Web Authoring

Difference between HTML and XHTML

[1]HTML5 has two parsing modes or syntaxes: HTML and XML. The difference depends on whether the document is served with a Content-type: text/html header or a Content-type: application/xml+xhtml header.

If it's served as text/html, the following rules apply:

- Start tags are not required for every element.
- End tags are not required for every element.
- Only void elements such as br, img, and link may be "self-closed" with />.
- Tags and attributes are case-insensitive.
- Attributes do not need to be quoted.
- Some attributes may be empty (such as checked and disabled).
- Special characters, or entities, do not have to be escaped.
- The document must include an HTML5 DOCTYPE.

HTML Syntax

Let's look at another HTML5 document.

```
<!DOCTYPE html>
 <html>
  <head>
   <meta charset=utf-8>
   <title>Hi</title>
   <!--
   This is an example of a comment.      The lines below show how to include CSS
   -->
   <link rel=stylesheet href=style.css type=text/css>
   <style>
    body{
     background: aliceblue;
    }
   <style>
  </head>
  <body>
   <p>
    <img src=flower.jpg alt=Flower>
    Isn't this a lovely flower?
   <p>
    Yes, that is a lovely flower. What kind is it?
   <script src=foo.js></script>
  </body>
</html>
```

Again, our first line is a DOCTYPE declaration. As with all HTML5 tags, it's case-insensitive. If you don't like reaching for Shift, you could type < !doctype html> instead. If you really enjoy using Caps Lock, you could also type < !DOCTYPE HTML> instead.

[1] http://www.sitepoint.com/web-foundations/differences-html-xhtml/

Next is the head element. The head element typically contains information about the document, such as its title or character set. In this example, our head element contains a meta element that defines the character set for this document. Including a character set is optional, but you should always set one and it's recommended that you use UTF-8.

Our head element also contains our document title (). In most browsers, the text between the title tags is displayed at the top of the browser window or tab.

Comments in HTML are bits of text that aren't rendered in the browser. They're only viewable in the source code, and are typically used to leave notes to yourself or a coworker about the document. Some software programs that generate HTML code may also include comments. Comments may appear just about anywhere in an HTML document. Each one must start with .

A document head may also contain link elements that point to external resources, as shown here. Resources may include style sheets, favicon images, or RSS feeds. We use the rel attribute to describe the relationship between our document and the one we're linking to. In this case, we're linking to a cascading style sheet, or CSS file. CSS is the stylesheet language that we use to describe the way a document looks rather than its structure.

We can also use a style element (delineated here by

and) to include CSS in our file. Using a link element, however, lets us share the same style sheet file across multiple pages.

By the way, both meta and link, are examples of void HTML elements; we could also self-close them using />. For example, would become , but it isn't necessary to do this.

"XHTML5": HTML5's XML Syntax

HTML5 can also be written using a stricter, XML-like syntax. You may remember from Chapter 1 that XHTML 1.0 was "a reformulation of HTML 4 as an XML 1.0 application." That isn't quite true of what is sometimes called "XHTML5". XHTML5 is best understood as HTML5 that's written and parsed using the syntax rules of XML and served with a Content-type: application/xml+xhtml response header.

The following rules apply to "XHTML5":

- All elements must have a start tag.
- Non-void elements with a start tag must have an end tag (p and li, for example).
- Any element may be "self-closed" using />.
- Tags and attributes are case sensitive, typically lowercase.
- Attribute values must be enclosed in quotes.
- Empty attributes are forbidden (checked must instead be checked="checked" or checked="true").
- Special characters must be escaped using character entities.

Our html start tag also needs an xmlns (XML name space) attribute. If we rewrite our document from above to use XML syntax, it would look like the example below.

```
<!DOCTYPE html>
 <html xmlns="http://www.w3.org/1999/xhtml">
  <head>
   <meta charset="utf-8" />
   <title>Hi</title>
  </head>
  <body>
```

```
    <p>
      <img src="flower.jpg" alt="Flower" />
      Isn't this a lovely flower?
    </p>
    <script src="foo.js" />
  </body>
</html>
```

Here we've added the XML name space with the xmlns attribute, to let the browser know that we're using the stricter syntax. We've also self-closed the tags for our empty or void elements, meta and img. According to the rules of XML and XHTML, all elements must be closed either with an end tag or by self-closing with a space, slash, and a right-pointing angle bracket (/>).

In this example, we have also self-closed our script tag. We could also have used a normal tag, as we've done with our other elements. The script element is a little bit of an oddball. You can embed scripting within your documents by placing it between script start and end tags. When you do this, you must include an end tag.

However, you can also link to an external script file using a script tag and the src attribute. If you do so, and serve your pages as text/html, you must use a closing tag. If you serve your pages as application/xml+xhtml, you may also use the self-closing syntax.

Don't forget: in order for the browser to parse this document according to XML/XHTML rules, our document must be sent from the server with a Content-type: application/xml+xhtml response header. In fact, including this header will trigger XHTML5 parsing in conforming browsers even if the DOCTYPE is missing.

As you may have realized, XML parsing rules are more persnickety. It's much easier to use the text/html MIME type and its looser HTML syntax.