

INTERNATIONAL UNIVERSITY OF JAPAN
Public Management and Policy Analysis Program
Graduate School of International Relations

ADC5030401 (2 Credits)
Introduction to Electronic Government
 Winter 2019

This document introduces XHTML and summarizes basic elements (tags) of XHTML. Since all pieces of information of electronic government are ultimately presented in markup languages, understanding of these languages is a basic, but essential part of e-government and e-commerce. Despite the availability of HTML5 as the new Web standard, this document discusses XHTML1.0.

1. HTML, XML, XHTML, CSS

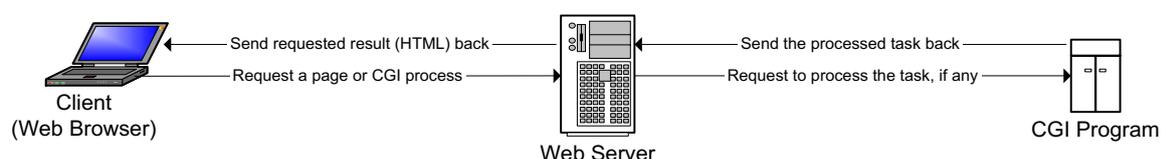
A Markup language consists of elements (tag) with their attributes, character-based data (letters and numbers), character entity references (e.g., &#amp;#38;), and others. They are text scripts used to generate various documents. Examples are LaTeX, HTML, and XHTML. Unlike WYSIWYG (What You See Is What You Get) software packages (e.g., WordPerfect and LibreOffice), a markup language uses tags to control formats and styles (e.g., font, color, and layout) of the content.

HTML (Hypertext Markup Language), based on SGML (Standard Generalized Markup Language), is to publish hypertext on the World Wide Web (Web). It was first proposed by Tim Berners-Lee in the early 1990s and its specification has evolved up to version 5. World Wide Web Consortium (W3C) announced HTML5 as a new Web standard in October 2014.

XML (Extensible Markup Language) is a “simple, very flexible text format derived from SGML” to structure various documents and textual data and thus improve simplicity, generality, and usability. Each data item is arranged in a systematic way like in a database. Instead, scripts seem a bit messier than HTML and XHTML.

XHTML (Extensible HTML) (version 1.0) is reformulation of HTML 4.01 in XML, and combines the strength of HTML 4 with the power of XML” (W3C). Its first version was recommended in 2000 and version 2.0 in 2010. Version 2.0 does not have 100 percent backward compatibility. HTML 5 (5.1 in 2016) introduced new tags (elements), such as header, footer, article, main, section, canvas, nav, audio, and video and attributes, eliminating deprecated tags such as applet, center, font, and frame.

CSS (Cascading Style Sheets), according to Wikipedia, is a style sheet language used to describe the presentation semantics of a document written in a markup language. It enables the *separation of document content from document presentation*, including elements such as the layout, colors, and fonts, improving accessibility and flexibility of the document.



2. How Can a Web Browser Read a HTML/XHTML Document?

When you click a hyperlink on a Web browser, the browser sends a request for the hyperlinked content to the Web server designated in the hyperlink. The Web server interprets the request and finds (runs CGI programs if requested) the requested contents (including related text, audio, video, etc.) and, sends them back to your Web browser (if the content is not found, the server will send an error code). If your request involves CGI (Common Gateway Interface) process, the Web server asks the designated CGI program to process requested task and provides necessary information that was passed through the Web browser (you). The Web server receives the output processed by the CGI program and then sends it (Web document) back to the Web browser. See the above Figure.

Once the document is sent back correctly, your Web browser interprets HTML/XHTML and displays formatted contents on the screen. This process is called *rendering* and this task is performed by a rendering engine or layout engine of your Web browser.

Given a variety of Web browsers (e.g., Mozilla Firefox, Safari, Google Chrome, and Internet Explore), there are numerous rendering engines depending on vendors. Each engine has its own settings (default color, font, layout, and others) and rendering method. Therefore, a HTML/XHTML document may appear different depending on the layout engine that a Web browser has.

Therefore, a HTML/XHTML document may appear different depending on the layout engine that a Web browser has. Mozilla and Firefox, produced by the same foundation, can correctly interpret HTML/XHTML as recommended by W3C, while Internet Explore (MSIE6) cannot (intentionally interprets in a different way).

To make it worse, Microsoft created its own tags that are not compatible with W3C standards, generating accessibility problems and being blamed severely. But most Web browsers like Safari and Google Chrome support W3C standards. It is important to generate HTML/XHTML documents compliant with W3C standards in order for accessibility and advanced services.

3. XHTML Script versus Word Document

What is the difference between a XHTML script and a Word document? Contents remain unchanged but the way of “markup” does not. In a word processor, all contents are written in WYSIWYG (What You See Is What You Get). By contrast, all formats including fonts, sizes, colors, and layouts are described by tags in XHTML.

Therefore, a Word document can be rewritten in HTML, XML, XHTML, and LaTeX to get the equivalent (not exactly the same due to default settings) result. Compare the left (MS-Word document) and right (XHTML document) below. All formats in the right are controlled by elements (tags) begging with “<” and ending with “>”

- `<!DOCTYPE...>` classifies the type of a document and specifies a markup language used.
- `<html>...</html>` declares the beginning and end of the XHTML text.
- `<head>...</head>` declares the beginning and end of the head information (including title, abstract, author, key words, and date of last modification)
- `<body>...</body>` declares the beginning and end of the content.

- `<h1>...</h1>`, `<h2>...</h2>`... are used for heading
- `<p>...</p>` indicates a paragraph

<p><u>Research paper</u></p> <p style="text-align: center;">Internet History</p> <p style="text-align: center;">Hun Myoung Park <i>January 25, 2013</i></p> <p>This paper summarize...</p> <p>1. Introduction</p> <p>The Internet became...</p> <p>...</p> <p>7. Conclusion</p> <p>...</p>	<pre> <!DOCTYPE ... > <html> <head> <title>Internet History</title> <meta name="description" content="This paper ..." /> <meta name="author" content="Hun Myoung Park" /> <meta name="revised" content="January 25, 2013" /> </head> <body> <h1>1. Introduction</h1> <p>The Internet became... </p> ... <h1>7. Conclusion</h1> <p>... </p> ... </body> </html> </pre>
---	---

4. Basic XHTML Rules

- 4.1 Tags (or element) are building blocks of XHTML/HTML.
- 4.2 Be well-formed and structured (nested box in a box)
- 4.3 Most tag have their opening and ending tags. Exceptions include `<meta ... />`, `
`, and ``.
- 4.4 All elements should be closed with a corresponding closing tag or `/>`
- 4.5 Each element may have zero, one, or multiple attributes and their values. For example, `<element attribute="value(s)" attribute="value(s)"... >...</element>` The element name and its attributes are separated a single space.
- 4.6 Always quote attribute values.
- 4.7 Use lowercase
- 4.8 Use character entities references: `&` (&), `>` (>), `≤` (≤), `©` (©)
- 4.9 Multiple spaces are equivalent to single space.
- 4.10 Comments: `<!-- ... -->` that is ignored by a Web browser when rendering.

5. Brief Classification of Elements (Tags)

- 5.1 Metainformation Module: `meta`
- 5.2 Structure Module: `body`, `head`, `html`, `title`
- 5.3 Text Module: `p`, `br`, `code`, `div`, `span`, `em`, `strong`, `h1`, `h2`, `h3`
- 5.4 List Module: `ol`, `ul`, `li`, `dl`, `dt`, `dd`
- 5.5 Tables Module: `table`, `thead`, `tbody`, `tfoot`, `tr`, `th`, `td`, `caption`
- 5.6 Hypertext Module: `a`
- 5.7 Image Module: `img`
- 5.8 Link Module: `link`
- 5.9 Object Module: `object`
- 5.10 Forms Module: `form`, `input`, `label`, `select`, `optgroup`, `option`, `textarea`

6. Basic Elements

6.1 Document type definition `<!DOCTYPE... />` declares what type of markup language is used.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

6.2 The root tag `<html>...</html>` indicates beginning and end of html document.

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

6.3 Head element `<head>...</head>` contains information about the Web document. This element allows title, meta, link, style, script, and other elements.

6.4 Title element `<title>...</title>` contains the title of the Web document.

6.5 Meta element `<meta... />` contains the information of the document. This element defines a property (e.g., author and keyword) and its value. The followings are example of the meta tag.

```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8;" />
<meta name="author" content="Hun Myoung Park;" />
<meta name="description" content="Personal Web Site; " />
<meta name="keyword" content="Policy Modeling; SAS; Stata; R;" />
<meta name="revised" content="01/20/2013" />
```

6.6 In the first meta element example above, `charset` (character set) specifies the type of character encoding that a document contains and how Web browsers should display those characters. In UNICODE, `utf-8` is recommended regardless of languages.

6.7 Link element `<link .../>` links an external files such as CSS and script files.

```
<link rel="main_css" type="text/plain" href="../site/egov.css ">
```

6.8 Style element `<style>...</style>` declares embedded cascading style sheets to be used in the Web document.

```
<style type="text/css">
p {font:10pt arial; text-indent:1em;}
</style>
```

6.9 Script element `<script>...</script>` specifies a script language like javascript to be used in the Web document.

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

6.10 Body element `<body>...</body>` contains all “renderable” html tags.

7. Text Elements (Paragraph)

7.1 A paragraph is represented by `<p>...</p>`.

- 7.2 White space(s) represent a blank; Consecutive white spaces are considered as a blank.
If you want to have more than one blank, add the reference entity ` `;
- 7.3 Empty elements do not have any content between opening and ending tags.
Oftentimes they have an opening tag with a closing slash in it. `
` represent a line break, whereas `<hr />` represents a horizontal line.
- 7.4 `` and `` are used for emphasis.
- 7.5 Headings are represented by `<h1>...</h1>`, `<h2>...</h2>`...

8. Grouping Elements: `<div>` and ``

Both `<div>... </div>` and `...` add structure to the document. `<div>` creates a logical division between what comes before and what comes after. Span element `...` is used to control a part of paragraph. `<div>` is a block-level element, while `` is an inline element that wraps up a phrase, a word, or a character without carriage return and/or line feed.

```
<p>He was a famous <span style="font-style:italic;">Congressman</span> who
initiated ...</p>
```

```
<div>
<h3>Chapter 1.</h3>
<p>This chapter discusses...</p>
...
</div>
```

`<div>` is useful for lay-outting without `<table>`, which is least recommended in layout. The `position` attribute has five properties of static, absolute, relative, fixed, and inherit. See <http://www.wickham43.net/divboxes.php> for examples of `position`. Some attribute values of the attribute `position` do not work in Microsoft Internet Explore (Javascript or other programs are needed at the expense of speed and compatibility)

```
<div style="position:absolute; left:500px; top:0px; width:240px; height:40px;">
<div style="position:relative padding:2px; border-bottom: 1px dotted #9e032a;">
<div style="position:fixed left:...;">
```

9. List Elements

- 9.1 Unordered list `` displays items with bullet characters rather than serial numbers.

`` denotes list item.

Syntax	Example	Output
<code></code>	<code></code>	<ul style="list-style-type: none"> • Apple • Orange • Grape
<code>...</code>	<code>Apple</code>	
<code>...</code>	<code>Orange</code>	
<code>...</code>	<code>Grape</code>	
<code></code>	<code></code>	

- 9.2 Ordered list `` displays items with serial numbers or ordered characters.

Syntax	Example	Output
<code></code>	<code></code>	<ol style="list-style-type: none"> 1. Apple 2. Orange 3. Grape
<code>...</code>	<code>Apple</code>	
<code>...</code>	<code>Orange</code>	
<code>...</code>	<code>Grape</code>	
<code></code>	<code></code>	

9.3 Definition list `<d1>` displays items in a formatted manner. `<dt>` and `<dd>` respectively denote title and definition.

Syntax	Example	Output
<code><d1></code>	<code><d1></code>	XHTML
<code><dt>...</dt></code>	<code><dt>XHTML</dt></code>	
<code><dd>...</dd></code>	<code><dd>A Web language</dd></code>	A Web language
...		
<code></d1></code>	<code></d1></code>	

10. Anchor Elements

10.1 Anchor `<a>` links to a hyperlinked document (e.g., a text document and video/audio clip). Its usage is `...`. The hypertext reference (`href=""`) attribute specifies the URL or address of the hyperlinked document.

10.2 URL can be either absolute address or relative address. An *absolute address* contains the full Internet address (URI or URL) including protocol used to request. A *relative address* points to the desired document's location relative to the location of the current document (where you are). Relative addresses are commonly used to points to internal links since it is efficient and easy to manage.

10.3 In order to move up to a parent directory, use `..` as in `../images/iuj.jpg` (move up from the current directory and then get down to images directory to get iuj.jpg). `../../index.html` moves up to the parent directory and again its parent directory, and then points to index.html. `pmis/pmis2013.pdf` gets down to the child directory `pmis` and then reads `pmis2013.pdf`.

10.4 Labeling elements in a document link specific parts of the document using the `id=""` attribute. This identification attribute plays a bookmark of a book. Add the attribute to the place where you want to remember. And provide `"#..."` as a URL in `href=""` (e.g., `...`).

11. Image Elements

11.1 Image element `` links to a specific image file whose formats are gif, jpeg, or png.

11.2 An example is ``

11.3 What is the `alt=` (alternate attribute) for? That is used in the anchor tag as well. In table, `summary=` is used to describe the table. This `alt` attribute provides the content to be displayed when the designated image is not available. Therefore, alternative attribute is very important for improving accessibility. This attribute, of course, is used in other anchors.

11.4 You can control the image using such attributes as `width=""` and `height=""`

```

```

12. Table Element

This element displays data and other information in a tabular format in a HTML/XHTML document. But `<table>` is least recommended for layouting a Web page. Its header, table body, and table footer tags are often complicated and need many `<td>` tags.

12.1 Table element `<table>...</table>` constructs a table. `<table>` has `border`, `width`, `cellspacing`, and `cellpadding` attributes to control the layout.

```
<table border="0" cellspacing="1" cellpadding="0" width="600" style="...;">
```

12.2 Caption element `<caption>...</caption>` provides a caption of the table.

12.3 A table consists of table header `<thead>...</thead>`, table body `<tbody>...</tbody>`, and table footer `<tfoot>...</tfoot>`.

12.4 Table row `<tr>...</tr>` defines a row of table header, footer, and body.

12.5 Table cell contain table data `<td>...</td>` or table header `<th>...</th>`.

12.6 Table cells can span multiple rows and columns. Use column span (`colspan`) and row span (`rowspan`) attributes in a table cell.

```
<td colspan="2"> <!-- spans two columns in a cell -->
<td rowspan="3"> <!-- spans three rows in a cell -->
```

12.7 Major `<table>` attributes include border, cellspacing, cellpadding, and width.

```
<table border="0" cellspacing="1" cellpadding="0" width="600" style="...;">
```

Example	Output															
<pre><table border="1"> <thead> <caption>Fruit Sale</caption> <tr> <th>Fruit</th> <th>Price</th> <th>Sale</th> </tr> </thead> <tbody> <tr> <td>Apple</td> <td>\$2</td> <td>\$500</td> </tr> <tr> <td>Orange</td> <td>\$3</td> <td>\$900</td> </tr> </tbody> <tfoot> <tr> <td colspan="2"></td> <td>\$1,400</td> </tr> </tfoot> </table></pre>	<table border="1"> <thead> <tr> <th colspan="3">Fruit Sale</th> </tr> <tr> <th>Fruit</th> <th>Price</th> <th>Sale</th> </tr> </thead> <tbody> <tr> <td>Apple</td> <td>\$2</td> <td>\$500</td> </tr> <tr> <td>Orange</td> <td>\$3</td> <td>\$900</td> </tr> <tr> <td colspan="2"></td> <td>\$1,400</td> </tr> </tbody> </table>	Fruit Sale			Fruit	Price	Sale	Apple	\$2	\$500	Orange	\$3	\$900			\$1,400
Fruit Sale																
Fruit	Price	Sale														
Apple	\$2	\$500														
Orange	\$3	\$900														
		\$1,400														

13. Uploading XHTML documents to the Web Server

Generally a Web server is located remotely and you need to transmit your XHTML documents to an (your) account in the Web server. Most Web servers like Apache are UNIX/Linux machines and require secured connection (e.g., secured FTP) for transmission. WinSCP (for Windows) and Samba in Finder (for Mac) are commonly used.

After uploading the documents, you need to adjust the permission of the documents. For general XHTML documents, permission type 644 is commonly used (document owners can read and write, while groups and others can read the documents only. Microsoft's Internet Information Services (IIS) for Windows is quite accessible and does not require permission adjustment; in tern, IIS is not as secure as UNIX servers.

14. Conclusion

You learn basics and core elements of XHTML and have to understand technical implications of Web standards (e.g., `alt=""` attribute) for the accessibility of usability of electronic government.

End of document.