HTML5 Demos

## Pre-Flight Check

* Hide clutter on desktop
* Make sure demos back to start
* Load Snippets:
  + "C:\Users\csell\Dropbox\ApplicationDevelopmentWithHTML5 - CIW\Demos\csDemoSnippets.txt"
* Open Visual Studio and Start the WebServer
* Load WebMatrix for Canvas Demo and startup IIS Express.

Semantic Markup

## Show off VS 2010 Features (HTML5 Schema, Intellisense)

* 1. **File| New MVC3** **Application**
     1. Show that the **MVC3 Tools** update includes the ability to select an **HTML5 Markup option**
  2. You will notice the validation of the document might be wrong based on what the editor is checking for. Notice the green lines.

<!DOCTYPE html>

<html>

<head>

     <meta charset="utf-8" />

     <title>Hello from</title>

</head>

<body>

     <p> HTML5!</p>

     </body>

</html>

* 1. **Update** the DocType validation
     1. In Visual Studio
        1. “**Tools | Options**” 🡪 “**Text Editor | HTML | Validation**” and change the target to **HTML5**.
  2. **Open** the HTML5 Page and show intellisense for aside tag and other HTML5 tags

<article>

     <p>HTML5!</p>

    </article>

## Basic Blog Page

1. **Open** up BlogPage.htm
2. **Walk** through the document structure.
   1. Talk about Sections, Article, Aside, Footers etc.
3. **Open** css/style and show how we are styling all of those types
4. **Run** the site

## Run the site in IE8 and show it broken

1. F12 **change** document mode to IE8
2. Rounded corners are gone, everything is unstyled

## Modernizr and Polyfills

1. **Show** modernizr-1.7.js in the JS folder, then **drag** a reference to BlogPage.htm

<script src="../js/modernizr-2.0.6.min.js" type="text/javascript"></script>

1. **Refresh** the page and show that things have gotten a little bit better because Modernizr has added all of the HTML elements to the DOM so that my styles will show up properly.
2. But I still don’t have rounded corners. To get rounded corners, I can go a step further.
   * 1. **Load** jQuery
        1. **add** jQuery to load locally but could use from a CDN like Microsoft or Google.
           1. Explain why load from a CDN like Microsofts
        2. We’re going to use jQuery to select elements in our document
        3. **Show** locally and **drag in**.

<script src="../js/jquery-1.5.2.min.js" type="text/javascript"></script>

* + 1. **Add** in the snippet to check Modernizr for the
       1. test: Modernizr.borderradius,
       2. nope: ‘js/jquery.corner.js’,
       3. Apply based on polyfill

<script type="text/javascript">

Modernizr.load({

     test: Modernizr.borderradius,

nope: 'js/jquery.corner.js',

callback: function() {

$("article").corner();

$("figure").corner();

}

});

</script>

* + 1. **Run** the site again and show the new rounded corners.
    2. **Go** to the **network tab** and turn on **capturing**. Show that the plugin is downloaded.
    3. **Change** the document mode back and show how the script isn’t loaded.

## Demo F12 Developer tools in IE9

1. **Show** that we’ve looked at the network tab and the html tab, as well as the ability to change the Document mode and browser mode.
2. **Show** that modernizr is minified and that this is a big problem when debugging production issues.
   1. Show the **Tools, Format JavaScript** option.

## Other Tags

1. Mark
   1. **Surround** some text with <mark>
   2. **Refresh** page with the highlight.
   3. Explain how you might use this in a search to highlight places in the page.
2. Time
   1. **Show** the existing time.
   2. Search engine would use the machine readable time to become more intelligent about the information in the document
3. Ruby
   1. Nothing to do with the language
   2. **Open** ruby.htm
   3. **Show** the markup and how it works.

Canvas and SVG

## Canvas Paintball Demo

1. IETestDrive:
   1. <http://ie.microsoft.com/testdrive/Performance/Paintball/Default.html>
2. **Show** in Chrome and IE, and IE 10 PP
3. **Hit** F12 and show off the canvas element

## [IF FAST] Create a simple canvas that we manipulate ourselves

1. **Open** StickFigure.htm and walk through the code.

## Data Vizualization with Canvas

* 1. **Open** Chart.html and show the table with the charting data we have. Run it with no styling
  2. **Add** jquery, visualize.jquery and the two Stylesheets
  3. **Add** the script snippet to call visualize and **run** the page

    <script type="text/javascript">

        $(document).ready(function () {

            $('#sales').visualize();

        });

    </script>

* 1. **Hit** F12 and show that there is a <canvas> element here in the background.
  2. Now **hide** the table.

$('#sales').hide();

* 1. **Refresh** the page.
     1. Now we have a nice graph
     2. Change the Document Type to IE8 and show the JS Error that happens when getContext() is called.
     3. We break now because we have no canvas support.
  2. **Add** Modernizr
     1. **Drag** a modernizr.js reference into the page
     2. **Modify** the script to check for Canvas and polyfill it.

<script type="text/javascript">

        $(document).ready(function () {

            var drawChart = function () {

                $('#sales').visualize();

                $('#sales').hide();

            };

            Modernizr.load({

                test: Modernizr.canvas,

                nope: "js/excanvas.js",

                complete: function () {

                    drawChart();

                }

            });

        });

    </script>

* 1. Comment on how this canvas method is accessible because the main data still exists on the page

## SVG

* 1. **Open** SVG.html and show the two xml-based elements
  2. **Run** the page
  3. Now **add** CSS to style them, and set the fill property on the Style logo

<style>

svg {

padding: 5px;

background-color: #CCCCCC;

border-radius: 10px;

-ms-transform: scale(.75);

box-shadow: 4px 4px 10px rgba(0, 0, 0, 0.5);

}

svg:hover {

-ms-transform: scale(.9);

}

svg#style path {

fill: #009933;

}

</style>

* 1. [IF FAST] Talk about the –ms-transform property and box-shadow

Audio and Video

## Show off WebMatrix HTML5 Features

* 1. **Open** Video site as a WebMatrix Site
  2. **Explain** what WebMatrix is and how it relates to the eco-system
  3. **Create** a new page called Video.html
  4. **Show** the default HTML5 markup in WebMatrix and Intellisence

## Add Full Video Tag

1. **Add** video tag

<video>

<source src="video/video.mp4" type='video/mp4' />

</video>

1. **Launch** in browser
   1. At this point the video won’t play
2. **Add** the controls and autoplay attributes to the video element
3. **Launch** in browser
   1. Still won’t run because of mime mappings
   2. **Add** a web.config and mime maps to support video

<system.webServer>

<staticContent>

<mimeMap fileExtension=".mp4" mimeType="video/mp4"/>

<mimeMap fileExtension=".webm" mimeType="video/webm"/>

</staticContent>

</system.webServer>

1. **Launch** in browser, will play
2. **Add** the rest of the attributes
   1. id, poster, loop, preload, playbackRate, width

<video id='movie'

poster="images/big-buck-bunny\_poster.jpg"

autoplay

controls

loop

preload=auto

playbackRate="1"

width="800">

## JavaScript to Manipulate Video

1. Add Script to play, pause and ++ the playbackRate when the video is clicked

<script type="text/ecmascript">

var rate = 1;

var v = document.getElementById('movie');

v.onclick = function () {

if (v.paused) {

v.play();

v.playbackRate = rate++;

} else {

v.pause();

}

};

</script>

## Fallbacks, and older browser fallbacks

1. At this point we have only been working with one codec.
2. What if that codec isn’t on the target machine?
3. Let’s **add** another fallback to ‘webm’

<source src="video/video.webm" type='video/webm' />

1. What about a browser that doesn’t support any of that?
2. Fallback to Silverlight or Flash
3. **Add** script to fallback to Silverlight (goes to different video)

<object data="data:application/x-silverlight-2,"

height="288"

type="application/x-silverlight-2"

width="512">

<param name="minRuntimeVersion" value="4.0.50401.0" />

<param name="source" value="http://channel9.msdn.com/scripts/Channel9.xap?v=1.5" />

<param name="initParams" value="mediaurl=http://media.ch9.ms/ch9/06c8/8113d5d7-acc7-4890-affc-9ebe016c06c8/Part8MvcScaffolding\_2MB\_ch9.wmv,thumbnail=http://media.ch9.ms/ch9/06c8/8113d5d7-acc7-4890-affc-9ebe016c06c8/Part8MvcScaffolding\_512\_ch9.jpg,deliverymethod=progressivedownload,autoplay=false,entryid=8113d5d7acc74890affc9ebe016c06c8" />

</object>

1. **Reload** the page in IE8

Geolocation

## Basic Geolocation

1. **Launch in Browser** ~/Geolocation/Geolocaiton.htm
   1. **Call out the Browser Alert.** 
      1. Geolocation is opt-in only.
      2. Sites and browsers cannot use it without your permission.
      3. **Show** what happens when you decline.



1. **Select** Allow
2. Talk about accuracy and compare to that of the phone.
   1. Yelp Example.
3. **ReRun** in new tab but **disallow**
   1. Nothing happens
4. **Change Broswer DocType**
   1. **Hit** F12
   2. **Change** to run in **IE8.**
   3. Show errors.
5. Up to this point we have been talking about pollyfilling, but there is another technique
   1. Graceful degradation or progressive enhancements
6. **Add** Modernizr support
   1. Gracefull fallback toa form that allows the user to enter their location.
   2. Show Modernizr Referenced on the page
   3. **Uncomment** the try block with the modernizr references.
   4. Also show that this mechanism is being used if the user says no.
7. **ReRun** showing form usage.

Storage

1. **Launch** in browser ~/ Storage/localStorage.html
2. Use local storage to save user preferences from a form (copy snippet) and show saving

$('#save').click(function () {

window.localStorage.name = $('#name').val();

window.localStorage.email = $('#email').val();

window.localStorage.color = $('#color').val();

window.localStorage.catOrDog = $('#catOrDog').val();

window.localStorage.freeform = $('#freeform').val();

$('#save').val('Done!');

//activateSettings();

});

1. **Run**, **save**, Hit **F12** and **show** the window.localStorage object in the console
   1. **Change** a value(s) in the console to show the interaction.
2. **Add** snippet to check for values to populate the form

$('#name').val(window.localStorage.name);

$('#email').val(window.localStorage.email);

$('#color').val(window.localStorage.color);

$('#catOrDog').val(window.localStorage.catOrDog);

$('#freeform').val(window.localStorage.freeform);

//activateSettings();

1. **Close** and **re-open** the browser and show things preserved
2. **Add** activateSettings snippet to act on preferences

function activateSettings() {

if (window.localStorage.color) {

$('input').css('background-color', window.localStorage.color);

}

if (window.localStorage.catOrDog) {

if (window.localStorage.catOrDog === 'cats') {

$('body').css('background-image', 'url("images/kitty.jpeg")');

} else {

$('body').css('background-image', 'url("images/dog.jpeg")');

}

$('body').css('color', 'white');

}

}

CSS3

1. **Show off Expression Web 4 SP1 Features**
   1. Open Demos and show new intellisense in any file
   2. Click New Style… and show the style editor dialog
   3. Open style.css and show pseudo-class selector intellisense by typing “input:”
2. **Add rounded corners to the regions of my form**
   1. Open BlogPage.html and run to show my page without the rounded corners
   2. Click New Style... and select the borderradius property
   3. Set it to 10px and look at the generated property.
   4. Copy in the snippet to add moz and webkit support and talk about vendor prefixes in CSS3
3. **CSS Pseudo Selectors**
   1. Open BlogPage.html and add :not(.odd) to the article element, and move the background style from the above declaration.
4. **@font-face**
   1. Add the font-face snippet
   2. Open page, then open F12 and show the font being retrieved
5. **[IF FAST] Media Queries Demo**
   1. Navigate to sweethatclub.org and show what happens when I resize the browser.
   2. Open sweethatclub.css and show how they are using media queries.

IE10 PP

1. **Show Flexbox Demo (CSS3 Flexible Layout demo)**
2. **Show CSS Gradient Background Maker**
3. **[IF FAST] Show the SVG Gradient Maker, and add an SVG Background to my Existing SVG Demos**

Web Sockets

1. **Show the chat demo for sockets from html5Labs (**[**http://html5labs.cloudapp.net/WebSockets/ChatDemo/wsdemo.html**](http://html5labs.cloudapp.net/WebSockets/ChatDemo/wsdemo.html)**).**
2. **[IF FAST] Show this running locally.**

FileAPI

1. **Open FileAPI.html and show loading the contents of a text file.**
2. **Show loading the contents of an image into a blob URL in memory then displaying that image on the page**
3. **[IF FAST] Show audio.html**
   1. Progresive Enhancement by using the File API to pull ID3 tags and show album metadata on a tooltip jQuery plugin