HTML
What The Web Looks Like

HTML
HyperText Markup Language

HTML5
Treemaps are static on January 25, 2013
- W3C Recommendation
- Proposed Recommendation
- Candidate Recommendation
- Last Call
- Working Draft
- Non-W3C Specifications
- Deprecated

Some HTML

```html
<!doctype html>
<html>
<head>
  <title>Hello World</title>
</head>
<body>
  <p>A Basic HTML Page.</p>
</body>
</html>
```

HTML Defines Content and Structure

- Content consists of Text, Images, Links, Media Assets, etc
- Structure defines the basic formatting and semantic meaning of elements
  - i.e. `<title>Hello World</title>` defines the title of the page
- Programs can analyze the structure of a document to derive meaning
  - `<h1>, <h2>, <h3>` tags could be used to generate a document outline
  - Headers in a table `<th>` could be used by screen readers to describe data to a visually impaired individual
- We can use the document structure to define display styles

Structure of an Element

- The entire line is referred to as the **title element**
- The `name` of this element is "title"
- `<title>` is an **opening tag**
- `</title>` is a **closing tag**
- Hello World is the **content** of this element
Not All Elements Need a Closing Tag

```html
<body>
  <p>
    Paragraph elements can have closing tags
  </p>
  <p>
    or not
  </p>
  <ul>
    <li>
      List Item elements
    </li>
    <li>
      may also omit closing tags
    </li>
  </ul>
</body>
```

http://www.w3.org/TR/html5/syntax.html#optional-tags

Question...

```html
<body>
  <p>
    Is this first bit of text the content of a Paragraph element?
  </p>
  <p>
    What about this second bit of text?
  </p>
</body>
```

The HTML Docs

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

- Is the first bit of bare text a legal content for the `<body>` tag?
- Where in the HTML specification is this question answered?

HTML vs XHTML

- HTML is not XML compliant
  - No namespaces
  - More forgiving parsing
- XHTML is XML compliant
  - Namespaces
  - Very strict document parsing. Any errors usually result in the document completely failing to be displayed
  - Not allowed to omit closing tags
  - `void elements` must be self closed
    - `<br/>`

Not All Elements Have Content

- `<br>` the Break tag acts as a newline character for HTML.
- `<hr>` the Horizontal Rule tag draws a line across a page.
- `<img src="foo.gif">` the Image Tag tells the browser to go load an image in this location.
- These elements are called **void elements** and must not have closing tags.

http://www.w3.org/TR/html5/syntax.html#void-elements

Attributes

- `<img src="foo.gif" class="thumbnail">`
- Attributes for an element are defined in the element's opening tag.
- Attributes always have an attribute name.
- Attributes may optionally have a value.
- Attribute values may be surrounded with either single quotes, double quotes, or nothing, depending on the content of the value.

http://www.w3.org/TR/html5/syntax.html#attributes-0

HTML Elements
Categories of Elements

- Elements may belong to multiple categories
- We’ll mostly be dealing with Flow and Phrasing elements

Flow Elements

3.2.4.1.2 Flow content
Most elements that are used in the body of documents and applications are categorized as flow content.

- a, abbr, address, area, article, aside, audio, b, bdi, bdo, blockquote, br, button, canvas, cite, code, data,datalist, del, dfn, div, dl, em, embed, figure, footer, form, h1, h2, h3, h4, h5, h6, header, hr, i, iframe, img, input, ins, kbd, keygen, label, main, map, mark, math, nav, noscript, object, ol, output, p, pre, progress, q, ruby, s, samp, script, section, select, small, span, strong, sub, sup, svg, table, template, textarea, time, u, ul, var, video, wbr
Phrasing Elements

3.2.4.1.5 Phrasing content

Phrasing content is the text of the document, as well as elements that mark up that text at the intra-paragraph level. Runs of phrasing content form paragraphs.

- a abbr area [if it is a descendant of a map element] audio b bdi bdo br button canvas cite code datalist del div dl embed em
- iframe img ins kbd label map math meter noscript object output progress qruby s samp script select small span strong sub sum template textarea time u var video wbr

Most elements that are categorized as phrasing content can only contain elements that are themselves categorized as phrasing content, not any flow content.

http://www.w3.org/TR/html5/dom.html#phrasing-content-1

Pre-HTML5

- Pre HTML5, most elements were loosely grouped into "Block Elements" and "Inline Elements"
- HTML5 strives to better separate the logical structure of a document from the display
- Block Elements were mostly grouped by how they were displayed in a browser, i.e. expanding to fill the available width of a container.
- Browsers will likely continue to render them like this, but it is no longer mandated by the specification

Block Elements

```html
<!doctype html>
<html>
<head>
<title>Block Elements</title>
<style>
p, blockquote, ul { border: 1px solid black; }
</style>
</head>
<body>
<p>Paragraphs</p>
<blockquote>Block Quote</blockquote>
<ul>
<li>Ordered Lists</li>
<li>Un-ordered Lists</li>
</ul>
</body>
</html>
```
Inline Elements

```html
<!doctype html>
<html>
<head>
<title>Inline Elements</title>
<style>
strong, mark, img { border: 1px solid black; }
img { width: 14px; }
</style>
</head>
<body>
<p>Lorem ipsum dolor sit amet, <strong>consectetur adipisicing elit</strong>, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.</p>
</body>
</html>
```

- The `<!doctype ...>` preamble is not an HTML element.
- `<!doctype ...>` tells the rendering engine what type of markup to expect.
- HTML4.1 Transitional
- HTML5
  - `<!doctype html>`

DOCTYPEs are required for legacy reasons. When omitted, browsers tend to use a different rendering mode that is incompatible with some specifications. Including the DOCTYPE in a document ensures that the browser makes a best-effort attempt at following the relevant specifications.

http://www.w3.org/TR/html5/syntax.html#the-doctype

```
<html>
  
  • The `<html>` element is the root element of our element tree
  • The HTML Element can only be preceded by whitespace characters and comments
  • The HTML Element can only have two children: one `<html>` element and one `<body>` element
  • From the HTML specification:
    • An html element's start tag can be omitted if the first thing inside the html element is not a comment.
    • An html element's end tag can be omitted if the html element is not immediately followed by a comment.
```
<head>
• The <head> element represents a collection of metadata for the Document.
• A <title> tag is the only required child element

```html
<head>
  <meta charset="utf-8">
  <base href="http://www.example.com/">
  <title>A New Hope</title>
  <link rel="stylesheet" href="default.css"></link>
  <script src="example.js"></script>
</head>
```

<body>
• The <body> element represents the content of the Document.
• Basically this holds everything you see.

```html
<body>
  <img src="http://goo.gl/5LEYnp" alt="regrets">
  <form action="search.php" method="post">
    <input type="text" name="search">
    <input type="submit" value="Find Droids">
  </form>
</body>
```

Nested Elements

• We’ve already seen a lot of nested elements

```html
<!doctype html>
<html lang="en">
<head>
  <title>Validator!</title>
</head>
<body>
  <p>Let’s Test it</p>
</body>
</html>
```
Nested Elements

- Nested element tags must be balanced in a last-in, first-out manner.
- A close tag must always balance the most recent still-open tag.
- After the word “example” if we want to close something, we must close the `<em>` tag first.

Nested Elements

- What does the browser do if the elements are incorrectly nested?

Links

- `<a>` Anchor tag
- Used to define a link to another document, or location in the same document.
Links

- `href` attribute defines what to link to.
  - This is the Hyper in HyperText
  - Must contain a valid URL
    - Universal Resource Locator

URL

- A basic absolute URL
  - `http://www.arizona.edu`

- A basic relative URL
  - `../images/image.png`

URL

```
Username  Password  Path  Query String
http://user:pass@example.com:80/path?query=yes#fragment
```

- Scheme
- Host
- Port
- Fragment

URL

- Most of these parts are null most of the time
- The following are all valid URLs
  - `https://example.com`/path/to/something.html
  - mailto:fischerm@email.arizona.edu
  - foo
  - `//ajax.googleapis.com/libs/jquery.min.js`
  - `../somepage.php?key=123`
  - anotherpage.html#figure1
  - #droids
URI, URL, URN

- URI - Universal Resource Identifier
- URL - Universal Resource Locator
- URN - Universal Resource Name

These are NOT interchangeable. Each has a different meaning, although there can be significant overlap.

- We’re almost always going to use URLs unless otherwise explicitly mentioned.

URI

The generic URI syntax consists of a hierarchical sequence of components referred to as the scheme, authority, path, query, and fragment.

\[
\text{URI} = \text{scheme} :// \text{hier-part} \ [\ ?\ \text{query}] \ [\ #\ \text{fragment}] \\
\text{hier-part} = \text{”/” authority path-absolute} \\
\text{/ path-absolute} \\
\text{/ path-rootless} \\
\text{/ path-empty}
\]


Links

- Let’s assume we’re currently on the following page:
  - http://www.cs.arizona.edu/classes/cs337/spring16/
- There are two links on this page that are functionally equivalent:
  - href="http://www.cs.arizona.edu/classes/cs337/spring16/syllabus.html"
  - href="syllabus.html"
- The first one is an "absolute" URL and the second is a "relative" URL. Clicking on either of them from that page produces the same result.
- Is one preferable to the other? What circumstances would make you choose one way over another?
Anchor Fragments

The last part of a URL, following a # symbol targets a fragment within a page
Can target an element with a matching id attribute

URL Schemes

The Scheme tells the browser how to access the resource.
- file:// loads the file directly from the local filesystem
- http:// initiates an HTTP connection over TCP/IP
- https:// establishes a secure connection over SSL, then communicates via HTTP
- email: hands off control to an email client
- tel: hands off control to a phone client

More Elements

Ordered and Unordered Lists

- <ol> Ordered List
- <ul> Unordered List
- <li> List Element - Used for both types of lists
- Closing Tag for <li> may be omitted
Tables

- `<table>` begins a table
- `<th>` table header
- `<tr>` table row
- `<td>` table data

```
<table>
<thead>
<tr>
<th>Jedi</th>
<th>Lightsaber Color</th>
</tr>
</thead>
<tbody>
<tr>
<td>Luke Skywalker</td>
<td>Blue</td>
</tr>
<tr>
<td>Yoda</td>
<td>Green</td>
</tr>
<tr>
<td>Darth Vader</td>
<td>Red</td>
</tr>
</tbody>
</table>
```

Headings

- `<h1>` 1st level heading
- `<h6>` 6th level heading

```
A Midsummer Night’s Dream
Act I, Scene 1
```

Generic Block and Inline Elements

- `div` is a generic block element. It takes up all available horizontal space.
- `span` is the generic inline element. It takes up only as much horizontal space as its content.

```
<div>
  Pony ipsum dolor sit amet fluttershy hoof eu Daring Du Sweetie Drops.
</div>

<div>
  Cloud <span>officia Omnisincus pie</span> muffin Angel horn enim gryphon honesty.
</div>

<img src="pony.png"/>
```

Generic Block and Inline Elements

- Without styling it’s very difficult to discern the document structure with just `div` and `span` elements.
Generic Block and Inline Elements

• CSS classes are required to give the document some structure.

Semantic Tags

• HTML5 placed a new emphasis on semantics
• The meaning of a document should be understandable from the DOM, and not need visual styles to understand.

Semantic Tags

• The `<article>` element represents a complete, or self-contained, composition in a document. This could be a forum post, a magazine or newspaper article, a blog entry, etc.
• The `<header>` element represents introductory content for its nearest ancestor.
• The `<figure>` element represents some flow content, optionally with a caption, used to annotate illustrations, diagrams, photos, code listings, etc.
Styling with Tags

• The <pre> tag represents a block of preformatted text, in which structure is represented by typographic conventions rather than by elements.

• Whitespace and newlines are rendered as-is

Styling with Tags

• The <blockquote> renders as indented text

Styling with Tags

• Both pre and blockquote are pretty old elements.
• Defined in HTML 1.0
• Are these pure structure elements?
• Are the header tags (<h1> … <h6>) similar?
• What other tags typically define both structure and style?

http://www.w3.org/Markup/draft-ietf-ilir-html-03.txt

Images
Images

• Something other than text!
• The img tag is a void element, so it has no closing tag
• By default images are displayed at their native pixel size

Images

• Images can be resized with CSS, or with width and height attributes.
  • Resized images are not resampled. The full image is sent to the browser no matter what size the image is ultimately displayed at.
  • Assigning just width or height will scale the image and preserve the aspect ratio. (width:height)
• Examples!

Images

• The alt attribute should always be present, and should describe the image as best you can.
  • Accessibility should be thought about from the very start of an HTML project, and not at the very end.
  • If an image provides no useful information (a spacer image, or background gradient) an empty alt attribute should be used: alt=""

Images

• Three widely supported Image formats
  • GIF - Graphics Interchange Format
  • JPEG - Joint Photographic Experts Group
  • PNG - Portable Network Graphics
• HTML Specification does not mandate support for any particular format
GIF
- 256 distinct colors. Each GIF can have its own color pallet.
- One color can be designated as transparent.
- Can contain multiple frames for animation.
- Lossless compression, but limited format.

JPEG
- Millions of colors
- Lossy compression
  - Higher quality, less compression, larger file size
  - Smaller file size, higher compression, less quality
- Designed to be good at compressing photographs.
- No transparency

PNG
- Lossless compression
- No animation
- Several bit depth variants
  - PNG-8: 256 colors
  - PNG-24: 16 Million colors (3 8-bit channels)
  - PNG-32: 16 Million colors + 8-bit transparency
    - Allows for smooth anti-aliased transparency

Images

<table>
<thead>
<tr>
<th></th>
<th>GIF</th>
<th>JPEG</th>
<th>PNG</th>
</tr>
</thead>
<tbody>
<tr>
<td>Photograph</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Animated</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Icon or Drawing</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Transparency</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
</tbody>
</table>
Images

- You can spend a lot of time working on images
- Good discussion of images in *Head First HTML and CSS*

Character References

- The & and < characters are special, and are allowed in some places but not others
- `<p title="""> ... ok
- `<p>a<b</p> ... problem
- `<p>a<b</p> ... problem
- Solution: Use `&amp;` and `<` instead
  - `<p>a&lt;b</p>

Character References

- Used to escape certain characters
- Three different styles
  - & name: `&copy;` ©
  - &# decimal-digits: `&9733;` ★
  - &fx hex-digits: `&x221E;` ∞
- Mostly we'll use `&amp;` `&lt;` `&gt;` `&quot;` `&apos;` `&nbsp;`

[http://www.w3.org/TR/html5/syntax.html#named-character-references](http://www.w3.org/TR/html5/syntax.html#named-character-references)
• What category are the `<p>`, `<ul>`, and `<li>` elements?

• What are the text elements?

• Would all leaf elements be phrasing content?
Misc Details

- HTML Tags and attribute names are **not case sensitive**
- Comments: `<!--   -->`
  - Cannot nest comments. No inline comments
- Whitespace is mostly ignored. Multiple whitespace characters are condensed to a single space when rendered
- Text nodes and attribute values must be a tab, newline, form-feed, carriage-return or unicode characters ≥ than U+0020 (space)

Rendering
That Thing Browsers Do

```html
<!doctype html>
<html>
<head>
<title>Lorem Ipsum</title>
</head>
<body>
<p>Lorem ipsum dolor sit amet</p>
<p>consectetur adipisicing elit</p>
<ul>
<li>sed do eiusmod tempor incididunt</li>
<li>tempor incididunt</li>
<li>©</li>
<li>™</li>
<li>™™</li>
</ul>
</body>
</html>
```

A quick aside... Lorem Ipsum

- Nonsense text to fill up space. Its latin-ish...
- Great to block out sections of a design.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum

[http://www.lipsum.com](http://www.lipsum.com)
Lorem Ipsum

Bob Ross Ipsum

Little trees and bushes grow however makes them happy. I sincerely wish for you every possible joy life could bring. If you don’t think every day is a good day - try missing a few. You’ll see. And that’s when it becomes fun - you don’t have to spend your time thinking about what’s happening - you just let it happen. Of course he’s a happy little stone, cause we don’t have any other kind.

http://www.bobrosslipsum.com

• Riker Ipsum - http://www.rikeripsum.com
• Hodor Ipsum
• Samuel L Ipsum (nsfw)

.. back to rendering

• A rendering engine takes HTML and turns it into something you can see (or hear)
• Gecko - Firefox
• KHTML - Konqueror
• Trident - Internet Explorer
• WebKit (based on KHTML) - Safari
• Blink (based on WebKit) - Chrome

Chrome Developer Tools
Validation and Errors

Validation

<!doctype html>
<html lang="en">
<head>
<title>Validator!</title>
</head>
<body>
<p>Let's Test it</p>
</body>
</html>

http://validator.w3.org/#validate_by_input

Errors

- Check errors in validator
- Check for errors in browser developer tools.
- Examples

- Why are browsers so permissive in handling errors?
- What risks are posed by being so forgiving?
- Do you think browsers are as forgiving with CSS and JavaScript as they are with HTML?
- What other types of software is as permissive?
HTML Review

- `<doctype html>` starts a document. NOT an element
- Structure of an HTML element
- Attributes
- Open / Closing tags. Optional Tags
- "Block/Inline" elements
- Flow / Phrasing elements

- `<html> <head> <body>`
- Nested Elements
- Links / Anchors / Fragments / Targets
- URLs
- Lists
- Tables
- Headings

- `<pre>`
- `<blockquote>`
- `<img>`
  - GIF, JPEG, PNG, Transparency
  - Alt attribute
  - Character References
    - `&amp;` &lt; &gt; `&quot;` `&apos;` &nbsp;
• DOM Tree
• Comments
• Whitespace
• Validation