# APPENDIX C

# HTML, CSS, XML, and XSL

This appendix is a very brief introduction to two markup languages and their style counterparts. The appendix is intended to give a high-level introduction to these languages for readers of the book. It is not intended to teach how to write documents in these languages, for which a more detailed text is required.

## C.1 HTML

**HyperText Markup Language (HTML)** is a markup language for creating web pages. In this text, when we mention HTML, except where otherwise specified, we mean HTML or XHTML. The differences between the two will be cleared up later. The term *markup language* comes from the book publishing industry; before a book is typeset and printed, a copy editor reads the manuscript and puts marks on it. These marks tell the compositor how to format the text. For example, if the copy editor wants a section of a line to be printed in boldface, she draws a wavy line under that section. Similarly, text and other information for a web page are marked by HTML to be interpreted and displayed by a browser. For easy reading, we have set the markup sections of documents in color.

### C.1.1 HTML Document

To display a document on a browser, we need to create an HTML document. The document should be unformatted (in standard ASCII) text. Most simple text editors such as Windows Notepad or Macintosh TextEdit create unformatted text, but if we use a word processing software, we should save the result as plaintext. We save the file with an *html* extension, for example, **fileName.html.** We can then open the file from any web browser.

### C.1.2 Tags

Tags are the basic element of HTML. Tags are hidden commands that tell the web browsers how to interpret and display text and other content of a web page. Most tags come in pairs: *beginning tag* and *ending tag.*

**<tagName>** … **</tagName >**

Note that the tag name is in lowercase and included inside pointed brackets; the ending tag has an extra slash. The content comes between the beginning and ending tags. For example, the pair **** and **** are bold tags:

**** This text will be displayed in bold. ****

Certain tags do not come in pairs. For example, we only put a **<br/>** tag where we want a line break. Such a tag is called an *empty tag* and has a slash written to the right of the tag name:

**<tagName/>**

Most tags have a set of optional attributes and corresponding values included in the beginning tag:

**<tagName attribute** = value **attribute** = value … **>** content **</tagName>**

### *Doctype*

The doctype declaration is version information that appears as the first line of any HTML document. It refers to a known **Document Type Definition (DTD)** that provides the tags and attributes of an HTML document. In any web page, click the right button and select *View Source*, and you will see the doctype declaration as the first line of the source code, which looks like

**<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"**

**"http://www.w3.org/TR/html4/loose.dtd">**

The following are explanations of different sections of the code:

| | |
|---|---|
| !DOCTYPE | Doctype declaration (uppercase) |
| PUBLIC | DTD is a public resource |
| W3C | Guardian of DTD |
| HTML 4.01: | Markup language and version |
| EN | English |
| http://www.w3.org/TR/html4/loose.dtd | URL for the location of the DTD |

### *Structural Tags*

#### *Head and Title*

The **<head>** tag usually comes after the *doctype declaration*. It is followed by important document information. One of the most important pieces of document information is the document title, which comes between the **<title>** and **</title>** tags. The title has only informational value and is not displayed by the browser, but it is displayed in the browser title bar. The **</head>** tag ends the header section of the document. The following shows an example.

```
<head>
        <title> Title of document goes here. </title>
        Other document information
        …
</head>
```

***Body***

The actual body of an HTML document is enclosed between **<body>** and **</body>** tags. An HTML document is normally organized with the head, title, and body tags as follows:

```
<head>
        <title> Title of document goes here. </title>
        Other document information
        …
</head>

<body>
        The content of the document goes here.
        …
</body>
```

***Heading***

Tags **<hn>** and **</hn>,** with $n = 1, 2, \ldots, 6$, are used to describe the six heading levels in HTML, with h1 the largest and h6 the smallest. Browsers apply a line break after the ending tag.

***Paragraph***

Tag ** ** is used to start a new paragraph; the ending counterpart ** ** is used to end the paragraph. Browsers apply a line break after the ending tag.

***Line break***

The **<br/>** tag, when placed at the end of a line, forces a line break.

***Center Tags***

Tags ** ** and ** ** are used to center a line of a text. Browsers apply a line break after the ending tag.

***Blockquote***

The **<blockquote>** and **</blockquote>** tags are used for marking up a block of text quoted from a person or a source. Normally by default, the text contained within these tags is displayed with left and right indentations. Browsers apply a line break after the ending tag.

***Preserve***

It is important to know that web browsers, unless specified by special tags, only honor the first space and ignore other white spaces such as carriage return and tab. For example, the nutrition fact table typed as

```
Total fat       5 g
Sodium          15 g
Protein         0 g
```

in a web browser appears as:

Total fat 5g Sodium 15 g Protein 0 g

However, we may use the **<pre>** and **</pre>** tags to preserve white spaces (such as spaces, tabs, and line breaks) in a document. For example, we can make the table appear exactly as typed by placing preserve tags before and after the table.

```
<pre>
        Total fat       5 g
        Sodium          15 g
        Protein         0 g
</
pre>
```

***List***

We can define two types of lists: ordered and unordered. The **<ol>** and **</ol>** tags are used for an ordered list; the **<ul>** and **</ul>** tags are used for an unordered list. Each item in either type of list is enclosed inside **<li>** and **</li>** tags. Browsers apply a line break after the **</li>** tag. Items in an ordered list are numbered, while the items in an unordered list are normally bulleted:

HTML text

```
<ol>
        <li> CIS 20 </li>
        <li> CIS 30 </li>
        <li> CIS 40 </li>
<ol>
```

Appearance in a web browser

1. CIS 20
2. CIS 30
3. CIS 40

***Anchor***

The special feature of HTML is hypertext links. The links in an HTML document allow the user to navigate from one document to another document. The **<a>** and **</a>** tags, called anchor tags or link tags, are used to create a link to another web page. One of the attributes used with the **<a>** tag is ***href*** (hyperlink reference), whose value is a URL that indicates the link's destination. For example, the following HTML line creates a link to the Behrouz Forouzan web page at McGraw-Hill Publisher.

**<a href = "http://www.mhhe.com/forouzan" >** Behrouz Forouzan **</a>**

***Image***

We may also include images in a document. The image tag has many attributes. Three are shown in the following tag.

**<img src =** "http://www…" **alt =** "family picture" **align =** middle **/>**

The ***src*** (source) attribute gives the location (URL) where the image is located. The ***alt*** (alternate) attribute defines the text to replace the image if for any reason the image cannot be displayed. The ***align*** attribute defines how the image should be aligned with respect to the text document. An image is not directly embedded in the document; using the above attributes, the browser finds the image and places it where the tag is located.

## Text Formatting

### Bold and Italic versus Strong and Emphasis

The two most common text formatting tags are bold tags, **** and ****, and italic tags, **** and ****. The use of these tags, however, is declining in favor of two almost equivalent tags: the strong tags, **** and ****, and emphasis tags, **** and ****. Just like the bold and italic tags, these tags make the text appear bold or italic respectively but also give semantic meaning to the contained text. Unlike the *bold* and *italic* tags, the *strong* and *emphasis* tags indicate how the corresponding words marked by these tags should be spoken by a speech reader.

### Small and Big

To increase or decrease the font size by one increment, we use **<big>** and **</big>** or **<small>** and **</small>** tags. For example, the HTML text

```
This is <small> smaller </small> but that one is <big> bigger </big>
```

appears as "This is smaller but that one is bigger".

### Other Formatting

Some other common text formatting tags are listed in the following table.

| | | |
|---|---|---|
| Strike | **<strike>** | Strike a line through the text. |
| Subscript | **<sub>** | Move the text half a character up. |
| Superscript | **<sup>** | Move the text half a character down. |
| Underline | **<u>** | Underline the text. |

You may want to use the **<sub>** and **<sup>** tags with *small* tags:

H **<sub>** **<small>** 2 **<small/>** **<sub/>** O appears as $H_2O$

### Advisory Tags

Four important advisory tags are abbreviation tags (**<abbr>** and **</abbr>**), acronym tags (**<acronym>** and **</acronym>**), definition tags (**<def>** and **</def>**), and cite tags, (**<cite>** and **</cite>**). All four have title attributes and a similar format:

**<tagName title** = "string"**> </tagName>**

For example, the following abbreviation tags show that DTD is an abbreviation for Document Type Definition.

**<abbr title =** "Document Type Definition"**> DTD </abbr>**

In a browser, when we put the curser over the contained text (DTD in our example) the title (Document Type Definition) is displayed (usually in a tool-tip).

### Nesting

We may use two or more tags in nested form. For example, we can nest italic tags inside bold tags:

** ** This text is in italic bold ** **

Make sure to nest them correctly. The following is the wrong format:

  Wrong Format  

The order of nesting can change the appearance of the text. For example, compare the two nested expressions.

| HTML Text | Appearance in a web browser |
|---|---|
|   First  Second  | ***First* Second** |
|   First  Second  | ***First* *Second*** |

### C.1.3 XHTML

The **Extensible HyperText Markup Language (XHTML)** is almost identical to HTML 4.01 but it also conforms to the restricted syntax of XML. This compliance makes XHTML a structured markup language. A document marked up with XHTML will be a "well-formed" document and, consequently, will be interpreted and displayed by browsers the way the author intended. Some of the most important requirements of XHTML are:

- Elements must be properly nested.
- Elements must always be closed: Normal tags such as paragraph tags  and  must have the beginning and ending tags; the empty tags such as line break must be written as <br/> and not as <br>. The slash after *br* indicates closing of the element.
- Elements and attributes must be in lowercase.
- Attribute values must be quoted.
- Documents must have three main parts: DOCTYPE declaration, head section, and body section.

## C.2 CSS

Logically, a simple web document is made of two layers: content and presentation. Although it is possible to keep both layers together, separating them increases flexibility, reduces repetition, and increases efficiency. **Cascading Style Sheets (CSS)** were created to separate the document content from document presentation. We may apply styles to elements of an HTML document in three ways: *in-line, internal,* or *external*.

### C.2.1 In-line Style

We may specify a style to an individual element of an HTML document. For example, the following makes the font size of the contained paragraph 90% and the font color blue.

```
<p style = "font-size: 90%; color: blue" >
The size of the font is 90% and blue
</p>
```

### C.2.2 Internal Style Sheet

If we want to specify style rules that apply to a single HTML document, we can enclose the style sheet between **<style>** and **</style>** tags in the head section of the HTML document (such as **body**, **h1**, and so on). The general format of the style sheet rule is

**HTML content** {**attribute**: value; **attribute**: value; …}

Note that each attribute is separated from its value by a colon. Attributes are separated by semicolons, and the entire attribute block is placed inside curly brackets {}. For example, the following internal style sheet applies rules to heading 1 and the body of the document.

```
<head>
        <title> Internal Style sheet </title>
                <style type = "text / css" >
                h1{font-family: mono space; color: green}
                body {font-family: cursive; color: red}
                <style>
</head>
<body>
…
<body>
```

### C.2.3 External Style Sheet

To create an external style sheet, we create a text document and place all the desired style rules for each part of the HTML content in that document and save the document with *css* extension: **fileName.css.** The following is an example of such a document:

```
body {font-size: 10 pt; font-family: Times New Roman; color:
     black; margin-left: 12 pt; margin-right: 12 pt; line-height;
     14 pt}

p {margin-left: 24 pt; margin-right: 24 pt}
h1 {font-size: 24 pt; font-family: Book Antiqua; color: red}
h2 {font-size: 22 pt; font-family: Book Antiqua; color: red}
…
h6 {font-size: 12 pt; font-family: Book Antiqua; color: red}
…
a: link {color: red}
a: visited {color: blue}
…
```

Next we link this style sheet to any HTML document by including a **<link/>** tag in the head section of that document:

**<link rel =** "style sheet" **type =** "text/css" **href =** "URL" **/>**

The ***rel*** (relationship) attribute says that the reference document is a style sheet. The ***type*** attribute identifies the MIME type of the linked resource (text/css) and the ***href*** attribute gives the URL address of the css file.

## C.3 XML

The **Extensible Markup Language (XML)** is a language that allows users to define a representation of data or data structure and assign values to each field in the structure. In other words, XML is a language that allows us to define mark-up elements (our own tags and our own document structure) and create customized markup language. The only restriction is that we need to follow the rules defined in XML. For example, the following shows how we can define a student record with three fields: *name, id,* and *birthday.*

```
<?xml version = "1.0"?>
    <student>
        <name> George Brown </name>
        <id> 2345 </id>
        <birthday> 12- 08 - 82 <birthday>
    <student>
```

This is similar to a struct or class in languages like C, C++, or Java.

## C.4 XSL

The data defined and initialized to values in an XML document needs another language, a style language, to indicate how the document should be displayed. The **Extensible Style Language (XSL)** is the style language of XML, just as CSS is the style language of HTML and XHTML.