

Computer Science & Engineering 120

Learning to Code

Presenting Data I – HTML

Christopher M. Bourke
cbourke@cse.unl.edu

Part I: HTML Basics

Topic Overview

- ▶ HTML Basics
- ▶ Structure of an HTML Document
- ▶ Comments in HTML
- ▶ Special Characters
- ▶ Document Object Model

Introduction to HTML

- ▶ HyperText Markup Language (HTML)
- ▶ Introduced in 1993, de facto document language for the web
- ▶ Regular text document has *content* but no *semantics*
- ▶ HTML *markup* document content with *tags* giving *meaning*

Web browsers can read HTML files and render them into visible or audible web pages. Browsers do not display HTML tags and scripts, but use them to interpret the content of the page. HTML describes the structure of a website semantically along with cues for presentation, making it a markup language, rather than a programming language.

Basic Structure

- ▶ Document begins with a Document Definition Tag:
`<!DOCTYPE html>`
- ▶ Document content is placed inside opening/closing
`<html>...</html>` tag
- ▶ Document contains two main parts: a `<head>` and `<body>`
- ▶ `<head>` contains metadata and resources (scripts, style files)

Example

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>My Home Page</title>
5     <script type="text/javascript" src="jquery.js"></script>
6   </head>
7   <body>
8     <h1>Dr. Chris Bourke</h1>
9     <p>I'm an Assistant Professor of Practice at the
10    <a href="http://unl.edu">University of Nebraska-Lincoln</a>.
11    I enjoy Computer Science and sports.</p>
12    <div id="contactInfo">
13      <ul>
14        <li>Email: <a href="mailto:cbourke@cse.unl.edu">
15          cbourke@cse.unl.edu</a></li>
16        <li>Office: Avery 363</li>
17        <li>Phone: (402) 472-5008</li>
18      </ul>
19    </div>
20  </body>
21 </html>
```

Document Object Model

- ▶ HTML is highly structured
- ▶ Root `<html>` element
- ▶ Each element can be *nested*
- ▶ Forms a tree with a parent-child relation (just like JSON)
- ▶ Known as the *Document Object Model*

Document Object Model

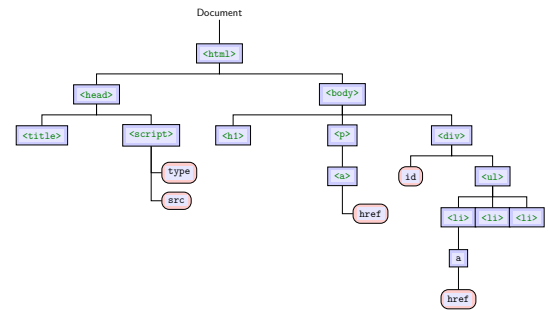


Figure : Document Object Model Tree. Element values have been omitted for clarity.

Comments

- ▶ Comments can be embedded in HTML files
- ▶ Begin/end with `<!-- ... -->`
- ▶ Ignored by the browser when the page is rendered
- ▶ Preserved in the source of the page

```
1 <!-- This is a comment on a single line -->
2
3 <!--
4 This is a multiline
5 comment,
6 neat!
7 -->
```

Special Characters

- ▶ As with JavaScript strings, some characters need to be *escaped*
- ▶ Failure to escape can lead to incorrect rendering
- ▶ Huge list of *HTML Entities* (<http://dev.w3.org/html5/html-author/charref>), but the most important:

Character	HTML
&	&
<	<
>	>
"	"
'	'

Table : Special Characters in HTML

Part II: HTML Tags

Topic Overview

- ▶ Basic HTML Tags
- ▶ Lists
- ▶ Tables
- ▶ Forms
- ▶ Tag Attributes

Basic Tags

Overview

- ▶ An HTML *tag* is denoted with angle brackets and a tag name:
`<tagName>`
- ▶ Every opening tag must be *closed* with a closing tag: `</tagName>`
- ▶ Case insensitive, but standard to use all lower case
- ▶ Dozens of official tags
http://www.w3schools.com/tags/ref_byfunc.asp
- ▶ Many more non-standard tags are used in practice

Basic Tags

Headings

- ▶ 6 levels of *headings* tags are supported
- ▶ Typically used to denote section/subsection titles

```
1 <h1>Main Title</h1>
2 <h2>Subsection Title</h2>
3 <h3>Heading Level 3</h3>
4 <h4>Heading Level 4</h4>
5 <h5>Heading Level 5</h5>
6 <h6>Heading Level 6</h6>
```

Basic Tags

Paragraphs

- ▶ Paragraphs can be denoted using `<p>...</p>`
- ▶ You cannot/should not nest paragraphs

```
1 <p>This is the first paragraph</p>
2
3 <p>This is a second paragraph.</p>
```

Basic Tags

Anchors/Links

- ▶ Anchors are how links are defined, using `<a>`
- ▶ Can link to other pages, external resources, parts of the same page
- ▶ Resource is specified using the `href` *attribute* and a URL
- ▶ Text between opening/closing tag is what is rendered in the page

```
1 <a href="http://cse.unl.edu">CSE Home Page</a>
2
3 <a href="syllabus.pdf">Course Syllabus</a>
```

Basic Tags

Images

- ▶ Images can be displayed in a page using the `` tag
- ▶ The file of the image is specified using the `src` (source) attribute
- ▶ Text is usually not used with an image; the opening/closing tag can be combined by including the forward slash at the end of the opening tag

```
1 
2
3 
```

Lists

- ▶ Lists
- ▶ Unordered (bullet-pointed): ``
- ▶ Ordered (automatically numbered for you): ``
- ▶ Each item is denoted with a list-item `` tag for both types
- ▶ Can nest lists

```
1 <ul>
2   <li>An item</li>
3   <li>Another item</li>
4 </ul>
5 <ol>
6   <li>Item 1</li>
7   <ol>
8     <li>Item 1A</li>
9     <li>Item 1B</li>
10  </ol>
11  <li>Item 2</li>
12 </ol>
```

Tables I

- ▶ Tables can be created using `<table>`
- ▶ Tables have a head, `<thead>` and body, `<tbody>`
- ▶ Rows can be created using `<tr>`
- ▶ Cells can be included using a table-data tag, `<td>`
- ▶ Cells in the head should be done using table-header tags, `<th>`
- ▶ Care must be taken to properly nest/balance tags, ensure that the number of columns is consistent.

Tables II

```
1 <table>
2 <thead>
3 <tr>
4 <th>Course</th>
5 <th>Semester</th>
6 <th>Instructor</th>
7 </tr>
8 </thead>
9 <tbody>
10 <tr>
11 <td>CSCE 120</td>
12 <td>Fall 2015</td>
13 <td>Bourke</td>
14 </tr>
15 <tr>
16 <td>MATH 106</td>
17 <td>Spring 2013</td>
18 <td>Smith</td>
19 </tr>
20 <tr>
21 <td>ENGL 151</td>
22 <td>Fall 2014</td>
23 <td>Jones</td>
24 </tr>
25 </tbody>
26 </table>
```

Forms I

- ▶ Form elements allow users to enter input
- ▶ Basic tag: `<input>`
- ▶ Type of input is indicated by `type` attribute

Basic text box:

```
<input type="text" id="textBox" value=""/>
```

Forms II

Radio Buttons

- ▶ Select one among several options
- ▶ To "group" buttons, give them all the same `name` attribute

```
1 <input type="radio" name="major" value="1">Mathematics</input>
2 <input type="radio" name="major" value="2">Computer Science</input>
3 <input type="radio" name="major" value="3">Accounting</input>
```

Forms III

Checkboxes

- ▶ Select any number of options
- ▶ To "group" buttons, give them all the same `name` attribute

```
1 <input type="checkbox" name="lang" value="1">JavaScript</input>
2 <input type="checkbox" name="lang" value="2">C++</input>
3 <input type="checkbox" name="lang" value="3">Java</input>
```

Forms IV

Select Menu (select an option from a drop down menu)

```
1 <select id="hobby">
2 <option value="wood">Woodworking</option>
3 <option value="crochet">Crochet</option>
4 <option value="art">Painting</option>
5 </select>
```

Forms V

Button (we'll enable functionality later)

```
<button>Compute!</button>
```

Tag Attributes

- ▶ Attributes can be placed inside opening tags
- ▶ Represent key-value pairs
- ▶ Some recognized as standard, any can be defined for apps
- ▶ Value is "assigned" with an equal sign, string
- ▶ Other important *style* attributes: `style`, `class`