HTML5 – are we there yet?

Webpage Design


3
RICH MEDIA
THE HISTORY OF THE WEB is punctuated with technological improvements. One of the earliest additions to HTML was the img element, which fundamentally altered the web. Then, the introduction of JavaScript allowed the web to become a more dynamic environment. Later, the proliferation of Ajax made the web a viable option for full-fledged applications.
Web standards have advanced so much that it's now possible to build almost anything using HTML, CSS, and JavaScript—almost anything.
There are some gaps in the web standards palette. If you want to publish text and images, HTML and CSS are all you need. But if you want to publish audio or video, you'll need to use a plug-in technology such as Flash or Silverlight.
"Plug-in" is an accurate term for these technologies—they
33 HTML5 FOR WEB DESIGNERS
CANVAS
1
A BRIEF HISTORY OF MARKUP
HTML IS THE UNIFYING LANGUAGE of the World Wide Web. Using just the simple tags it contains, the human race has created an astoundingly diverse network of hyperlinked documents, from Amazon, eBay, and Wikipedia, to personal blogs and websites dedicated to cats that look like Hitler.
HTML5 is the latest iteration of this lingua franca. While it is the
A BOOK APART
Brief books for people who make websites
No. 1
Jeremy Keith
HTML5 FOR WEB DESIGNERS
FOREWORD BY Jeffrey Zeldman
kindle

Overview

- Historical context
- DOCTYPEs and such
- Can/should I use HTML5 now?
  - The problem with browsers...
- Syntax
- Semantics
  - The new (structural) elements
  - HTML5 content model
- Audio, Video and Canvas
- HTML5 APIs

HTML5 is the first markup language to have its own official logo!

XML is the future of the Web

- Somewhere along the way, W3C lost sight of the purpose of web markup when they proposed the radical XHTML 2 – an attempt to move the web to XML.
- It envisaged a new web built with new tools – and it was not backwards compatible with existing web content.
- It's as if they were mainly interested in the markup language itself and not with those who were going to use it (browser-makers and front-end developers).
- Things came to a head in 2004 when Ian Hickson proposed an extension of HTML which was rejected by W3C.
- Hickson and other rebels formed the Web Hypertext Application Technology Working Group (WHATWG)

OK, we got it wrong!

- WHATWG wanted a markup language that could be used to build web applications and not just webpages.
- They began developing Web Forms 2.0 and Web Apps 1.0, which merged to become HTML5.
- In 2006, W3C admitted that their approach just wasn’t working.
- Shortly afterwards, W3C issued a new charter for an HTML Working Group.
- They decided to use the work begun by WHATWG as the basis for the new specification.

HTML5 here we come!

- For a while, W3C were developing two different and totally incompatible markup languages.
- But in 2009, they announced the end of XHTML2 development.
- Once the ambiguity was removed, web designers pounced on HTML5 as the “new kid on the block”.
- Both WHATWG and W3C continue to work on HTML5 but they are working together and progress has been faster than expected.

HTML5 development principles

- Partly as a reaction against the idealist nature of XHTML 2 development, HTML5 is a very pragmatic markup language.
- Rather than attempting to impose a new order on the web, it looks at what people are already doing and tries to codify it. The introduction of the <header> and <footer> elements are a good example of this.
- HTML5 development principles:
  - Support existing content
  - Do not reinvent the wheel
  - Pave the cowpaths

HTML Design Principles by W3C

DOCTYPES

The HTML 4.01 DOCTYPE

<!DOCTYPE HTML PUBLIC » "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">

The XHTML 1.0 Strict DOCTYPE

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict //EN "
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

The HTML5 DOCTYPE

<!DOCTYPE html>

"...the doctype for HTML5 is very pragmatic. Because HTML5 needs to support existing content, the doc-type could be applied to an existing HTML 4.01 or XHTML 1.0 document. Any future versions of HTML will also need to support the existing content in HTML5, so the very concept of applying version numbers to markup documents is flawed."

Jeremy Keith *HTML5 for Web Designers*

Content Type

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8“ />

<meta charset="UTF-8">

<link rel="stylesheet" type="text/css" href="file.css“ />

<link rel="stylesheet" href="file.css">

<script type="text/javascript" src="file.js"></script>

<script src="file.js"></script>

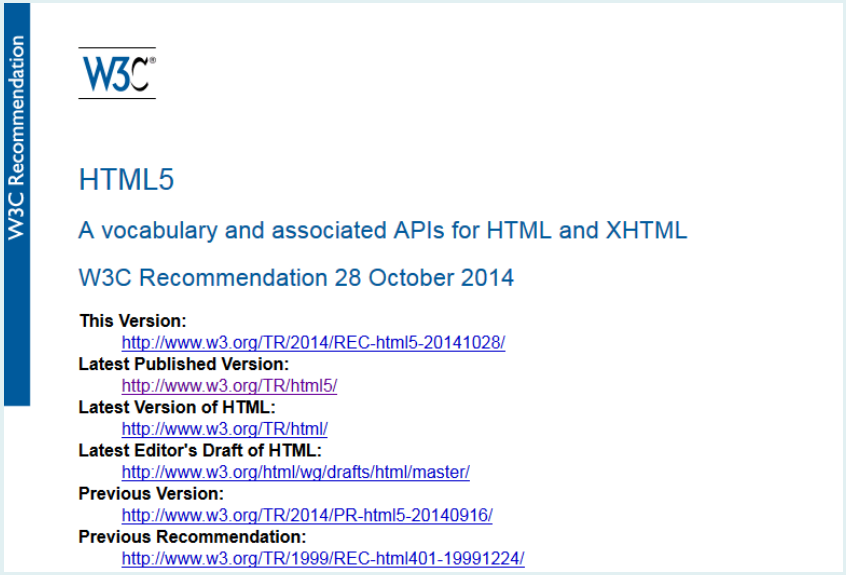
HTML5 aims to simplify markup, requiring only that information that is essential for the browser. For example, there is no need to specify a “type” attribute for the script tag because javascript is the **only** script type used.

Can/should I use HTML5 now?

- There is no imperative to use HTML5 now. XHTML 1.0 works just fine and is supported in all browsers – even older ones.
- HTML5 is still new. Some browsers do not support the new elements. Some of the concepts are “unresolved” – there is plenty of discussion in the web design community.
- HTML5 is not “finished”, it’s still a work in progress and may change. Although the complete definition was recently announced, 2014 is the year in which full adoption is predicted and it’s unlikely to be “recommended” until 2022…

Stop Press!

The W3C changed the status of the HTML5 specification to “Recommended” on 28th October 2014.



HTML5 by W3C

I want it now!

- There are plenty of reasons why you may want to use HTML5 now.
- It's sexy.
- Freedom from proprietary technologies (almost).
- Better functionality.
- Future-proofing websites.

```
<!--[if lt IE 9]>
<script src="//html5shiv.googlecode.com/svn/trunk/html5.js"></script>
<![endif]-->
```

The key difficulty in implementing HTML5 now is that older versions of Internet Explorer (those before IE9) do not understand the new HTML5 elements. The workaround is to use JavaScript to tell IE how to interpret them. This is best done using Remy Sharp's lovely html5shiv, now hosted at Google.

Syntax

Strict XHTML 1.0 syntax

```
<h2>Students</h2>
<p>Typically, our students fall into three groups...</p>
<p>See <a href="/our-students/">Our students</a> for more information.</p>
<img src="/our-students/graduation.jpg" width="552" height="413"
alt="Graduation" />
```

Valid HTML5 syntax

```
<H2>Students</H2>
<p>Typically, our students fall into three groups...</P>
<p>See <A Href="/our-students/">Our students</a> for more information.</p>
<img src="/our-students/graduation.jpg" width=552 height="413"
ALT="Graduation">
```

Mixed upper and lower case tags and attributes, quoted and unquoted values, self-closing elements unclosed...

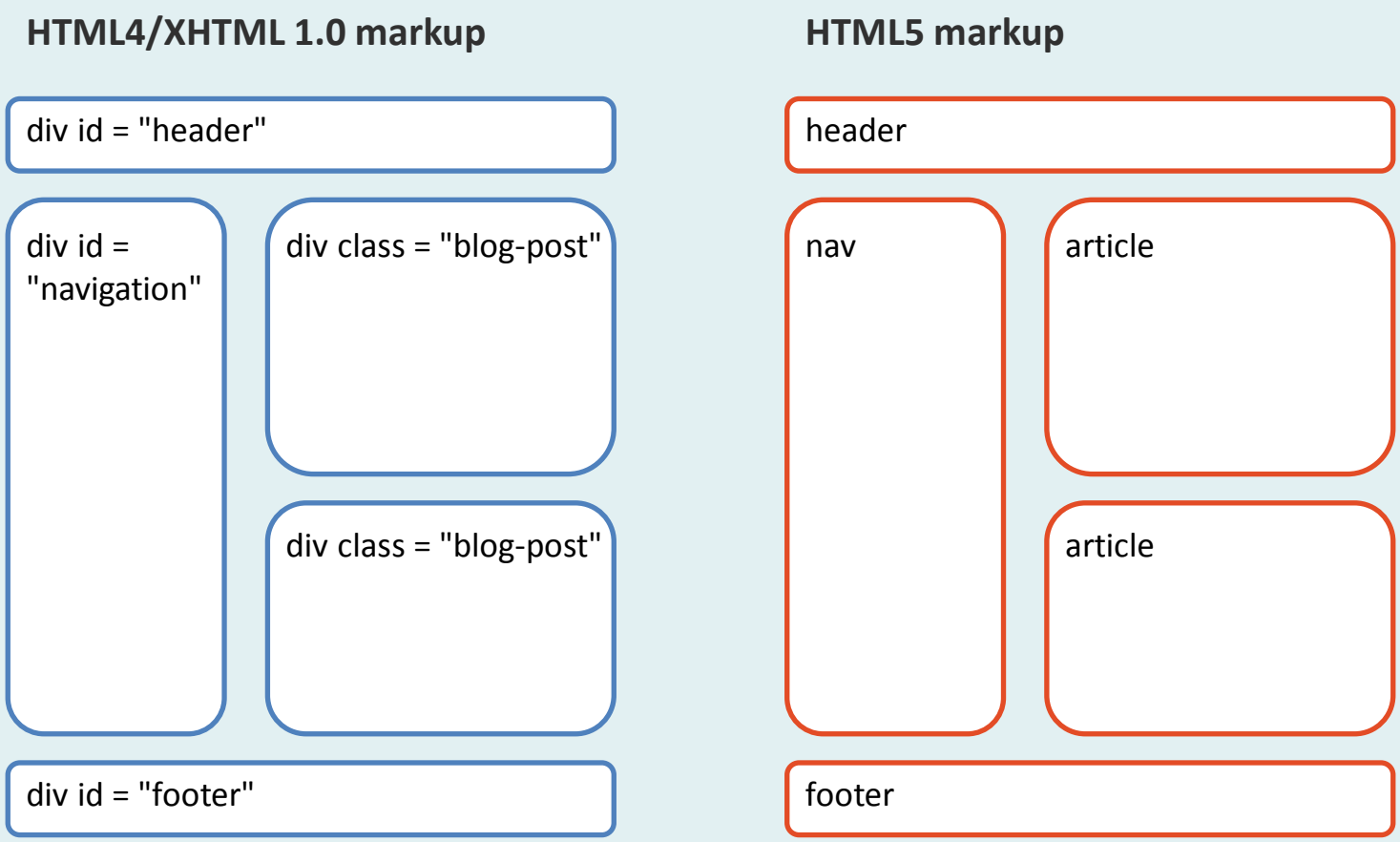
...HTML5 has a pretty liberal syntax, however, we should still observe good coding practice. The strict syntax (top) also validates as HTML5. Most front-end developers agree that maintaining XHTMLs strict syntax in HTML5 is a good idea.

New Structural Elements

<main>	Defines the main content on a page
<article>	Defines an article
<aside>	Defines contents aside from the page content
<embed>	Defines external interactive content or plugin
<figcaption>	Defines the caption of a figure element
<figure>	Defines a group of media content, and caption
<footer>	Defines a footer for a section or page
<header>	Defines a header for a section or page
<nav>	Defines navigation links
<section>	Defines a section
<wbr>	Defines a word break opportunity (long strings)

New Structural Elements in HTML5 by Chris Mills & Bruce Lawson

Building web pages with HTML5



HTML5 has been designed to give web designers the tags they need to markup common page elements without needing to use id and class names.

Built-in Obsolescence

HTML5 aims to be backward compatible. Therefore there are no *deprecated* elements, only *obsolete* elements. This means that you can use obsolete elements, which your browser will still render, however your web page will be "non-conforming" due to the use of these.

Below are a list of elements that are obsolete:
<frame>
<frameset>
<noframes>
<acronym> (use <abbr> instead)

Presentational elements such as:
<font>
<big>

<strike>
are also obsolete.

The HTML Elements

A
- <a>
- <abbr>
- <acronym>
- <address>
- <applet>
- <area>
- <article>
- <aside>
- <audio>

B
- <b>
- <base>
- <basefont>
- <bdi>
- <bdo>
- <bgsound>
- <big>
- <blink>
- <blockquote>
- <body>
-
- <button>

C
- <canvas>
- <caption>
-
- <cite>
- <code>
- <col>
- <colgroup>
- <command>

D
- <data>
- <datalist>
- <dd>
- <del>
- <details>
- <dfn>
- <dir>
-
- <dl>
- <dt>

E
-
- <embed>

F
- <fieldset>
- <figcaption>
- <figure>
- <font>
- <footer>
- <form>
- <frame>
- <frameset>

G H
- <h1>
- <h2>
- <h3>
- <h4>
- <h5>
- <h6>
- <head>
- <header>
- <hgroup>
- <hr>
- <html>

I
- <i>
- <iframe>
- <img>
- <input>
- <ins>
- <isindex>

J K
- <kbd>
- <keygen>

L
- <label>
- <legend>
-
- <link>
- <listing>

M
- <main>
- <map>
- <mark>
- <marquee>
- <menu>
- <meta>
- <meter>

N
- <nav>
- <nobr>
- <noframes>
- <noscript>

O
- <object>
-
- <optgroup>
- <option>
- <output>

P
- <p>
- <param>
- <plaintext>
- <pre>
- <progress>

Q
- <q>

R
- <rp>
- <rt>
- <ruby>

S
- <s>
- <samp>
- <script>
- <section>
- <select>
- <small>
- <source>
- <spacer>
-
- <strike>
-
- <style>
- <sub>
- <summary>
- <sup>

T
- <table>
- <tbody>
- <td>
- <textarea>
- <tfoot>
- <th>
- <thead>
- <time>
- <title>
- <tr>
- <track>
- <tt>

U
- <u>
-

V
- <var>
- <video>

W
- <wbr>

X Y Z
- <xmp>

The symbol indicates that the element has been added in HTML5. Note that other elements listed here may have been modified or extended by the HTML5 specification. Dimmed elements are non-standard, obsolete, or deprecated; they must not be used in new Web sites, and should gradually be removed from existing ones.

[HTML Elements](#) by Mozilla Developer Network

Semantic Shift

Some previously presentational elements have not been made obsolete but have gone through a semantic shift so that their meaning has changed.

<small> – no longer means “render this at a small size.” It now means “this is the small print” for legals, T’s & C’s or privacy agreements.
<b> – previous meaning “render in bold.” New meaning “to be visually different from the normal text, without conveying extra importance.”
<i> – previous meaning “italicize.” New meaning “in an alternate voice or mood”.

These changes are largely made to keep the specification and elements relevant to non visual user agents such as screen readers.

‘Bold’ and ‘Italic’ are words that only make sense visually. The idea of developing towards non visual user agents encourages designers to think beyond basic visual design and development.

HTML5 obsolete and deprecated elements by Paul Wallas

Anchors Aweigh!

One element in the HTML5 specification that has almost been re-invented.

The <a> (anchor) element has always been an inline element meaning multiple <a> elements were needed for hyperlinks that span multiple elements.

In HTML5 multiple elements can be wrapped into a single <a> element. For example:

```
<a href="index.html">
   <h1>Welcome to my site</h1>
   <p>Paragraph</p>
</a>
```

"Block-level" links in HTML5 by Bruce Lawson

Document Structure

XHTML 1.0 markup for a blog

```
<div class="header">
  <h1>My wonderful blog</h1>
</div>
<div class="blog_articles">
  <div class="article">
    <h2>The Zen of HTML5</h2>
    <p>HTML5 is an evolution of HTML4...</p>
    <p class="author">by David Watson</p>
  </div> <!-- /article -->
  <div class="article">
...
  </div> <!-- /article -->
</div> <!-- /blog_articles -->
```

The HTML5 document structure aims to describe the content structure rather than the page structure because content can be portable (syndicated) and is not always defined within the context of a "page".

HTML5 markup for a blog

```
<header>
  <h1>My wonderful blog</h1>
</header>
<section>
  <article>
    <header>
      <h1>The Zen of HTML5</h1>
    </header>
    <p>HTML5 is an evolution of HTML4...</p>
    <footer>
      <p class="author">by David Watson</p>
    </footer>
  </article>
  <article>
...
  </article>
</section>
```

Document Outlines

HTML5 introduces a new concept to markup: **sectioning**. The new structural elements such as <section> and <article> are not just semantically rich versions of , they are used to provide structure and hierarchy to a document. One nested within another gives no meaning and conveys no hierarchy, whereas, an <article> nested within a <section> is considered subordinate to its parent as well as providing semantic information.

```
<h1>HTML5 is weird</h1>
<div>
    <h1>XHTML has its problems</h1>
</div>
```

```
<h1>HTML5 is weird</h1>
<article>
    <h1>XHTML has its problems</h1>
</article>
```

If we were to describe the document outline described by the markup above, it would look like this:

- HTML5 is weird
  - XHTML has its problems

- HTML5 is weird
- XHTML has its problems

...where indentation is used to describe hierarchy in the document outline.

Element Hierarchy

In order to make sense of sectioning and to understand why it's possible to have many <h1> headings in a HTML5 document, whereas in XHTML it was best practice to have only one, it's important to realise that the nature of headings has changed in HTML5. In XHTML, <h2> is subordinate to <h1> but in HTML5, one <h1> can be subordinate to another, depending on context.

```
<h1>HTML5 is weird</h1>
<div>
    <h2>XHTML has its problems</h2>
</div>
```

```
<h1>HTML5 is weird</h1>
<article>
    <h1>XHTML has its problems</h1>
</article>
```

In the markup above, we can convey the hierarchy of headings by cascading through the descending levels of heading. On the left, we must use <h2> because conveys no hierarchy to its children. Whereas, the children of <article> are considered to be subordinate to elements at the same level as <article>. In this case, the second <h1> is subordinate to the first. So, in both cases, the document outline looks like this:

HTML5 is weird

XHTML has its problems

Logical Headings

The easiest way to think about this is to consider headings in XHTML to be *absolute*, whereas headings in HTML5 are *relative*. So, their calculated or logical value is dependent upon their context. Consider the markup below:

Markup heading value	Logical heading value
HTML5 is weird	**<h1>**
XHTML has its problems	**<h1>**
Markup languages...	
Problem number 1	**<h2>**
HTML5 is weird	**<h1>**
<article>	
XHTML has its problems	**<h2>**
Markup languages...	
Problem number 1	**<h3>**
</article>	

Notice that when <article> is used, the logical heading value is not the same as the markup heading value because headings in HTML5 are relative and their value may vary depending on where they are in the document hierarchy.

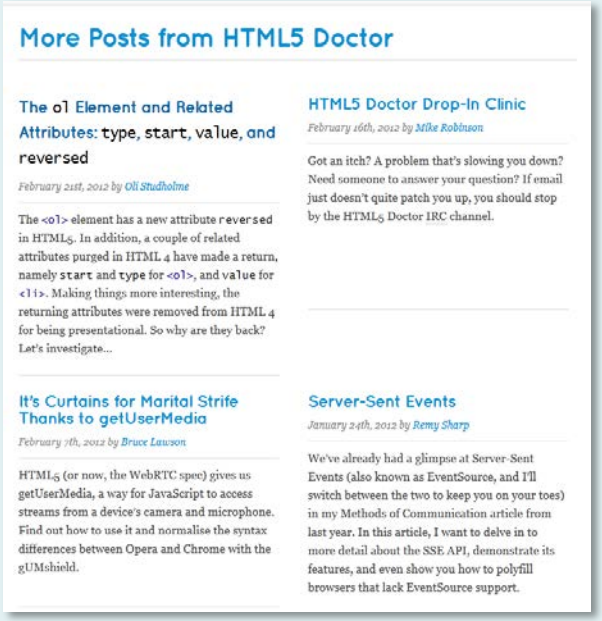
Why so complicated?

The reason for this apparently bizarre implementation is actually quite simple.

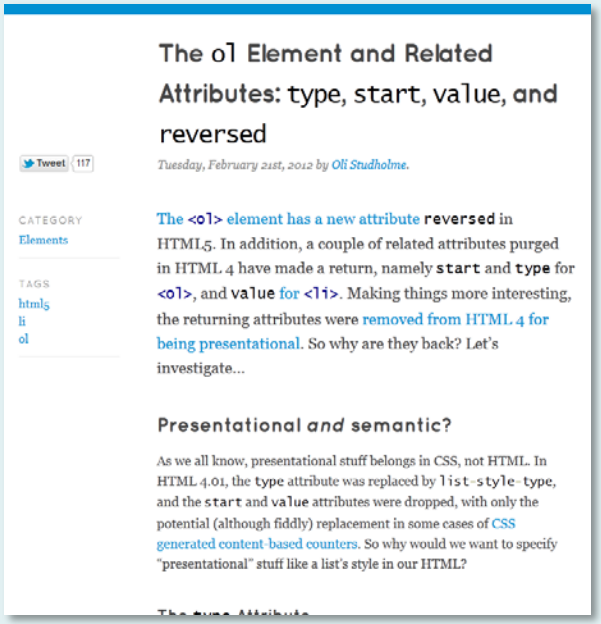
When XHTML was first codified, most websites were static and the context of any document element was obvious. Everything related to the webpage <body> and every element below (or within) that element was fixed in terms of its hierarchy.

As more and more websites became dynamic, pulling content from databases and populating templates, it became difficult to be specific about heading levels because one couldn't say where any particular chunk of content might end up.

A blog is a very good example of what can go wrong:



In this context the article title is <h2>...



but in this context the article title is <h1>

So, how should the heading be marked up in the databse? Well, in HTML5, it's easy, we mark it up as <h1> because its logical heading level will be calculated depending on its context.

Audio and Video

- For the first time, HTML5 provides a way to play audio and video in a browser without the use of plugins like Flash.
- Unfortunately, there is still a problem with media formats.
- The most popular formats like MP3, are not open and browser makers must pay to support them.
- So, although the “big boys” can afford to pay for MP3 support in their browsers, Mozilla cannot – Firefox does not support MP3 but it does support the open Ogg Vorbis format.
- Unfortunately, not all of the other browsers support the .ogg format.
- The result is that we must provide different formats to different browsers.

Come together (please!)

An example from *HTML5 for Web Designers* by Jeremy Keith

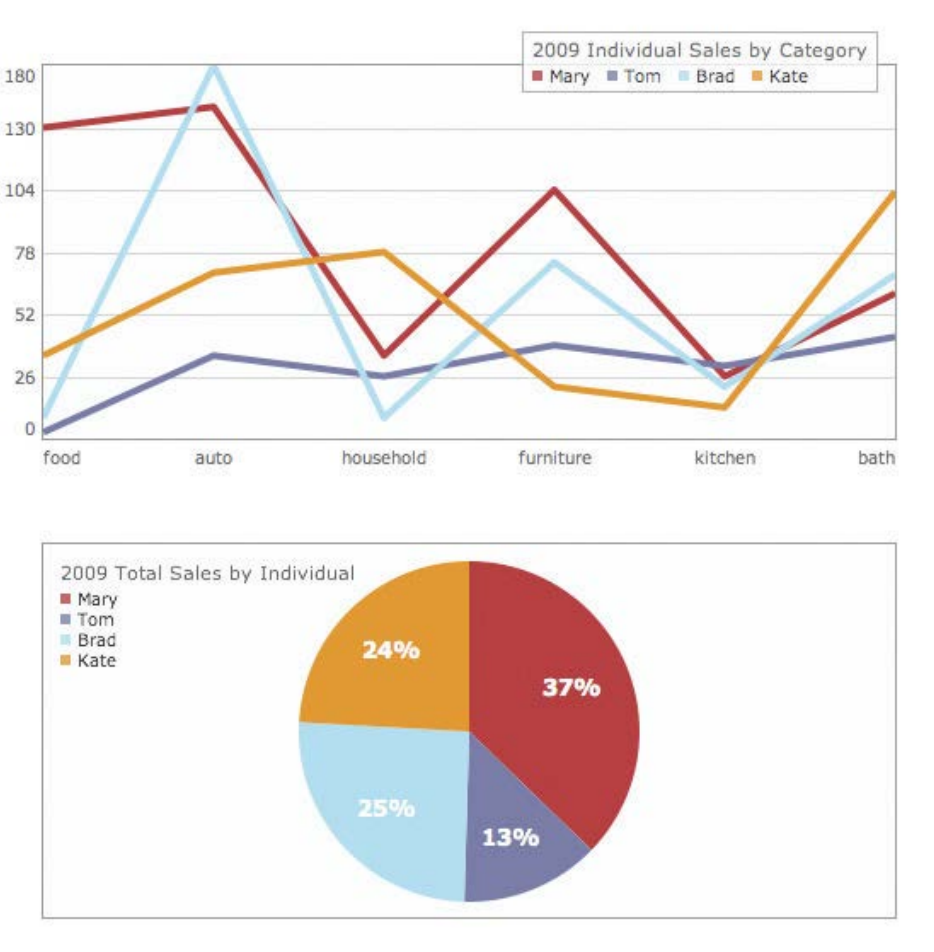
```
<audio controls>
   <source src="witchitalineman.ogg" type="audio/ogg">
   <source src="witchitalineman.mp3" type="audio/mpeg">
   <object type="application/x-shockwave-flash"
data="player.swf?soundFile=witchitalineman.mp3">
      <param name="movie"
value="player.swf?soundFile=witchitalineman.mp3">
      <a href="witchitalineman.mp3">Download the song</a>
   </object>
</audio>
```

This example has four levels of graceful degradation:

The browser supports the audio element and the Ogg Vorbis format.
The browser supports the audio element and the MP3 format.
The browser doesn't support the audio element but does have the Flash plug-in installed.
The browser doesn't support the audio element and doesn't have the Flash plug-in installed.

Canvas

- Canvas is a new element that can be used to create images dynamically.
- For example, data could be pulled from a database and <canvas> could be used to draw a graph of that data using an API.



APIs, the big deal

- If HTML5 were just another flavour of markup, it wouldn't be such a big deal.
- What makes HTML5 different are the new Application Programming Interfaces (APIs).
- The APIs enable HTML5 to work with JavaScript and empower developers to turn web pages into web applications.
- There are many APIs…

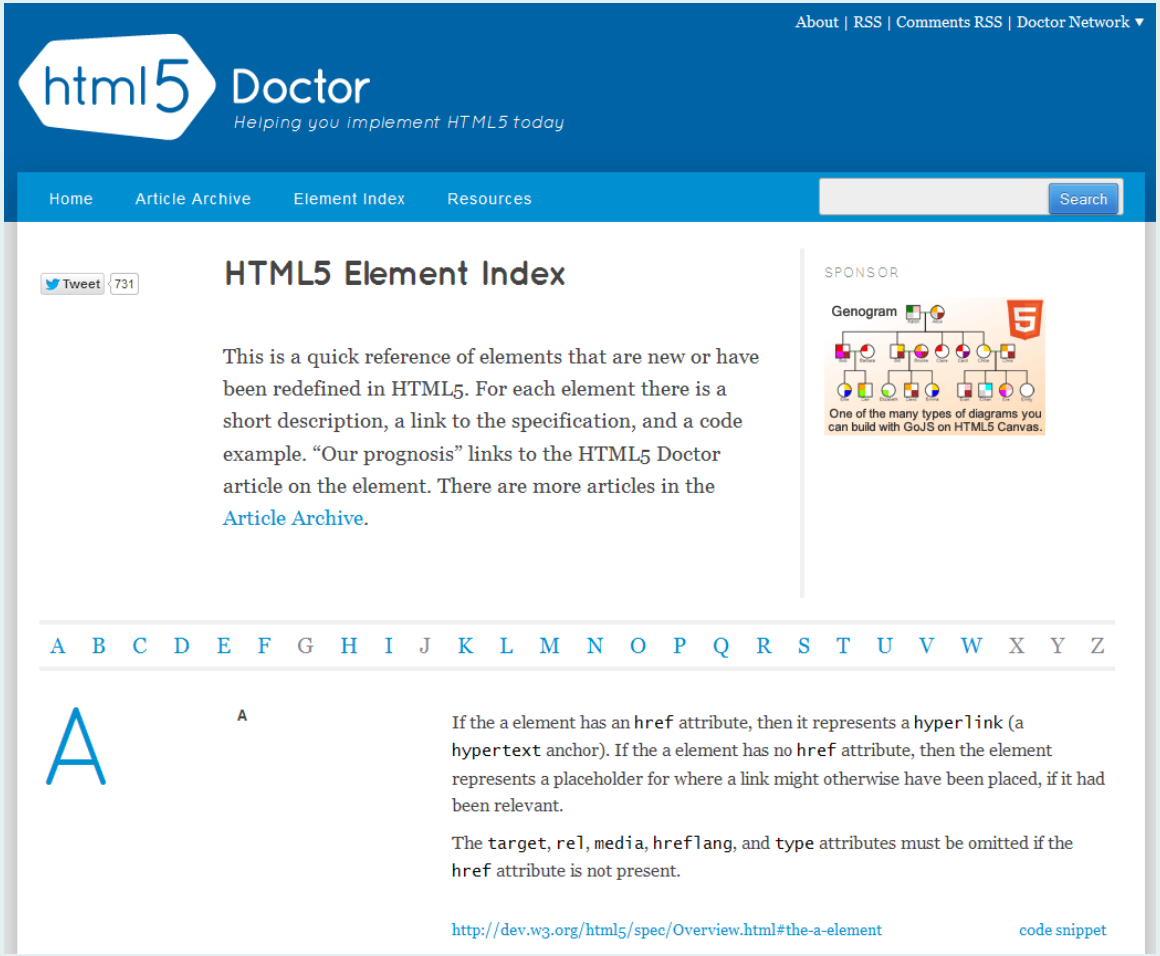
The HTML5 APIs

- …but some of the better known include:
- Media*: includes video, audio and controls.
- Drag and Drop*: allows native drag and drop functionality.
- Offline Web Applications/Application Cache*: allows apps to be used offline.
- Web Storage*: allows client-side data to be saved.
- Canvas 2D Context*: allows drawings to be made programmatically.

* W3C Specification
* WHATWG Specification

[The developer's guide to the HTML5 APIs] by Rich Clark

Learn more



HTML5 Doctor by Rich Clark, Steve Faulkner, Bruce Lawson, Remy Sharp, Oli Studholm and Ian Devlin

I've seen the

FUTURE

It's in my

BROWSER