

Intro to HTML5 for the RPG Developer

Brian May
IBM i Modernization Specialist
Profound Logic Software

Webmaster and Coordinator
Young i Professionals



Overview

- History HTML
- What is HTML5?
- New Features
- Features Removed
- Resources

HTML

A look back as how we got here

History of HTML

- HTML
 - HyperText Markup Language
 - Markup language understood universally by all web browsers to render text, images, and other media
 - First draft of HTML standard in 1993

HTML



History of HTML

- Nov. 1995 – HTML 2.0 published by Internet Engineering Task Force (IETF)
- Jan. 1997 – HTML 3.2 published by World Wide Web Consortium (W3C)
- Dec. 1997 – HTML 4.0 published by W3C
- Dec. 1999 – HTML 4.01 published by W3C



History of HTML

- HTML 4.01 is still the official HTML standard
 - 13 years old now
 - W3C spent early 2000's concentrating on XHTML standards
 - Web technology has continued to progress even though the HTML standard did not
 - Increased Javascript use
 - Browser Plug-in technologies
 - Nonstandard implementations of these new functionalities have led to current applications being complicated by extra code for browser compatibility

What is HTML5?

What is HTML5?

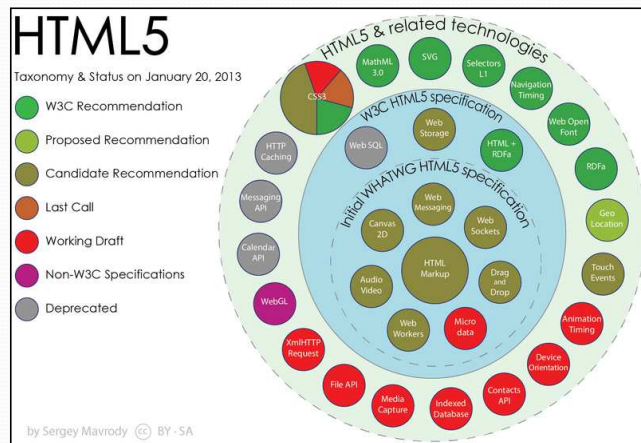
- Latest revision to the HTML standard
- Web Hypertext Application technology Working Group (WHATWG)
- Initially rejected but later adopted by the W3C
- Supersedes HTML4.01, XHTML1 and DOM Level 2 HTML
- Currently is a W3C Candidate Recommendation
 - Not finalized

What is HTML5?

- Because of W3C's concentration on XHTML standard, HTML5 is partially an attempt to "catch up" with techniques already widely used
- Standardization of API's used for modern web development
- More concentration on Web Applications instead of static content
- More semantic markup

What is HTML5?

- There are many different technologies surrounding HTML5
- Obviously the lines between these technologies are easily blurred



What is HTML5?

- W3C Spec
 - <http://www.w3c.org/TR/html5/>
- WHATWG Spec
 - <http://www.whatwg.org/html5/>

What's New in HTML5

What's New in HTML5

- Simplified DOCTYPE and Character Set
 - In HTML5, there is only one DOCTYPE
 - `<!DOCTYPE HTML>`
 - Previously DOCTYPEs were very complex
 - `<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">`
 - HTML is no longer a subset of SGML
 - Setting the Character Set
 - `<meta http-equiv="Content-Type" content="text/html; charset=utf-8">`
 - `<meta charset="UTF-8">`

What's New in HTML5

- Semantic Elements
 - `<section>`
 - Used to denote a generic section in an application or document.
 - Generally something with a heading.
 - `<nav>`
 - Used to mark the section of a page that contains navigational controls
 - `<article>`
 - Denotes an independently distributable composition within a site. Good examples would be a blog post, forum post, comment, or an article from an e-zine.

What's New in HTML5

- Semantic Elements
 - `<aside>`
 - Used to mark content that is tangentially related to the main content. Think of this like content in a sidebar.
 - `<hgroup>`
 - Used to mark the heading of a section. This is can contain multiple heading lines (subheadings) using the `<h1>` through `<h6>` elements.
 - `<header>`
 - Defines the header of a section. Usually contains an `<hgroup>` or `<h1>` element.

What's New in HTML5

- Semantic Elements
 - `<footer>`
 - Denotes footer for current section or document. A `<footer>` would normally contain things like the author of an article or the date posted.
 - `<time>`
 - Indicates to the browser that the contents represent a time value.
 - `<mark>`
 - Used to enclose part of text as marked or highlighted.

What's New in HTML5

- Why do we need all these new semantic tags?
 - HTML4 only allowed 6 levels of headings in a page (h1 – h6).
 - Semantic elements can each have their own set of 6 heading levels thus making well structured content much more flexible.

What's New in HTML5

- Example of Semantic Content

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>Semantic Example</title>
</head>
<body>
  <header>
    <hgroup>
      <h1>Example Blog Output</h1>
      <h2>This is obviously not a real working blog!</h2>
    </hgroup>
    <nav>
      <ul>
        <li><a href="test.html">home</a>
        <li><a href="test.html">about</a>
      </ul>
    </nav>
  </header>
  <article>
    <header>
      <time datetime="2013-07-14" pubdate>July 14, 2013</time>
      <h1>
        <a href="test.html" rel="bookmark" title="link to post">My First Blog Entry</a>
      </h1>
    </header>
    <p>This is the content for my first blog post. Ain't it cool?</p>
  </article>
  <article>
    <header>
      <time datetime="2013-07-16" pubdate>July 14, 2013</time>
      <h1>
        <a href="test.html" rel="bookmark" title="link to post">My Next Blog Entry</a>
      </h1>
    </header>
    <p>This is the content for another blog post.</p>
  </article>
  <footer>
    <p>© 2013 <a href="mailto:bmay@profoundlogic.com">Brian May</a>
  </p>
</body>
</html>

```

What's New in HTML5

- Canvas
 - Provides scripts with a resolution-dependent bitmap canvas, which can be used for rendering graphs, game graphics, art, or other visual images on the fly.
 - Javascript is used to draw on the canvas.
 - Useful for rendering dynamic visual content such as charts

What's New in HTML5

- Canvas

```
<!DOCTYPE html>
<html>
<body>

<canvas id="exampleCanvas" width="200" height="200" style="border:1px solid;">
Sorry, your browser does not support the HTML5 canvas.</canvas>

<script>
var canvas=document.getElementById("exampleCanvas");
var context=canvas.getContext("2d");
context.beginPath();
context.arc(95,50,40,0,2*Math.PI);
context.stroke();
</script>

</body>
</html>
```

What's New in HTML5

- Video
 - <video> tag in HTML5 allows video content to be embedded into a webpage without needing any third party plugins (Flash, QuickTime, etc.)
 - While all of the modern browsers support the <video> tag (IE9+, Firefox, Chrome, Safari, Android), they do not all support the same formats (codecs).
 - The <video> tag allows you to list multiple files in different formats and the browser can pick one it supports.

What's New in HTML5

- Video

```
<!DOCTYPE html>
<html>
<body>

<video width="320" height="240" controls>
  <source src="http://www.w3schools.com/html/movie.mp4" type="video/mp4">
  <source src="http://www.w3schools.com/html/movie.ogg" type="video/ogg">
  Your browser does not support the video tag.
</video>

</body>
</html>
```

What's New in HTML5

- Local Storage
 - Until now, if you wanted to save information on the user's local device, you would use a Cookie
 - Cookies are transmitted with every HTTP request
 - Limited to ~4KB of data
 - In the Web Storage specification (previously a part of HTML5), local storage is simple and effective
 - Implemented in Key/Value pairs
 - Never transmitted back to the server
 - Implemented natively in browser
 - 5MB of storage allowed per origin

What's New in HTML5

- Local Storage
 - http://www.w3schools.com/html/tryit.asp?filename=try_html5_webstorage_local_clickcount

What's New in HTML5

- Form Enhancements
 - Support for new input types vary by browser
 - Unsupported types are treated as normal input fields
 - `<input type="color" name="myColor">`
 - Allows selection of color using a "Color Picker"
 - `<input type="date" name="myDate">`
 - Renders a date/calendar control
 - Validates date

What's New in HTML5

- Form Enhancements
 - `<input type="email" name="myAddr">`
 - Textbox contents automatically validated for proper format.
 - Mobile devices alter keyboard
 - `<input type="url" name="myPage">`
 - Textbox contents automatically validate for proper format.
 - Mobile devices alter keyboard
 - `<input type="number" name="myNum">`
 - Desktop browsers render as "spinners"
 - Mobile devices alter keyboards
 - Input validated before submit

What's New in HTML5

- Form Enhancements
 - Required Fields
 - `<input type="email" name="myAddr" required>`
 - Placeholder Text
 - Displayed inside the input field when empty.
 - Especially useful on mobile devices to save screen real estate.
 - Data Lists
 - New `<datalist>` element defines a list of values that can be attached to an input box for “autocomplete” functionality

What's New in HTML5

- HTML5 Form Example

```

<!DOCTYPE html>
<html>
  <body>
    <form>
      <datalist id="mySessions">
        <option value="Data Structures">
        <option value="XMLSERVICE">
        <option value="HTML5">
      </datalist>

      <input name="myClass" list="mySessions" required placeholder="Session Name">
      Date:<input type="date" name="classDate">
      Attendees:<input type="number" name="attendees" min="0" max="25">
      <input type="submit">
    </form>
  </body>
</html>

```

What's New in HTML5

- Other New Features
 - Geolocation
 - AppCaching (Offline web applications)
 - Web Workers
 - SVG

What's Removed in HTML5?

What's Removed in HTML5?

- Frames
 - While they have been widely discouraged in modern web design, HTML5 has actually dropped support for frames/framesets
 - `<iframe>` is still supported and has even been enhanced in HTML5
- Styling Related Tags
 - `<basefont>`, `<big>`, `<center>`, ``, `<strike>`
 - These should be replaced by CSS

What's Removed in HTML5

- `<applet>`
 - Replaced by the `<object>` tag

Introduction to CSS

What is CSS?

- CSS stands for Cascading Style Sheets. Styles are rules applied to HTML
- Standard maintained by World Wide Web Consortium (W3C)
- CSS eases coding and maintenance by removing the formatting from the HTML code

CSS Selectors

- Used to identify HTML elements to apply styles to.
 - By HTML tag name
 - By HTML tag id
 - By class name

```
input {  
}
```

- Style rules go inside braces

Tag name selector

- Applies to any HTML tag of given type, for example:

```
input {  
  
}
```

- Applies to all:

```
<input>
```

Tag id selector

- Applies to any HTML tag with given id, for example:

```
#main {  
  
}
```

- Applies to:

```
<div id="main">
```

Class selector

- Applies to any HTML tag with given class, for example:

```
.error-message {  
  
}
```

- Applies to:

```
<div class="error-message">
```

CSS style rules

- Coded inside selectors:

```
div.error-message {  
  
  color: red;  
  text-decoration: underline;  
  
}
```

- Available values defined by W₃C standard

Learn more about CSS

- W₃C CSS Tutorial:

<http://www.w3schools.com/css/>

- W₃C CSS Reference:

<http://www.w3schools.com/cssref/default.asp>

Introduction to JavaScript

What is JavaScript

- JavaScript is the browser's scripting language for client side programming
- JavaScript can respond to events like "mouse moved" or "key pressed" to make the applications more interactive
- JavaScript code runs *after* all webserver code has executed and the screen is displayed.

Why javascript

- JavaScript is popular. There are many resources available to help you learn it. The Web is full of JavaScript examples, code snippets, and tutorials. You can view other web site's JavaScript code.
- JavaScript is understood by any standard browser, and requires no special software to run.

Javascript vs java

- JavaScript is similar to Java in name only. The syntax looks similar, but the two are completely different languages and have different purposes.
 - JavaScript is strictly a client-side language.
 - JavaScript is interpreted – not compiled.
 - JavaScript runs “natively” in a Web browser, where Java runs in a special piece of software called a ‘JVM’.

How it works

- JavaScript is “loosely typed”. Variables can contain any data type, and can also hold references to objects. The “data type” of a variable is what it currently holds. This can be changed at any time. Due to this, the programmer has to be careful of what operations to perform on the data. There is no compiler to catch improper variable usage – this always results in a run-time error.
- JavaScript is object-oriented. Coding revolves around manipulating objects, which are usually HTML elements on the page.

Code organization

- JavaScript code is organized into functions, which are similar in concept to ILE RPG sub-procedures. The syntax is as follows:
 - `function helloworld(parms) { }`
- JavaScript code can be placed into the HTML itself.
- Code can also be placed into external files.
- The preferred method is to use external files. This allows for better code organization, and also for the modification of the JavaScript code without having to adjust multiple pages.

Properties and methods

- JavaScript objects have “properties” that you’ll use to retrieve the values of HTML elements.
- JavaScript objects have functions that you can use to retrieve or set HTML element attributes. These functions are called “methods” in JavaScript.
- Properties can themselves be objects with their own properties and methods.

Additional Resources

Additional Resources

- General HTML5 learning
 - W3Schools – <http://www.w3schools.com>
 - Dive Into HTML5 – <http://www.diveintohtml5.info>
- Browser Support Reference
 - Can I Use... – <http://www.caniuse.com>
- Browser Feature Detection
 - Modernizr – <http://www.modernizr.com>
- HTML5 Specs
 - W3C Spec – <http://www.w3c.org/TR/html5/>
 - WHATWG Spec – <http://www.whatwg.org/html5/>

Questions?

Thank You!

About the Presenter

Brian May is an IBM i Modernization Specialist for Profound Logic Software. He is also webmaster and coordinator for the Young i Professionals (<http://www.youngiprofessionals.com>). He is a husband and father of two beautiful girls. Brian can be reached at bmay@profoundlogic.com.

