HTML

Lecture 2, CMSC 126

John Roy Daradal / Instructor

Previously on CMSC 126

- Internet
- World Wide Web
- Client-Server Architecture
- Web Programming

Today's Topics

- HTML
- Structure & Syntax
- HTML Elements

Today's Topics

- HTML vs XHTML
- HTML5
- HTML Good Practice

HTML

- **H**yper**t**ext **M**arkup **L**anguage
- Markup language used to display **web pages** in a web browser

HTML

- **Hypertext**: text that contains **hyperlinks**
- **Markup**: annotate documents

HTML

- *Original intent*: general **layout** of documents
- Controlled by *standards body* **W3C** and **WHATWG**

HTML File Extension

.html

- standard file extension

.htm

- arose from Microsoft Web Devt tools

.chm

- compiled HTML
- usually for Help files

Semantic HTML

- Choose tags based on **meaning** of *content*, not **appearance**
- *Example*: h1 or h3? big or em?

Separation of Concern

HTML

- content and structure of information

CSS

- styling and appearance

JavaScript

- behavior and interaction

Semantic HTML

- Important for **accessibility**
- Makes document *flexible* and *reusable*

HTML Structure

- Elements
- Tags
- Attributes

Elements

- HTML = collection of **elements**
- Element = **Tag** + **Content**
- **<tag> Content </tag>**
- *DOM*: document.getElementById

Tags

- Used to define HTML element
- Start / opening tags: **<tag>**
- End / closing tags: **</tag>**
- Correct **nesting**

Singleton Tags

- Tag with **no content**: **<tag />**
- Combines opening & closing tag in one tag
- aka *self-closing* or *empty tag*

Attributes

- Element properties: **attr = "value"**
- Values in *single / double quotes*
- *e.g.* <div id="header" class='dark'>...</div>

Global Attributes

- **id**: unique name assigned to element
- **class**: one or more class names, for styling
- **style**: inline styling
- **title**: tooltip info

Comments

- <!-- This is a comment -->
- Used for **disabling** page sections
- Cannot be *nested*

Character Entities

- **&lt;** = <
- **&gt;** = >
- **&quot;** = double quote
- **&apos;** = single quote
- **&nbsp;** = non-breaking space
- **&amp;** = &

Ignored by Browsers

- Comments
- Line breaks
- Multiple spaces and tabs
- Unrecognized tags

HTML Elements

- Structural
- Head
- List
- Links
- Images
- Text formatting

Layout Model

inline

- contain text or other inline elements
- e.g. *label, span, b*

block

- more significant; own space
- contain block or inline elements
- e.g. *p, h1, div*

Structural Elements

- Building blocks that provide structure in a web page
- *Example*: heading, paragraph, grouping elements

Structural Elements

html

- root element
- outer container for everything
- *descendants*: head, body

Structural Elements

head

- wrapper for *head* elements
- *required element*: title

body

- wrapper for **content** visible to user

Structural Elements

h1-h6

- **heading** elements
- *h1*: highest, *h6*: lowest
- contain text, inline elements

Structural Elements

p

- **paragraph** of text
- contain text, inline elements
- often interchanged with *div*

Structural Elements

div

- **container**: groups elements together
- hook for styling, interaction
- no semantic meaning
- contain other block, inline elements

Structural Elements

br

- line break / newline

- no semantics

hr

- horizontal rule

- line; divider

- *Singletons*: <br />, <hr />

Head Elements

- Meta information
- Stylesheets
- Scripts

Head Elements

title

- Set *page title*
- Displayed in browser tab, bookmarking, search engine results
- Meaningful titles for SEO

Head Elements

style

- Page-specific style embedded in HTML
- **Bad**: Mixes HTML and CSS!

```
<style type="text/css">

   h1 { color: red; }

</style>
```

Head Elements

link

- References **external stylesheet**
- Singleton

```
<link rel="stylesheet" type="text/css" href="basic.css" />
```

Head Elements

script

- Inline or external **JavaScript**

```
<script type="text/javascript">

   alert("Hello World!");

</script>



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

Head Elements

meta

- Provide **info** about document content
- Info for browser to decide how to render
- Simulate HTTP response headers

Head Elements

```
<meta name="author" content="John Roy Daradal" />

<meta name="description" content="Short description"

<meta name="keywords" content="keyword1, keyword2" />
```

Head Elements

```
<meta name="robots" content="noindex,follow" />

<meta http-equiv="Content-Type"

      content="text/html; charset=UTF-8" />

<meta http-equiv="Refresh" content="5" />
```

List Elements

- Unordered list
- Ordered list
- Definition list

List Elements

ul

- **unordered list**
- no order of precedence
- e.g. shopping lists, group of links

List Elements

ol

- **ordered list**
- order of importance, sequence
- e.g. rankings, steps

List Elements

li

- **list item**
- used by *ul, ol*
- can contain *nested lists*

List Elements

dl

- **definition list**
- key-value pairs

List Elements

dt

- **definition term**
- can be followed by multiple dd

dd

- **definition description**
- one dd can follow multiple dt

Links

- **a**: anchor tag
- *Lifeblood* of the Web
- **Links** billions of web pages together

Links

```
<a href="page2.html"> New Web Page </a>

<<a href="#references"> In-Page Link </a>

<a href="page3.html" target="_blank"> New Tab </a>
```

In-Page Links

```
<div id="header"> .... </div>


......


<a href="#header"> Back to Top </a>
```

Link Default Styles

- **Unvisited**: blue, underlined
- **Visited**: violet, underlined
- **Active**: red, underlined (on click)

Images

- **img**: singleton
- Excessive use **slows down** page load

```
<img src="photo.jpg" alt="Alternate Text" />
```

Media elements

- **applet**: Java applets
- **param**: used with applet to pass info
- **embed**: multimedia
- **object**: ActiveX, video, PDF, Flash, applets

Text Formatting Elements

- Some **useful**, some **deprecated**
- **Inline** elements

Text Formatting Elements

- b: **bold**
- i: *italic*
- No *semantic* meaning!
- Not deprecated, but used less

Text Formatting Elements

em

- **emphasize** text
- rendered in *italics*
- preferred over *i*
- **semantic meaning**: this text is *slightly* more important

Text Formatting Elements

strong

- **strongly emphasize** text
- rendered in *bold, non-italics*
- preferred over *b*

Text Formatting Elements

- **u**: underline
- **strike**: ~~strikethrough~~
- No *semantic* meaning!
- Deprecated

Text Formatting Elements

ins

- identify **inserted** content
- rendered with underline

del

- identify **removed** content
- rendered with ~~strikethrough~~

Text Formatting Elements

- **sub**: subscript
- **sup**: superscript
- e.g. $s_1 + x^2$
- Used in math / chemical formula, footnote reference

Text Formatting Elements

code

- short section of computer code

pre

- preserve whitespace and carriage return

Text Formatting Elements

blockquote

- block quotations
- indented, italicized

q

- inline quotations
- adds quotation marks

Text Formatting Elements

big, small

- increase / decrease text by one size
- not deprecated, but used less

Text Formatting Elements

- **font**: apply inline font syles
- **center**: centers content
- Deprecated; no semantic meaning

Text Formatting Elements

span

- No semantic meaning
- One of the **most useful** elements
- Provide **hook** for you to add *style*, *interactivity* to wrapped text

noscript

- Provide content for browsers with **JavaScript disabled**

```
<noscript>

    <p> Please turn on your JavaScript </p>

</noscript>
```

Form Elements

- User Input
- Allow user to **enter data**, submitted to server for processing

form

- **Container** for form elements
- Block-level element
- Page can have *multiple* forms

form attributes

- **action**: address of form-handling page
- **method**: HTTP method, GET or POST
- **name**: form name, for multiple forms in a page

input

- Different **types**
- **Singletons**: <input type="text" />

Common *input* Attributes

- name
- value
- disabled
- *Others*: accesskey, tabindex

input:text

- Single-line box
- **Short text** data
- Most **common** type
- **Default** type, if unspecified

input:password

- Similar to text
- Characters are **masked** (***, ⋯)
- **Sensitive** info: password, PIN

input:radio

- Can only select **one from choices**
- Same **name** attribute: group choices
- Different *id* for choices

input:checkbox

- Answer **yes/no** (e.g. EULA)
- Set **value** attribute (data)
- No text display, associate **label** element
- Different *id* for choices

input:checkbox

- **Multiple** possible answers
- **name** should end in **[]**

```
<input type="checkbox" name="color[]" value="red" />

<input type="checkbox" name="color[]" value="blue" />
```

input:button

- General purpose **button**
- Use *JavaScript* to add behavior
- **value** attribute = button text

input:submit

- Button that **sends form data** to server
- *Default value*: Submit
- Rename to *meaningful* label (Save, Login, Register)

input:reset

- Button that **clears form data**
- *Destructive* - be careful

input:hidden

- Not visible
- Store value
- Used for passing additional data to server that **doesn't require user input**

input:file

- For **file uploads**
- **accept** attribute = MIME types
- *form* must have attribute: **enctype**="multipart/formdata"

select

- **Dropdown list**
- Pick one, except if *multiple* attribute is set

select vs input:radio

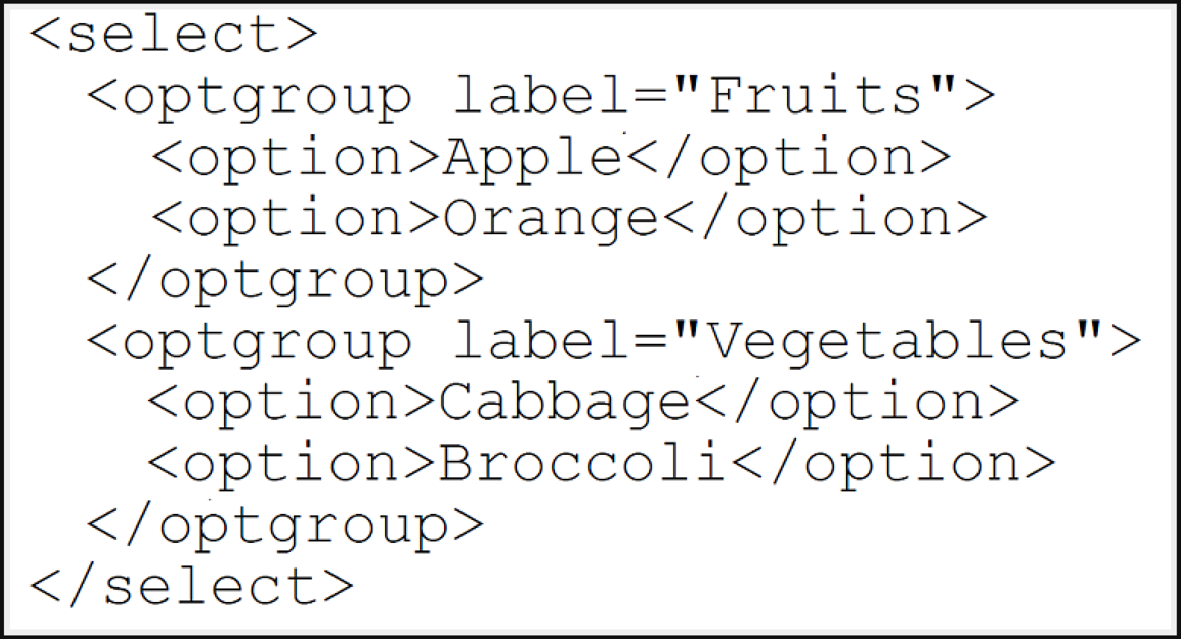
- *input:radio* ⟶-4 choices
- *select &rarr*; many choices (save space)
- *Usability*: **input:radio** is better, can see all choices at once

option

- Option in *select*
- **value** = data sent to server, if selected
- **selected** = makes option currently selected

optgroup

Groups related items in select



```
<select>
  <optgroup label="Fruits">
    <option>Apple</option>
    <option>Orange</option>
  </optgroup>
  <optgroup label="Vegetables">
    <option>Cabbage</option>
    <option>Broccoli</option>
  </optgroup>
</select>
```

textarea

- Similar to input:text
- Allows **multiple lines**
- Has open/close tags, not singleton

label

- Descriptive text for form element
- For radio button, checkbox, textbox, etc
- Improves **usability** & **accessibility**

label

To link labels to input field:

- label's **for** attr = target element's **id**

```
<label for="name">Name:</label>

<input type="text" id="name" />
```

fieldset

- Group related items in a form
- Creates **box** around grouped items
- Can be nested

legend

- Required element for *fieldset*
- Provide **caption text** for grouped items

Table Elements

- Present data in grid-like fashion
- Don't use for layout
- All inside **table** element

Table Elements

- **thead**: groups *header rows*
- **tbody**: groups *data rows*
- **tfoot**: groups **footer rows**
- Use as *hook* to apply CSS to selected rows, to avoid *class-itis*

Table Elements

- *Order*: thead, tfoot, tbody
- Render table header, footer first
- Incrementally add table body rows (esp. if many, or slow connection)

Table Elements

- **tr**: table row
- **th**: table header cell (bold)
- **td**: table data cell

Table Attributes

- **colspan**: cell spans multiple columns
- **rowspan**: cell spans multiple rows

HTML vs XHTML

XHTML

- Extensible **HTML**
- HTML + XML
- aka **valid HTML**

XHTML Structure

- **Required**: html, head, title, body
- **Root**: *html* element
- **Doctype** declaration

XHTML Structure

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict/EN"

"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
```

Doctype

Document Type Definition

- **rules**, **grammar** for a markup version
- *first item* in **valid** web page
- "this is the version I used"
- browser will *try* to **render** content accordingly

Browser Render Modes

Standards

- according to specification
- still depends on browser
- use *strict.dtd*

Almost Standards

- vertical sizing of table cells
- use *loose.dtd*

Browser Render Modes

Quirks

- violate normal web formatting spec
- avoids poor rendering, breaking of pages
- no *doctype*

- Switch *browser render mode* using **doctype**

HTML vs XHTML

- Case sensitivity
- Closing tags, even *singletons*
- Quoted attribute values
- Explicit attribute values
- Element nesting

XHTML > HTML

- **HTML**: lax syntax; sloppy docs
- **XHTML**: strict syntax; clean, standard docs
- W3C XHTML **Validator** (validator.w3.org)

Why use HTML Standards?

- More **interoperable** across different web browsers
- More likely that pages will **display correctly** in the future
- Good practice

HTML Standards

- **IETF**: Internet Engineering Task Force
- **W3C**: World Wide Web Consortium
- **WHATWG**: Web Hypertext Application Technology Working Group

HTML Timeline

HTML 1.0

- Tim Berners-Lee
- Derived from **Standard Generalized Markup Language**
- 1993, first HTML spec → IETF

HTML 2.0

- 1995, IETF
- first standard core HTML features

HTML Timeline

World Wide Web Consortium

- 1994
- Tim Berners-Lee

HTML 3.2

- 1996
- W3C's first Recommendation
- tables, applets, sup, sub

HTML Timeline

HTML 4.0

- 1997
- W3C Recommendation

HTML 4.01

- 1999
- fixed bugs in HTML 4.0
- majority of web pages still use this

HTML Timeline

- Web browsers have **forgiving** parsers
- 99% of pages have at least one error
- **W3C**: HTML → SGML to **stricter XML**

HTML Timeline

XHTML 1.0

- 2000, W3C
- HTML 4 as XML, not SGML

XHTML 1.1

- 2001, W3C
- redefinition of XHTML 1.0

HTML Timeline

- **2004**: Ian Hickson (Opera)
- Proposed extending HTML to allow **web apps** creation
- **Rejected** by W3C
- **XHTML 2** → W3C departs from HTML 4

HTML Timeline

- Mozilla, Apple, Opera reps were **unhappy** with this direction
- **W3C** → *theoretically* pure standards, unrelated to *real needs* of web designers
- They formed **WHATWG** (2004)

WHATWG

- Needs of *real-world authors*
- **Web apps** support
- **Backwards compatibility**: HTML parsers compatible with existing Web content

HTML Timeline

- WHATWG = **HTML 5**, W3C = **XHTML 2**
- Slow adoption of **XHTML 2** by web browsers
- **2006**: TBL admits HTML → XML movement was **not working**

HTML Timeline

- **2007**: W3C asked *HTML Working Group* to adopt work of WHATWG
- **2009**: W3C announced death of XHTML 2

HTML Timeline

HTML5

- 2008, Working Draft
- 2014, W3C Recommendation
- 2016, HTML 5.1
- 2017, HTML 5.2

HTML5

- Support for richer graphics and video
- Structural semantics
- Backwards compatible

HTML5

- <!DOCTYPE html>
- **canvas**: create dynamic images (JS)
- **audio**: embed audio file
- **video**: embed video file
- **meter**: measurement
- **progress**: changing value

HTML5

Web Forms 2.0

- input: autofocus
- input: placeholder
- input: required
- input: datalist

HTML5

Web Forms 2.0

- input: email
- input: range
- input: number
- input: date
- input: color

HTML5

New **Structural** Elements = cure for **div-itis**

- section
- header
- footer
- nav
- article
- figure
- aside

HTML Good Practice

- Tags based on semantic meaning, not appearance
- Don't skip heading order (h1-h6)
- Don't use tables for layout
- Close tags

HTML Good Practice

- Proper nesting
- Indent nested elements
- Separate siblings with blank line
- Be pragmatic

HTML Good Practice

- Comment end of divs and forms
- Meaningful id and class names
- Whitespace Consistency (tabs vs spaces)
- Use debugging tools

Clean, Meaningful Markup

Benefits:

- Easier to **maintain**
- **Search engine** optimization
- Users find **content** easily
- More **accessible** to people with disability

Summary

- HTML
- Structure & Syntax
- HTML Elements
- XHTML vs HTML
- HTML5

References

- *HTML Primer*, Oxford Brookes University, 2003
- *The Ultimate HTML Reference*, Ian Lloyd, 2010
- *Programming the World Wide Web 6E*, R. Sebesta, 2010

References

- *HTML5 for Web Designers*, Jeremy Keith, 2010
- *Web Standards Solutions Special Edition*, Dan Cederholm, 2009

Questions?