Demo 1 –What happens if browser does not understand a tag?

http://html5demo1.azurewebsites.net/Demo01-MyOwnElements.html

1) Created a page with a "new" HTML element called "`<SomeHTMLElementThatDoesNotActuallyExists>`"
2) I then added a style for this element that said to put it in italics
3) If I view this in any browser except older version of IE it actually WILL put the content in italics
4) If I put IE into "compatibility mode" the italics will disappear
5) To solve this, add JavaScript to add the element to the DOM

```
<script>
    document.createElement("SomeHTMLElementThatDoesNotActuallyExists");
</script>
```

Demo 2 – Show some of the new HTML 5 elements

http://html5demo1.azurewebsites.net/Demo02-UsingHTML5ShivOrChromeFrame.html

There are a lot of new HTML 5 elements… this page has a number of the new elements. As before, to get these to be recognized properly you would need to add the elements to the DOM… this page demonstrates to other approaches that may be considered

1) Using HTML5Shiv which takes care of all of the elements plus does some basic styling too (see https://github.com/aFarkas/html5shiv/)
2) Using Google Chrome Frame which is a plug-in to IE… which does WAY more… basically is Google Chrome with all of its "goodness" within IE. To accomplish this we add the following within the "<head>" section of the page

```
<meta http-equiv="X-UA-Compatible" content="chrome=1">
```

**Notice how the "compatibility mode" disappears in IE

Demo 3 – Transforming a page to HTML 5

http://html5demo1.azurewebsites.net/Demo03a-PagePreHTML5.html

and

http://html5demo1.azurewebsites.net/Demo03b-PageusingHTML5.html

1) Open up "03-PagePreHTML5.html" and review the contents. Looks like a "typical" sort of looking page" with "meaning" kind of coming from "id's"… which was one of the things the browser group saw… same "looking" pages over and over again
2) Open up "04-PageusingHTML5.html" and review the contents:
    a. Simplified DOCTYPE
    b. No xmlns anymore since, for good or for bad, no longer XHTML based (i.e. you will find elements with no closing tag character
    c. Meta tag for character set is simplified tremendously… and no "Content-Type" anymore
    d. The "meaningful" id's becomes actual HTML elements
    e. H"x" tags now can be grouped in something called an "hgroup"
    f. Since images are always self-closing, there is no need to have the ending "/" (although for me personally I am not crazy about this ☹)
    g. By convention, any links within the site should be wrapped in a "nav"
    h. Here I used the "aside" tag… but there is quite a bit of talk about when to use this one
    i. Finally, the copyright is now in an actual "footer" tag
    j. The stylesheet that corresponds with these had to have some adjustments made to it accordingly… so, instead of: "`#header {`" it becomes just "`header {`"

NOTE: it is important to know though that no FUNCTIONALITY actually gets put in when you do this… it is simply "semantics" to give better "meaning" to the page

Credit for this demo goes to http://www.webucator.com/self-paced-courses/course/comprehensive-introduction-html5.cfm

Demo 4 – New Input Types

http://html5demo1.azurewebsites.net/Demo04-NewInputTypes.html

1) Open up Demo04-NewInputType.html and review all of the input types
2) As per usual… open up in IE and put it into "compatibility mode" to view it in "old-IE"… oh wait, on my machine which is running IE 9 NONE OF THE NEW FORM TAGS WORK!
    a. Easy solution to this is, as before, "Google Chrome Frame"
    b. There are other options out there too, but… will leave this as an exercise for the curious
3) Open up in Chrome, Safari, Firefox, etc and see controls in action. Note that this is with no added JavaScript!

    Some notable things:

    a. Safari… "Search field" will have an "x" so can clear contents
    b. Implementations of "time" varies… Maxthon vs. Chrome for example
4) Now use a simulator to see how these behave on mobile devices. Probably best resource is http://www.browserstack.com which although not free, does have a trial that I am using for this purpose
    a. Notice how the keyboard actually changes when in Url field vs Email field
    b. Number uses a keypad
5) Try out the contenteditable areas too!

Demo 5 – Multimedia

http://html5demo1.azurewebsites.net/Demo05-Multimedia.html

1) Open up Demo05-Multimedia.html
2) First link brings you to Solvera's LipDub preview video in YouTube… nothing special here https://www.youtube.com/watch?v=JlxISJcsqxs
3) Second link navigates to YouTube with HTML 5 "switch" set to true https://www.youtube.com/watch?v=JlxISJcsqxs&html5=1
    NOTE: If use IE it may still work… because Google is using Google Chrome Frame!
4) Page also demonstrates… using the "HTML Gang Sign" video… the new <video> HTML 5 element

Demo 6 – Show off Azure

https://manage.windowsazure.com

1) Navigate to https://manage.windowsazure.com and login
2) Touch on what is available
3) Briefly show how can create a website
   a. Can pick from gallery
   b. Can get code from GitHub, Bitbucket, Codeplex
4) Show how one of the sites is set to publish whenever checkin to GitHub https://manage.windowsazure.com/#Workspaces/WebsiteExtension/Website/HTML5Demo1/quickstart
5) The other site's "publishing profile" is configured within my Visual Studio project so can deploy to it easily https://manage.windowsazure.com/#Workspace/WebsiteExtension/Website/SignalRDemo1/quickstart

Demo 7 – Opera Mobile Emulator and importance of ViewPort

http://www.opera.com/developer/tools/mobile/

1) Open up the installed Mobile Emulator
2) Navigate to local page (currently http://localhost:60602/HTML5Demo1/Demo01-MyOwnElements.html) with and without the top "viewport" meta tag

Demo 8 – SignalR

http://www.asp.net/signalr/

1) Start-up demo site on Azure (if not already started since I am not crazy about the concept of an anonymous application)
2) Navigate to one or both of:
   a. Hit
   b. Chat
3) Go to the site in many different browsers… or tabs within a single browser