : Inline Javascript
  - Set Background

```
document.body.style.backgroundColor= "green";
```

- Example : Internal Javascript
  - show/hide data with display:none/visible

- Lab : create a document with a button to hide and show a div.

5 – JAVASCRIPT

- Functions

- Function x(){}
- Y=x() will call the function
- X=function(){} is another way of writing the function x
- Return
  - Used to pass data back to the line of code which called the function.

```
Function x(){
    return {“a”:1}
}
```

5 – JAVASCRIPT

- Parameters

- Function x(a,b,c){//use a,b,c in the function}
- Y=x(1,2,3) will call the function
- (1,2,3) are the parameters passed in to the function which get received as variables a,b,c

5 – JAVASCRIPT

- Storing Data in JSON Objects

- JSON data structure

```
{
   “a”:1,
   “b”:”some text”
}
```

- creating JSON objects from strings using JSON.parse
- creating strings from JSON objects using JSON.stringify
- See example in section 4.4

5 – JAVASCRIPT

Selecting Elements

- Elements are selected in the same way that other DOM elements are selected by standard CSS
- Document.getElementById('x')
- Document.getElementsByTagName('p')
- Document.getElementsbyClassName('myclass')
- Document.forms.form1.childNodes[0] will select the first item in given form.

5 – JAVASCRIPT

- Changing Elements

- Using Javascript To Alter HTML DOM Elements
  - innerHTML
  - createElement
  - var newRadioButton = document.createElement("<INPUT TYPE='RADIO'
  - NAME='RADIOTEST' VALUE='First Choice'>")
  - insertBefore/After
  - document.body.insertBefore(newRadioButton);
  - appendChild

5 – JAVASCRIPT

Using jQuery

- jQuery is a powerful Javascript library which adds the ability to use Javascript features with the minimum of code
- <script src="jquery.js" ></script>
- <script src="http://online cdn ></script>
- Once the script has been placed on the page then code is run using
- $(document).ready(function(){});

5 – JAVASCRIPT

- Using jQuery

- $('selector').doSomething() eg
- $('#item1').hide()

- Lab : discover and apply some other cool jQuery features to your page. Check out the word 'toggle' which can add a feature then remove it again repeatedly.

5 – JAVASCRIPT

- setTimeout setInterval

- Var x = setTimeout(function(){},2000);
- Var x = setInterval(function(){},2000);

- Lab : Metronome : can you create a metronome which flashes on the screen to a set interval? Can you create a button to stop and start the metronome? Can you set a button to increase the speed? A second button to decrease the speed? Real world – if you have hosting space online can you also add an audio file and make it beep, and upload it?

5 – JAVASCRIPT

- setTimeout setInterval

- Setting a delay with setTimeout
  - Var x = setTimeout(function(){},2000);

- Setting regular intervals with setInterval
  - Var x = setInterval(function(){},2000);

- Lab : Metronome : can you create a metronome which flashes on the screen to a set interval? Can you create a button to stop and start the metronome? Can you set a button to increase the speed? A second button to decrease the speed? Real world – if you have hosting space online can you also add an audio file and make it beep, and upload it?

5 – JAVASCRIPT

- Web Storage

- HTML is a STATELESS PROTOCOL

- It does not have any inherent 'memory' of previous page visits

5 – JAVASCRIPT

- Storing Data Temporarily : Cookies, Application and Session State Data

- Storing Data
- Storing Data Between Clicks - The 'State' Of The Application or Session
- Persistence is the ability to store data
- Let’s look at 3 ways to store data
  - Cookies
  - Application State
  - Session State

5 – JAVASCRIPT

- Storing Data Temporarily : Cookies, Application and Session State Data

- Cookies
- Cookies are small text files which can be used to transmit data between the client and the server which reflect the state of the session or application.
- Small file size (4kb)
- Sent with every HTTP request
- Can be accessed by both client and server code

5 – JAVASCRIPT

- Storing Data Temporarily : Cookies, Application and Session State Data

- Application State

- The data pertaining to the application as a whole for all users
- For example,
  - database path
  - Max RAM for this application
  - Number of available database connections

5 – JAVASCRIPT

- Storing Data Temporarily : Cookies, Application and Session State Data

- Session State

- The data pertaining to one user after they log in, relevant for that user for that logged in 'session'

- Data is valid until they either log out or they stop using the application in which case after a certain time span the session will automatically log them out and the session data will cease to exist.

5 – JAVASCRIPT

Storing Data Permanently : LocalStorage

- Local storage is the ability to store files and data permanently on the client machine.
- This is fully persistent ie it has no time span after which the data will expire, so in that sense the data is permanent
- If the client clears their cache and temporary storage areas out which they are more than welcome to do at any stage, the data will be erased.

5 – JAVASCRIPT

- Storing Data Permanently : LocalStorage

- Can only be accessed by client browser code (not server-side code!)

- Can timestamp and purge items from local storage cache in order to manage space effectively

- Lab – PDF - does your browser support local storage???

5 – JAVASCRIPT

Storing Data Permanently : Browser Cache

- The browser cache is used to store a copy of a website on the local computer before it is displayed to the end user.

- Files such as .js and .css files which are referenced as part of the web page, for example a jQuery library file, are stored in the user's cache of the browser.

5 – JAVASCRIPT

Storing Data Permanently : AppCache

- AppCache is a permanent store on the client computer that can be used to ensure files don't get downloaded more than once. If they are needed again they can be used from the cache
- Files are specified for caching using the MANIFEST.appcache file
- Differs from the browser cache in that it also specifies behaviour when the app is offline, which the browser cache doesn't

5 – JAVASCRIPT

- Storing Data Permanently : AppCache

- Lab : Go to
Chrome://appcache-internals
And have a look at some manifest files.

- Check out some links present in the appcache file and ensure they do link to real, valid files

5 – JAVASCRIPT

- Storing Data Permanently : AppCache

- AppCache Manifest File :
  - CACHE
  - NETWORK
  - FALLBACK
- CACHE section specifies files to cache, explicitly named
- NETWORK section specifies files to explicitly force to be available online only
- FALLBACK specifies display files to use when the application is offline and the user has to be notified

5 – JAVASCRIPT

- Storing Data Permanently : AppCache

- AppCache Manifest File :
  - CACHE
  - NETWORK
  - FALLBACK
- CACHE section specifies files to cache, explicitly named
- NETWORK section specifies files to explicitly force to be available online only
- FALLBACK specifies display files to use when the application is offline **and the user has to be notified**

5 – JAVASCRIPT

- Storing Data Permanently : AppCache

- <html manifest="MANIFEST.appcache">

- Instead of listing a file inside the MANIFEST.appcache file, the above code can be put on an HTML file to enable a page to be cached or stored locally for offline use.

- https://www.html5rocks.com/en/tutorials/appcache/beginner/

5 – JAVASCRIPT

- Interacting with the File System : Loading and Saving Files with HTML File API

- HTML5 now has a File API which makes it much easier to handle files, both uploading and downloading and manipulating files

- HTML5 File API
  - Having The User Select Files For Upload With <input type='file'>
    - Handling One File
      - File
      - input type="file" direct from HTML
    - Handling Multiple Files
      - FileList array is the list of files selected for input

5 – JAVASCRIPT

- Interacting with the File System : Loading and Saving Files with HTML File API

- HTML5 File API
  - Handling Large Files
    - Blob
  - Handling Text Files With FileReader
    - FileReader
  - the array 'FileList'

5 – JAVASCRIPT

- Interacting with the File System : Loading and Saving Files with HTML File API

- Validation Of File Type before loading

5 – JAVASCRIPT

- AJAX
- Sending And Receiving Data Using AJAX
- XMLHttpRequest object

5 – JAVASCRIPT :DRAWING WITH CANVAS

- Drawing with Canvas

- Canvas is our ‘palette’ which is manipulated, live, with Javascript to produce dynamic effects on our screen and is also used to produce games

- Clock Example

- Animated Box Example HTML5 File API : User Upload

5 – JAVASCRIPT :WEB WORKERS

- Web Workers

- Creating Separate Processes To Get Work Done : Web Workers
  - Worker environment separate to that of the DOM so can execute Javascript but can't manipulate DOM elements
    - "Isolated Thread"
  - Sending messages to a web worker thread
  - Receiving messages back from a web worker thread

5 – JAVASCRIPT

- Interacting with the hardware on your device using HTML5: getUserMedia()

- Accessing Your Camera Using Native HTML5
  - navigator.getUserMedia() to interact with devices eg to input data from a
    - camera - image
    - camera - video
    - microphone - audio

5 – JAVASCRIPT

- Interacting with the hardware on your device using HTML5

- https://www.html5rocks.com/en/tutorials/getusermedia/intro/

```
function hasGetUserMedia() {
  return !!(navigator.getUserMedia ||
        navigator.webkitGetUserMedia ||
            navigator.mozGetUserMedia ||
                navigator.msGetUserMedia);
}

if (hasGetUserMedia()) {
    // Good to go!
  } else {
    alert('getUserMedia() is not supported in your browser');
```

5 – JAVASCRIPT

- Interacting with the hardware on your device using HTML5

- Geolocation API
  - Detect position of phone on the globe
  - getting your current position using navigator.getCurrentPosition()
  - Watching your position using navigator.watchPosition()
- Accelerometer
  - Accessing Your Accelerometer using HTML5

5 – JAVASCRIPT

- Web Sockets

- Talking Across The Web : Using Web Sockets

- Full Duplex 2-way communication across the web using websockets protocol similar to HTTP protocol but without the overhead of headers each time data is transferred
- ws:// protocol
- wss:// secure protocol

5 – JAVASCRIPT

- Web Sockets

- Uses of WebSockets : Real Time eg Chat or Stockbroker Applications
- Creating a socket
- Opening and closing a socket
- Sending a message with a socket