HTML5
INTERACTIVE WEB SITES
LECTURE 2
CS 50.12 :: Fall 2013 :: Corrine Haverinen
Santa Rosa Junior College

TEXTBOOK WEB SITE

- **Introducing HTML5**
  Errata, links, example files, some new stuff

MINIMAL HTML5 DOCUMENT

```html
<!DOCTYPE html>
<meta charset="utf-8">
<title>Always Title Your Documents</title>
<p>Your exciting new stuff</p>

Should we use <html> <head> <body> tags?

MINIMAL HTML5 DOCUMENT, SAFER

Reasons to use <html> <head> <body> tags

- <html> element with language good for accessibility
  <html lang="en">
- IE8 and below:
  require the <body> element in order to apply CSS

MINIMAL HTML5 DOCUMENT, SAFER

```html
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Always Title Your Documents</title>
</head>
<body>
<p>Your exciting new stuff</p>
</body>
</html>

SEMANTIC HTML
• conveying the meaning behind Web pages
• understanding how things are related to each other
• embed semantics in Web docs with:
  ▪ HTML elements, body, #d headings, lists, blockquote
  ▪ use new HTML5 sectioning elements
• use the correct element or tag to describe the content
• semantic tags used for their purpose, not for styling
• ARIA enhancement
• Microdata such as Microformats, RDFa

REASONS TO USE SEMANTIC HTML
• lighter, cleaner code
• accessibility
• search engine optimization
  ▪ if your important content is not in the HTML (JS, CSS, etc)
    search engines will not see it
  ▪ easier for search engines to get what your site is about
  ▪ semantically correct the right way as opposed to such practices as hidden text used to
    increase relevance
• Hold meaning even though repurposed (RSS feeds, redesign of layout)
• Easier for developers to work with

OLD METHOD - DIV TAGS
• <div> based layouts - standard way to layout and style content
• <div> is flow content element, does not stand apart in the structure of the document
• <div> tags have no meaning semantically, no impact on the computable structure of the
  page or document outline

HTML5 NEW SEMANTIC ELEMENTS

<header>
<nav>
<hgroup>
<main> NEW!!
<section>
<article>
<aside>
<footer>
HTML5 ELEMENTS - OLD BROWSERS

- New tags styable in all modern browsers...
  except Internet Explorer IE8 and below (works in IE9)

- Browsers use their own style sheets when rendering a page
  - HTML5 elements are not on some older browsers style sheets
  - New HTML5 tags will not be treated as block level elements
  - CSS will not be applied

HTML5 SHIV

- Make new tags styable in all modern browsers...

- JavaScript is used to apply CSS to HTML5 elements
  - Remy Sharp originally made an enabling script
  - Put up on Google, all you need to do is reference Google script
  - html5shiv - put this in the head of HTML5 docs

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

HTML5 SHIV IS A CONDITIONAL COMMENT

- Only recognized in IE browsers
  - good for giving special instructions to IE browsers
  - supported back to IE5

- basic structure is same as an HTML comment  <!-- -->

- [if lt IE 9] = less than IE 9
- [if lte IE 9] = less than or equal to IE 9

OTHER HTML5 SHIV SOLUTIONS

- Newer version includes support for printing, dynamic injection of HTML5 elements
  - Latest Version: need to Download Script and link to it

- If using Mondernizr, html5 shiv will be included
  - Modernizr checks the availability of HTML5 features
  - Used for detection, does not fix, except with the html5shiv

<HEADER>
- a container for a group of introductory or navigational aids
- a document can have multiple header elements
- can be used within section and article elements
- does not have to be at the top of the document
- not needed if all you have is a heading tag
- not to be confused with `<head>`

`<NAV>`
- indication to the browser that element contains navigation
- elements within excluding from the document outline
- intended for major navigation
- groups of links, not enough reason to use
- quite often appears within a header

`<HGROUP>`
- related multilevel headings
- title + tagline or subtitle
- example: **Web Designer Wall**

`<SECTION>`
- thematic grouping of content
- closest to a div container
  - div has no semantic meaning
- used explicitly for grouping related content
  - section is for generic chunk of related content
  - while headers, footers, navs & asides are chunks of specific content
- when replacing divs with sections, ask...
  - Is all of the content related?

**USE A `<DIV>`**
- if what you need is just a wrapper for styling
- if you would not use a heading

`<ARTICLE>`
• independent item of content
• self-contained post, news-item, comment
• interactive widget
• comments on blogs are articles within articles
• also a special type of <section>

<MAIN>
• Represents the main content of the body of a document or application
• Replaces content wrapper
• Target for Skip to Navigation link for accessibility
• Consists of content that is directly related to or expands upon the central topic of a document or central functionality of an application
• Must not include more than one main element in a document
• Must not include the main element as a child of an article, aside, footer, header or nav element

<ASIDE>
• tangential content
• side comments
• pull quotes
• like a sidebar, but not in postioning
  • it is the content that matters

<FOOTER>
• contains information about its containing element
  • section
  • article
  • body
• author, copyright, links to related content or nav
• privacy policies, disclaimers, legal restrictions
• typically at end of section, but does not have to be

example - Storm Large

2 METHODS OF USING NEW SEMANTIC ELEMENTS
• use nested divs with semantic classes
  `<article><div class="article">
  .. </div></article>`
  `.article { /* article tag styling here */}`

• replace divs with semantic elements
  ▪ think about the content, use them as warranted
  ▪ new elements created not to replace divs...
  ▪ but to provide user agents new ways of understanding content

OUTLINING

• like a table of contents for a Web page
• outlining algorithm clearly defined in HTML5 spec
• support is slow to come
• can be used by assistive technology and search engines
• does not read CSS
• uses sectioning elements to create

OUTLINE

<table>
<thead>
<tr>
<th>1. BLOG.COM</th>
</tr>
</thead>
<tbody>
<tr>
<td>2. Your site slogan</td>
</tr>
<tr>
<td>1. Your main article title</td>
</tr>
<tr>
<td>1. Article title</td>
</tr>
<tr>
<td>2. Article title</td>
</tr>
<tr>
<td>3. Article title</td>
</tr>
<tr>
<td>4. Article title</td>
</tr>
<tr>
<td>1. Navigation</td>
</tr>
<tr>
<td>2. Useful Links</td>
</tr>
<tr>
<td>3. Useful Links</td>
</tr>
<tr>
<td>4. Our office</td>
</tr>
<tr>
<td>2. Color Switcher Options Styles</td>
</tr>
<tr>
<td>1. Default Colors</td>
</tr>
</tbody>
</table>

OUTLINING SITES

• HTML 5 Outliner at gsnedders
• Chrome extension
• Google Bookmarklet
HTML4 SECTIONING

- In HTML4, every section is part of the document outline
  - But documents often not that linear
  - no way to have section containing info related not to the doc but site as a whole - logos, menus, table of contents, etc

OUTLINING - SECTION ELEMENTS IN PLAY

- article
- aside
- h1 - h6 headings within a section
- nav
- section
- footer and header are NOT used in the Outline

HTML5 SECTIONING

- article, section, nav and aside are always subsections of their nearest ancestor section
- each section can have its own heading hierarchy
- the first HTML heading element defines the heading of the section
- the structure of the section determines the outline, not the heading rank of the sections

OUTLINING AXIOMS

- <body> is the first section
- <article> <section> <nav> <aside> make subsections
- subsections may contain more sections (subsections)
- footer belongs with the highest-level section

HEADING ORDER

- each section can have its own heading hierarchy
- all sections can have an h1
  - for screen readers that do not yet support HTML5 outline algorithm - best to order headings (one h1 on a page)
  - alternatively WHATWG recommends to either use only h1 throughout or rank appropriately within a section
- Outlining examples both ways on juicystudio.com
HEADINGS IN OUTLINING

- Sectioning elements that do not contain a child heading will be labelled as Untitled
  - to avoid Untitled, need to add headings to aside and nav
  - Don’t always want a heading, so OK to have Untitled

1. Modernizr
   1. Untitled NAV
   2. Untitled ASIDE
   3. Untitled SECTION
      1. Why use Modernizr?
      2. How it works
   4. Download Modernizr 2.6.2
      1. Untitled ASIDE

- Within an hgroup, outliner will disregard all headings except the highest ranking

SECTIONING ROOTS

- elements that have their own outlines
- sections and headings inside these elements do not contribute to the outlines of their ancestors

OUTLINING EXAMPLE SITES

- Modernizr
- Web Designer Wall
- dev.opera - Introduction to WAI ARIA
- Boston Globe

ACCESSIBLE RICH INTERNET APPLICATIONS (WAI-ARIA)

- NVDA – non visual desktop access
- Series of attributes that browsers and assistive technologies can hook into
- Easy to implement
- Can use to add semantic structure to a page
- Roles, states, updates of a widget are not available to assistive technology

ROLES DEFINE STRUCTURE OF DOCUMENT
• Helps screen reader understand role of section and orient within document
• Landmark roles include:
  <div role="article"> - standalone content
  <div role="banner"> - site oriented (title, logo)
  <div role="complementary"> - supporting
  <div role="contentinfo"> - footnotes, etc.
  <div role="main"> - directly related content
  <div role="navigation"> - links to navigate related content
  <div role="search"> - search form for site

EXAMPLE ADDING ROLES

ARIA test
Check out the code

MAKING WIDGETS KEYBOARD ACCESSIBLE

• we may use a graphic and scripting to create a richer experience
• visually, we can easily navigate, but not keyboard accessible
• use <role> attribute to define widget, page structure, etc.
• aria attributes are used to support the widget roles

EXAMPLE

<input type="image" src="thumb.gif"
  alt="Effectiveness"
  role="slider"
  aria-valuemin="0"
  aria-valuemax="100"
  aria-valuenow="42"
  aria-valuetext="42 percent"
  aria-labelledby="leffective">

MICRODATA

Give more semantics to your content
Jay’s Video
• Microformats
  ▪ simpler and more common, but less robust and standard
  ▪ small chunks of semantic markup for address book info, event listings and machine-readable metadata
• RDFa
• Microdata
  ▪ supported by HTML5 standard and schema.org
  ▪ implemented as 5 attributes

CSS REVIEW

• CSS syntax on w3schools.com
  ▪ w3schools How To - Cascade Order
• Review of CSS on HTMLdog.com
• maxdesign.com slideshows
  ▪ CSS Inheritance
  ▪ CSS Lineheight
  ▪ CSS Cascade
• Selectutorial CSS selectors