Intro to HTML(5)

Prerequisites

- Ruby Web Servers

What you'll learn

- W3C & Specifications
- (Short) History of HTML
- Encoding HTML
- Semantic HTML
- HTML Elements, Attributes, and Values
- HTML Nesting
- HTML5

*

What you won’t learn

Coming Soon:

- CSS3
- Javascript

Intro to Internet Recap

- What is HTTP?
- Which HTTP method is used by default to retrieve a document from the web server?
- What part of the response contains HTML?

See Intro to Internet slides

Codepen.io

For quick HTML demoing, we'll use www.codepen.io

What is HTML?

Programming Language VS Markup Language

Markup languages didn’t evolve from Turing. In fact, they’re not related to computers at all. Instead, they come from the publishing world.

The term markup is derived from the traditional publishing practice of "marking up" a manuscript, which involves adding handwritten annotations in the form of conventional symbolic printer's instructions in the margins and text of a paper manuscript or printed proof. For centuries, this task was done primarily by skilled typographers known as "markup men" who marked up text to indicate what typeface, style, and size should be applied to each part, and then passed the manuscript to others for typesetting by hand. Markup was also commonly applied by editors, proofreaders, publishers, and graphic designers, and indeed by document authors.

http://en.wikipedia.org/wiki/Markup_language

Typesetting

These instructions would be used by typesetters to manually lay out metal “types” into words and lines of text. A page of these cast metal “types” would be bound together into a “forme". The forme was mounted into a printing press, inked, and used to make an impression on paper.
http://en.wikipedia.org/wiki/Typesetting

Movable Type

This process of using “movable type” - movable components to reproduce letters and punctuation - predates computers by centuries. The European mechanical printing press - the one that produced the Gutenberg Bible in the 15th century - is regarded as one of the key factors fostering the Renaissance. In fact, this process is over 1000 years old, dating back to China in the 10th and 11th centuries.
http://en.wikipedia.org/wiki/Movable_type

Welcome to the Digital Age

When computers arrived in the mid 20th century, publishers transitioned from character-by-character typesetting to fully-digital imagesetting. Instead of rendering text into an inky panel of lettered metal blocks, they rendered into a high-resolution digital image, then sent the image to a printing device.

The transition to digital publishing required the creation of 2 new types of computer languages. The first, the **markup language**, allowed humans to tell computers how to lay out a page of text. The second, the **printer control language**, allowed computers to tell printing devices how to print a digital image onto paper.

2 new languages

1. Markup language
2. Printer control language

Why two languages?

When digital printers emerged in the 1960s, they each had their own printer control language. You couldn't use the same language to talk to another brand of device. This was annoying so scientists attempted to standardize on one markup language for humans. Computers would translate this general purpose markup language into the appropriate printer control language for your device.

First came William Tunnicliffe's GenCode and Charles Goldfarb's GML in the 1970s. Goldfarb later combined GenCode and GML into SGML in 1974, and collaborated with Tunnicliffe on making it an international standard.

Donald Knuth created another markup language called TeX in the 70s and 80s that became (and remains to this day) popular for printing academic and scientific journals.

Tim Berners Lee

Sir Timothy John "Tim" Berners-Lee is a British computer scientist, best known as the inventor of the World Wide Web.

He made a proposal for an information management system in March 1989 and he implemented the first successful communication between a Hypertext Transfer Protocol (HTTP) client and server via the Internet sometime around mid November 1989.

Fun Fact: In a Times article in October 2009, Berners-Lee admitted that the initial pair of slashes ("//") in a web address were actually "unnecessary". He told the newspaper that he could easily have designed URLs not to have the slashes. "There you go, it seemed like a good idea at the time"

Tim Berners Lee

- aka TimBL
- HTML was created in 1990 by Tim Berners-Lee
- TimBL invented the first browser and web server as well
- The first website was published in 1991 (info.cern.ch) in Europe
- http://www.w3.org/2004/Talks/w3c10-HowItAllStarted/?n=0

TimBL looked to SGML rather than TEX for inspiration when creating HTML. He considered HTML a specific application of the SGML standard.

HTML is now the main markup language for creating web pages and is likely the most used markup language in the world today.

Tim Berners Lee

In 2004, Berners-Lee was knighted by Queen Elizabeth II for his pioneering work. (In contrast, Turing never received knighthood) http://news.bbc.co.uk/2/hi/8249792.stm

Berners-Lee was honoured as the "Inventor of the World Wide Web" during the 2012 Summer Olympics opening ceremony, in which he appeared in person, working with a vintage NeXT Computer at the London Olympic Stadium. He tweeted "This is for everyone", which instantly was spelled out in LCD lights attached to the chairs of the 80,000 people in the audience.

W3C

- The World Wide Web Consortium (W3C)—directed by the inventor of the Web and HTML, Tim Berners-Lee—is the organization responsible for shepherding the development of Web standards.
- www.w3.org

Berners-Lee is the founder and director of the World Wide Web Consortium (W3C), which oversees the Web's continued development. He is also the founder of the World Wide Web Foundation, and is a senior researcher and holder of the Founders Chair at the MIT Computer Science and Artificial Intelligence Laboratory (CSAIL). He is a director of the Web Science Research Initiative (WSRI), and a member of the advisory board of the MIT Center for Collective Intelligence.

Specifications

- http://www.w3.org/TR/html5/

- Specifications (or specs, for short) are documents that define the parameters of languages like HTML and CSS. In other words, specs standardize the rules.
- With standards in place, we can build our pages from the agreed-upon set of rules, and browsers—like Chrome, Firefox, Internet Explorer (IE), Opera, and Safari—can be built to display our pages with those rules in mind.
- Specifications go through several stages of development before they are considered final, at which point they are dubbed a Recommendation

Specifications

- Why do we need specs?

Incompatible versions of HTML are offered by different vendors, causing inconsistency in how Web pages are displayed. The consortium tries to get all those vendors to implement a set of core principles and components which are chosen by the consortium.

Specifications

HTML5 is a *candidate recommendation*

- HTML5 is still in Candidate Recommendation status, not a "Recommendation" yet.
- It takes time (literally years) for the standardization process to run its course.
- Browsers begin to implement a spec's features long before it becomes a Recommendation,
- Browsers already include a wide variety of features in HTML5 and even though they aren't Recommendations yet.
- Quiz: What is the risk of using HTML5 features now?

Validating HTML

How do I know if my HTML meets the standards established by the specification?

W3C Markup Validation Service:
http://validator.w3.org/
Validator.nu
http://html5.validator.nu/

The validator will catch things that can escape the naked eye. Invalid HTML is a great way to get unexpected, inconsistent behavior because each browser may handle it differently. Examples: unclosed or mismatched tags, invalid or broken attributes, quotes where they shouldn't be, unterminated entity strings, improper nesting, missing required attributes, etc.

Web Page components

- Text content
- References to other files
  - audio,video,images,styles,Javascript
- Markup
  - HTML elements (around 100)

Quiz: Which HTML elements do you already know?

HTML5 - A little bit of history...

HTML 1.0 was described in 1993. A competing HTML+ standard arrived in 1993 as well. Both drafts expired without being adopted.
HTML 2.0 was published in 1995 and *was* adopted as a standard.
HTML 3.0 was proposed in 1995, but wasn't adopted.
HTML 3.2 was published in Jan. 1997. 3.2 was adopted (then ignored by browsers).
HTML 4.0 was published in Dec. 1997 and added support for CSS.
By May 2000, after further updates, HTML settled on "HTML 4.01 Strict" as the international standard.
HTML5's first draft was published in 2008, and the process of standardization and adoption continues to this day.

Show this http://evolutionofweb.appspot.com/

Why HTML5?

XHTML had draconian error handling: Pages stop rendering when an error was found!
HTML5 has algorithms for parsing errors.

Plus, you get...

A variation of HTML called XHTML 1.0 was published in 2000, and the various international web organizations spent much of the 2000s perfecting it, before abandoning it in 2009 in favor of HTML5. XHTML attempted to combine HTML and XML, a more restrictive subset of SGML that was friendlier to computers, but harder on humans. "The attempt to get the world to switch to XML … all at once didn't work. The large HTML-generating public did not move “ said Tim Berners Lee in 2006.

HTML5 New Features!

- Integrated APIs
  - Video and Audio API
  - Local Storage
  - Geolocation
  - Messaging
  - Canvas
- new semantic HTML elements
- reduce need for third-party plugins like Java & Flash

What if….

...out of habit, I'm using some "old" HTML4 tags in HTML5?

Don't worry!
HTML is **backwards compatible**
HTML3.2 => HTML4 => HTML5

When can I use HTML5?

In 2022!

ishtml5ready.com
(View source if you need convincing)

Fun Fact: In 2008, Ian Hickson, the editor of HTML5, gave a now famous interview with TechRepublic, where he laid out his vision for the timeline of HTML5. By Hickson's estimation, HTML5 would issue a last call for comment in 2009, issue a candidate recommendation around 2012, issue test suites shortly after that, and finish testing around 2020. Now, the real eyebrow raiser was Hickson stating that he envisioned HTML5 reaching recommendation status in 2022.
Link to interview: http://blogs.techrepublic.com.com/programming-and-development/?p=718

Where is HTML5 now?

You don't have to wait till 2022 to start using some aspects of HTML5!

Many parts are ready to be used.

Increasing Number of devices and browsers feature partial implementation.

Current HTML5 support

What can I use now?
What level of support do browsers offer?

HTML5 support is changing rapidly:
caniuse.com
fmbip.com
html5test.com

Microsoft supports HTML5 starting with IE9. => Know your target audience!

HTML5 vs HTML4

Doctype has been simplified

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

<!DOCTYPE HTML>

If you know HTML4...you'll be comfortable with HTML5!

HTML5 vs HTML4

Script tag has been simplified
<script type=”text/javascript”></script>

<script></script>

Used for Javascript.

HTML5 vs HTML4

Link tag has been simplified

<link rel=”stylesheet” type=”text/css” href=”style.css”>

<link rel=”stylesheet” href=”style.css”>

Used to link in external CSS files.

HTML5 vs HTML4

meta declaration has been simplified

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

<meta charset="UTF-8">

The <meta charset> attribute sets the character encoding of an HTML document. It is extremely important that you explicitly declare the encoding of every HTML document; failing to do so will cause browsers to guess the encoding, which can lead to cross-site scripting attacks if they guess incorrectly.

https://code.google.com/p/doctype-mirror/wiki/MetaCharsetAttribute

Encoding

Which encoding should you use for HTML?

“Authors are encourage to use UTF-8”.

For HTML5, the default character encoding is UTF-8.

This has not always been the case. The character encoding for the early web was ASCII.

In theory, any HTML document can specify any character encoding. In practice, browsers ship with a limited set of encodings that they support.

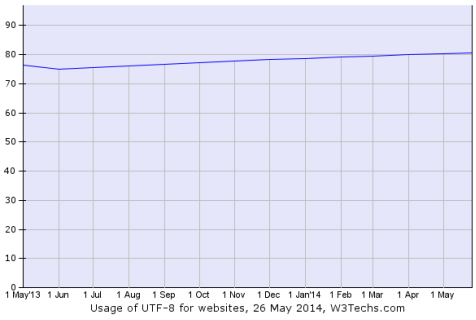
HTML 5 states: "Authors should not use JIS_X0212-1990, x-JIS0208, and encodings based on EBCDIC. Authors should not use UTF-32. Authors must not use the CESU-8, UTF-7, BOCU-1 and SCSU encodings. ... Authors are encouraged to use UTF-8."

UTF-8 is a variable-width encoding; it encodes each symbol using one to four 8-bit bytes. Symbols with lower numerical code point values are encoded using fewer bytes. So every possible unicode character is either 1, 2, 3, or 4 bytes.

Good read on Unicode & encodings: http://www.joelonsoftware.com/articles/Unicode.html

Usage of UTF-8

“UTF-8 is used by 80.7% of all the websites whose character encoding we know.”



Usage of UTF-8 for websites, 26 May 2014, W3Techs.com

http://w3techs.com/diagram/history_technology/en-utf8

From HTML4...

```
<div id="header">
<h1>Welcome<h1>
</div>
<div id="mainContent">
<h2>Main</h2>
<div id="footer">
<p>Footer</p>
</div>
```

Spot the differences...

to HTML5:

```
<header>
<h1>Welcome</h1>
</header>
<section>
<h2>Main</h2>
<footer>
<p>Footer</p>
</footer>
```

The HTML5 shown above is more *semantic* than the HTML4 shown. Let's go over what makes HTML semantic and why that matters in the first place.

Semantic HTML

- HTML markup describes the *meaning* of the content, that is, the semantics.
- HTML markup does not define *appearance* of the content; that's the role of CSS (Cascading Style Sheets).

If HTML doesn't define appearance, why is the <h1> tag bigger and bolder? Isn't that a contradiction?

Semantic HTML

- Some tags have a different appearance because every Web browser has a built-in CSS file (a style sheet) that dictates how each HTML element displays by default, unless you create your own that overwrites it.
- The default presentation varies slightly from browser to browser, but on the whole it is fairly consistent.
- When creating web pages, choose the elements that best describes the meaning of your content **without regard for their presentation.**

Semantic HTML

- Semantic HTML: <h1>Welcome to PetsMartOnline.com! </h1>
- Non-Semantic HTML: Welcome to PetsMartOnline. com!

Semantic HTML

- alternate voice
  - She said, Buenos Dias, Wyncode
- <b> mood
- strong importance
- stress emphasis
  - Wyncode, Miami’s first code school is enrolling now

Semantic HTML

Create this webpage fragment:

```
<!DOCTYPE html>
<html>
  <head>
   <meta charset="utf-8" />
  </head>
  <body>
    <h1>Buenos D&iacute;as Wyncode!</h1>
    <p>It’s a wonderful day to <em>write some code</em>!</p>
  </body>
</html>
```

Semantic HTML - why?

Why should you bother writing semantic HTML?

- Improved accessibility and interoperability (content is available to assistive technologies for visitors with disabilities, and to browsers on desktop, mobile, tablet, and other devices alike)
- Improved search engine optimization (SEO)
- (Typically) lighter code and faster pages
- Easier code maintenance and styling

Accessibility is the practice of making your content available to all users, regardless of their capabilities (see www.w3.org/standards/webdesign/accessibility).

HTML Syntax

- HTML consists of
  - element/tag
  - attribute
  - values

Wyncode

Quiz: add a title attribute to the element.

HTML5 adds custom attributes to the spec by data- prefix.

Character Escapes

```
Buenos D&#237;as Wyncode!

Buenos D&#x000ED;as Wyncode!

Buenos D&iacute;as Wyncode!
```

You can use a character escape to represent any Unicode character in XML or (X)HTML using only ASCII characters.

A decimal NCR. This uses a decimal number to represent the same Unicode code point.

A hexadecimal NCR. All NCRs begin with &# and end with ;. The x indicates that what follows is a hexadecimal number representing the code point value of a Unicode character. The hex number is not case-sensitive

A character entity reference. Character entity references are defined in the markup language definition. This means, for example, that for HTML only a specific range of characters (defined by the HTML specification) can be represented as character entity references (and that includes only a small subset of the Unicode range). Note that the entity name is case sensitive: in HTML, Á represents the *uppercase* letter Á, whereas á represents the lowercase á.

http://dev.w3.org/html5/html-author/charref
http://unicode-table.com/en/

Do I need HTML Entities?

Buenos Días Wyncode!

If the encoding is set correctly (and the document is saved as UTF-8) you should be able to work with just the characters.

From the W3C:

Using an encoding such as UTF-8 means that you can avoid the need for most escapes and just work with characters.

The Mac has a Character Viewer that you can turn on in the Keyboard System Preference, and you can find and then drag and drop the characters you need, or use the matching Keyboard Viewer to see which keys to type. For example, trademark is Option+2.

http://www.w3.org/International/questions/qa-escapes

However, you still need to use entities for the reserved HTML markup characters: less than, greater than, ampersand.

Also, if you use an HTML entity name, or number, the character will always display correctly.
This is independent of what character set (encoding) your page uses!

HTML Comments

<!-- This is a comment -->

<title>

- The text between <title> and </title> appears as the title at the very top of the browser window and on a browser tab.
- Additionally, it's typically the default name of a browser bookmark or favorite and is valuable information for search engines.

http://www.w3.org/TR/html51/

- The em element means "stress emphasis."
- Remember that because HTML describes the meaning of content, em dictates semantic, not visual, emphasis even though it’s common to render em text in italics.

http://www.w3.org/TR/html51/

- The img element is the primary choice for displaying an image
- The alt attribute provides text that displays if the image doesn't load or if the page is viewed in a text-only browser (e.g. by blind people).

```
<img src="profile.jpg" alt="Profile Pic" />
```

http://www.w3.org/TR/html51/

<a>

```
I’m with <a href="https://www.wyncode.co"
>Wyncode.</a>
```

- <a> defines a link to another page with the a element (“anchor”), which is the most powerful element in all of HTML because it makes the Web a web (as in a web of connected pages).
- It links one page to another page or resource

<small>

```
<small>&copy; Wyncode</small>
```

- Initially, this element was intended to make text smaller than regular text.
- However, in HTML5 small represents fine print, such as a legal disclaimer

<input>

```
<input type="text" name="lastname">
```

The <input> tag specifies an input field where the user can enter data.

<input> elements are used within a <form> element to declare input controls that allow users to input data.

An input field can vary in many ways, depending on the type attribute.

Buttons

```
<input type="submit" value="Submit" />

<input type="button" value="Submit" />

<button type="button">Submit</button>
```

The <button> tag defines a clickable button.

Inside a <button> element you can put content, like text or images. This is the difference between this element and buttons created with the <input> element.

<header>

```
<header>
Logo
Navigation
</header>
```

If a section of your page has a group of introductory or navigational content, mark it up with the header element.

<nav>

```
<nav>
Navigation
</nav>
```

Earlier versions of HTML didn’t have an element that explicitly represents a section of major navigation links, but HTML5 does: the nav element. Links in a nav may point to content within the page, to other pages or resources, or both.

<article>

```
<article>
</article>
```

You can use article to contain content like a newspaper article. However, it isn't limited to that.

The article element represents a self-contained composition in a document, page, application, or site and is, in principle, independently distributable or reusable, e.g., in syndication. This could be a forum post, a magazine or newspaper article, a blog entry, a user-submitted comment, an interactive widget or gadget, or any other independent item of content.

<section>

```
<section>
</section>
```

The section element represents a generic section of a document or application. A section, in this context, is a thematic grouping of content, typically with a heading.

Examples of sections would be chapters, the various tabbed pages in a tabbed dialog box, or the numbered sections of a thesis. A Web site's homepage could be split into sections for an introduction, news items, and contact information.

<article> vs <section>

The article and section elements are pretty similar.

How Do you Decide between article and section?

Is your content an independent piece of content or a widget that would be appropriate for syndication? It doesn't mean you have to syndicate or otherwise distribute article content, just that the content is fit for it.

If yes, use article.

Otherwise, in most cases use section.

<aside>

```
<aside>
</aside>
```

Sometimes you have a section of content that is tangentially related to the main content on your page but that could stand on its own. In this instance, use the <aside> tag.

Examples of aside include a pull quote, a sidebar, a box of links to related articles on a news site, advertising, groups of nav elements (for instance, a blog roll), a Twitter feed, and a list of related products on a commerce site.

<footer>

```
<footer>
</footer>
```

The footer element represents a footer for the nearest article, aside, blockquote, body, details, fieldset, figure, nav, section, or td element in which it is nested. It's the footer for the whole page only when its nearest ancestor is the body. And if a footer wraps all the content in its section (an article, for example), it represents the likes of an appendix, index, long colophon, or long license agreement, depending on its content.

Generic Containers

If neither article, section, aside, header, footer, or nav are applicable semantically, you need to use a generic container such as div (for division) or span.

Like header, footer, article, section, aside, nav, h1–h6, p, and other block-level elements, div automatically displays on a new line by default.
Span, however, is an inline element.

,

- The tag identifies a logical division of the page, like a banner, navigation bar, sidebar, or footer.
- You can place tags around individual words and phrases (often called inline elements) within paragraphs to format them independently.

,

- div is a block
- span is used for inline elements

Example

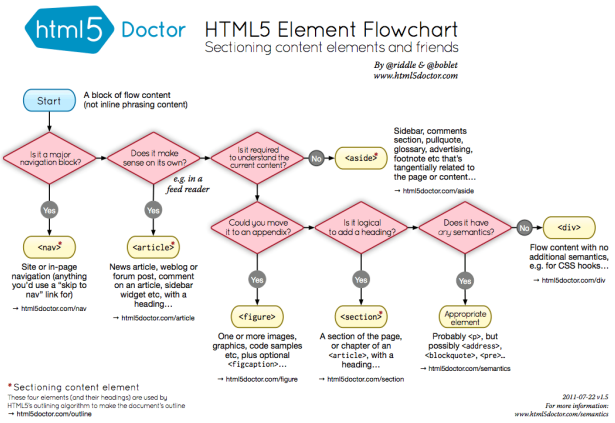
```
<div id="main">
<p><span>Content</span> goes here</p>
</div>
```

Sectioning Elements

In HTML5, the following elements are considered sectioning elements:

- article
- section
- nav
- aside

HTML5 Element Flowchart



http://html5doctor.com/downloads/h5d-sectioning-flowchart.png

Final HTML5

```
<body>
<header><h1>Intro to HTML5</h1></header>
<article>
  <h1>Buenos D&iacute;as Wyncode!</h1>
   <p>It's a wonderful day to <em>write some some</em>!
</p>
 </article>
<footer><p><small>Copyright Rima Gerhard.
</small></p></footer>
</body>
```

HTML nesting

- When elements contain other elements, each element must be properly nested, that is, fully contained within its parent.
- In other words, first open element 1, then open element 2, then close element 2, and then close element 1

Quiz: What are the issues in this HTML?

```
<body>
<p><a href="#">Link</p></a>
```

Document Outline

http://gsnedders.html5.org/outliner/

- Each HTML document has an underlying outline, which is like a table of contents, as defined by the heading elements.
- The outline isn't something that displays in your page explicitly

HTML5 & IE8

Getting IE8 to understand HTML5:

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

Unfortunately, Internet Explorer 8 and earlier don't recognize new HTML5 tags, and won't respond to any CSS you apply to them.
There is a way to kick those old versions of IE into gear, so they'll understand all the CSS that applies to HTML5 tags.

This is an "Internet Explorer conditional comment" (IECC for short) to embed a bit of JavaScript code that's only visible to versions of Internet Explorer earlier than IE 9. In other words, only IE 6, 7, and 8 respond to this code, and all other browsers (including newer versions of IE) simply ignore it. This code makes earlier versions of IE load a small JavaScript program that teaches the browser to recognize HTML5 tags and apply CSS to those tags. This code only affects how the browser displays and prints HTML5 tags; it doesn't make the browser "understand" an HTML5 tag that actually does something. For example, IE 8 and earlier don't understand the <video> tag and can't play HTML5 video (even with the added JavaScript code).

Accessibility

Accessibility Guidelines
http://www.w3.org/TR/WAI-WEBCONTENT/full-checklist.html

Note that accessibility is a strict requirement for many government sites.

Debugging HTML

View Source!

Also:
Chrome Developer Tools
IE Developer Tools
Firefox Developer Tools

Demo of Chrome Developer Tools: Element Tab
View Source