Unit 13 XHTML – 1

Structure:

13.1 Introduction

In previous unit, you have learnt JavaScript, it's more traditional programming language, and it writes a series of instructions to compute kindof action. HTML is the basic code for websites; it can utilize JavaScript to docustomized web page. In this unit you are going to learn XHTML, XHTML rules, benefit of XHTML, framework tags, text format tags and image formats.

Objectives:

After studying this unit, you will be able to:

- describe benefits of XHTML
- define meta tags
- describe text format tags
- describe formats of Images

13.1.1 Introduction to XHTML

"XHTML" stands for *Extensible Hypertext Markup Language*. XHTML is the modern reformulation of HTML using the rules of XML (Extensible Markup Language). The rules of XML are more consistent than the rules of HTML.

One key issue when using XHTML is based on HTML. In HTML, it was common to specify where the browser placed an element in a web page or the fonts and colors used to display an element. XHTML allows only a document's content and structure to appear in a valid XHTML document, and not it's formatting. XHTML can be an upgraded and stricter version of HTML 4.01, while HTML 5 is an expansion and clarification of the markup language for the Web. Neither XHTML nor HTML 5 is practically difficult to write. However, there are some important techniques to writing properly formed code.

There are some simple and most important rules are:

- Tags must all be in lower case.
- Elements must be nested correctly.
- Tag elements must be closed.
- Documents must be well-formed

When marking up documents for the Web, you are performing a very similar process, except you do it by adding things called *tags* to the text. With XHTML the key thing to remember is that you are adding the tags to indicatethe *structure* of the document, which part of the document is a heading, which parts are paragraphs, what belongs in a table, and so on. Browsers such as Internet Explorer will use this markup to help present the text in a familiar fashion, similar to that of a word processor (headings are bigger than the main text, there is space between each paragraph, lists of bullet points have a circle in front of them). However the way these are presented is up to the browser.

The XML Declaration

Sometimes you will see something that is known as the XML Declaration at the beginning of an XHTML document. The XHTML language was actually written using another language called XML (Extensible Markup Language, which is used to create markup languages), and any XML document can begin with this optional XML declaration:

<? xml version="1.0" encoding="UTF-8" ?>

If you include the XML declaration, it must be right at the beginning of the document; there must be nothing before it, not even a space. The encoding attribute indicates the encoding used in the document.

An encoding (short for character encoding) represents how a program or operating system stores characters that you might want to display. Because different languages have different characters, and indeed because some programs support more characters than others, there are several different encodings.

13.1.2 Benefits of XHTML

The benefits of adopting XHTML or migrating your existing sites to the new standards are many. Document developers and user agent designers are constantly discovering new ways to express their ideas through new markup. In XML, it is relatively easy to introduce new elements or additional element attributes. The XHTML family is designed to accommodate these extensions through XHTML modules and techniques for developing new XHTML-conforming modules (described in the XHTML Modularization specification). These modules will permit the combination of existing and new feature sets when developing content and when designing new user agents.

A well written XHTML page is more accessible than an old style HTML page, future browser versions might stop supporting elements deprecated elements from old HTML drafts, and so many old basic HTML sites may start displaying incorrectly and unpredictably.

Some of the benefits of XHTML are:

Simple doctype: XHTML supports doctype definition.

Strict tradition: Most developer uses the XHTML strict standard that make the code easier to read and more predictable.

Validation support: validation is very useful code, it eliminates goofy coding mistakes. With help of JavaScript, XHTML supports this validation code.

Self Assessment Questions

1. XHTMLstands for_____.
2. XML stands for_____.
3. An_____represents how a program or operating system stores characters that you might want to display.

13.2 XHTML Page Frame Work Tags

According to the HTML standard, tag and attribute name are not case-sensitive. There is no difference in effect between <head>, <Head>, <HEAD> or even <HeadD>, they are all equivalent. With XHTML case is important. All current standard tag and attribute names are in lowercase.

The four main elements that form the basic structure of every document:

<html>, <head>, <title>, and <body>. These four elements should appear in every XHTML document.

- **The <html> Element**

The <html> element is the containing element for the whole XHTML document. After the optional XML declaration and required DOCTYPE declaration, each XHTML document should have an opening <html> tag and each document should end with a closing </html> tag.

If you are writing Strict XHTML 1.0, the opening tag must also include something known as a *namespace identifier* (this indicates that the markup in the document belongs to the XHTML 1.0 namespace).

Example: the opening tag should look like this:

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

While it is not strictly required in Transitional XHTML documents, it is a good practice to use it on all XHTML documents.

Only two elements appear as direct children of an <html> element: <head> and <body> (although the <head> and <body> elements will usually contain many more elements).

- **The <head> Element**

The <head> element is just a container for all other header elements. It should be the first thing to appear after the opening <html> tag.

Each <head> element should contain a <title> element indicating the title of the document, although it may also contain any combination of the following elements, in any order:

<base> tag specifies the base URL/target for all relative URLs in a document. There can be at maximum one <base> element in a document, and it must be inside the <head> element.

<object>, which is designed to include images, JavaScript objects, Flash animations, MP3 files, QuickTime movies, and other components of a page.

<link>, is used to link to an external file, such as a style sheet or JavaScriptfile.

<style>, is used to include CSS rules inside the document;

<script> is used for including script in the document.

<meta>, which includes information about the document such as keywords and a description, which are particularly helpful for search applications

- **The <title> Element**

You should specify a title for every page that you write. It exists inside the <title> element (which, as you learnt earlier in the HTML, is a child of the <head> element). It is used in several ways:

- At the very top of a browser window.
- As the default name for a bookmark in browsers such as IE, Firefox, and Safari.
- By search engines that use its content to help index pages.

Therefore, it is important to use a title that really describes the content of your site. For example, the homepage of our site should not just say “Home Page”; rather it should describe what your site is about. For example, rather than just saying Wrox Home Page, it is more helpful to write:

<title>Wrox: Books for programmers written by programmers</title>

The test for a good title is whether a visitor can tell what user will find on that page just by reading the title, without looking at the actual content of the page.

The <title> element should contain only the text for the title; it may not contain any other elements.

- **The <body> Element**

The <body> element appears after the <head> element and contains the part of the web page that you actually see in the main browser window, which is sometimes referred to as *body content*. It may contain anythingfrom a couple of paragraphs under a heading to more complicated layouts containing forms and tables, and is likely to constitute the majority of any

XHTML document. Most of tags what you will be learning in this and the following unit will be written between the opening <body> tag and closing </body> tag.

The <body> element may carry all of the attributes from the *attribute groups* you are about to meet in the next section. If you are using Transitional XHTML or HTML 4.1, you can use any of the deprecated attributes on the <body> element. XHTML and HTML 5 require that some basic rules of syntax be followed. As with the tag rules discussed above, XHTML and HTML 5 syntax is not much more difficult that correctly formed HTML 4.0.

These rules are, however, much stricter and must not be violated. These rules follow.

- Attributes must be quoted.
- The Name attribute is replaced by the ID attribute
- Attribute shorthand must not be used.
- DOCTYPE statement must be used

13.2.1 Meta Tag

Meta tag live in the <head> rather than the <body> of a document and contains information about a document. The information can be used for a number of purposes including helping search engines index your site, specifying the author of a document, and if the document is time-sensitive, specifying when the page should expire.

The <meta> element is an empty element and so does not have closing tag. Instead, <meta> elements carry information within attributes, so you need a forward slash character at the end of element.

The <meta> element can take eight attributes, four of which are universal attributes dir, lang, xml and title. The other four, however, are specific to the <Meta>element: schema, name, content, and http-equiv.

The name and content attributes tend to be used together, as do the http-equiv and content. Use meta tags to make your web pages more accessible to search engines and web spiders. However, you should be careful to keep the keywords and description concise. Many search engines have "spamming" rules, allowing no more than a small amount of characters or keywords.

<meta > Usage:

- Define the description of the page for search engines.
  <meta name="description" content="This is the description page.">
- Define when the content on the page expires (a MIME header).
  <meta http-equiv="expires" content="thu,31 DEC 2013 00:04:00 PST">

13.3. 2 DOCTYPE statements

The DOCTYPE statement is part of Document Type Definition, DTD, and is used to specify which syntax is used in the Web page.

The Doctype statement has the following five components:

<! DOCTYPE html public/system DTD identifier URLof DTT>

1. It must start with an opening <! Which indicates it is an XML declaration, followed by the keyword DOCTYPE.
2. The type of document, this must be html for XHTML document.
3. A keyword follows which must be either PUBLIC, indicating that the DTD is publicly available or SYSTEM which indicates that it is not publicly available.
4. The identifier of the DTD.
5. Finally the URL of the DTD.

The only thing allowed to appear before a document type declaration in an XHTML document is the XML declaration. The !DOCTYPE declaration, should then be immediately followed by the opening <html> tag. The

! DOCTYPE declaration, which looks like the following

<! DOCTYPE html PUBLIC "../w3c/DTD/XHTML/EN"
"http://www/w3/org/xhtml-strict.dtd>

When writing transitional XHTML document, you do not need to include a !DOCTYPE declaration, although it is good practice to include one.

Self Assessment Questions

4. The_____tag should contain only the text for the title; it may not contain any other elements.
5. _____tag can be used to make your web pages more accessible to search engines and web spiders.

13.3 Text Format Tags

You've seen the skeleton structure of an XHTML document and the core attributes, so it is now time to get back to looking at how you markup text in order to describe its structure. Because almost every document you create will contain some form of text, the elements you are about to meet are the fundamental building blocks of most pages.

While going through this section it is important to remember that, while one browser might display each of these elements in a certain way, another browser could display very different results; the font sizes (and therefore the amount of space a section of text takes up) will change between browsers, as will the typefaces used.

In this section, you learn how to use what are known as basic text formatting elements:

- **Heading elements (h1, h2, h3, h4, h5, h6)**
- **Paragraph elements (p, br, pre)**

If you want people to read what you have written, then structuring your text well is even more important on the Web than when writing for print. People have trouble reading long, wide paragraphs of text on web sites unless they are broken up well, so getting into good habits from the start of your web development career will help ensure that your pages get the attention they deserve. Before you get started on the elements that you will use to mark up your text, it helps to know how text is displayed by default (it is up to you to tell the browser if you want it to treat text differently).

- **White Space and Flow**

Before you start to mark up your text, it is best to understand what XHTML does when it comes across spaces and how browsers treat long sentences and paragraphs of text.

You might think that if you put several consecutive spaces between two words, the spaces would appear between those words onscreen, but this is not the case; by default, only one space will be displayed. This is known as white space collapsing. Similarly, if you start a new line in your source document, or you have consecutive empty lines, these will be ignored and simply treated as one space, as will tab characters.

example look at the following paragraph

This paragraph shows how multiple spaces between words are treated as a single space. This is known as white space collapsing, and the big spaces between some of the words will not appear in the browser. It also demonstrates how the browser will treat multiple carriage returns (new lines) as a single space, too.

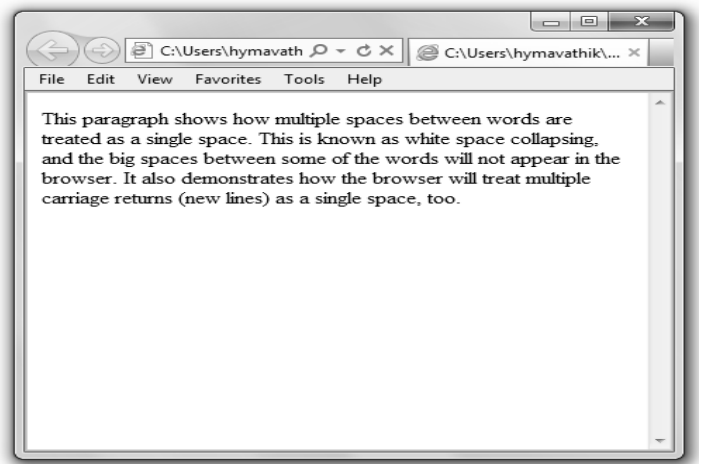

Figure 13.1: Web page for white space flow

As you can see in figure, the browser treats the multiple spaces and several carriage returns (where text appears on a new line) as if therewere only one single space.

13.3.1 Headings and paragraph text

- **Creating Headings**

No matter what sort of document you are creating, most documents have headings in some form or other. Newspapers use headlines, a heading on a form tells you the purpose of the form, the title of a table of sports results tells you the league or division the teams play in, and so on.

In longer pieces of text, headings can also help structure a document. If you look at the structure of contents for this unit, you can see how differentlevels of headings have been arranged to add structure to the book, with subheadings under the main headings.

XHTML offers six levels of headings, which use the elements **<h1>, <h2>, <h3>, <h4>, <h5>, and <h6>.**

While browsers can display headings differently, they tend to display the <h1> element as the largest of the six and <h6> as the smallest, CSS can be used to override the size and style of any of the elements.

- **Creating Paragraphs Using the Element**

The element offers another way to structure your text. Each paragraph of text should go in between an opening and closing tag.

Example:

Here is a paragraph of text.

Here is a second paragraph of text.

Here is a third paragraph of text.

When a browser displays a paragraph, it usually inserts a new line before the next paragraph and adds a little bit of extra vertical space

The element can carry all of the universal attributes and the deprecated align attribute.

- **Creating Line Breaks Using the
 Element**

Whenever you use the
 element, anything following it starts on the next line. The
 element is an example of an empty element, where you do not need opening and closing tags, because there is nothing to go in between them.

The
 element has a space between the characters br and theforward slash. If you omit this space, older browsers will have trouble rendering the line break, whereas if you miss the forward slash character and just use
, it is not valid XHTML.

Most browsers allow you to use multiple
 elements to push text down several lines, and many designers use two line breaks betweenparagraphs of text rather than using the element to structure text, as follows:

Paragraph one

Paragraph two

Paragraph three

- **Creating Preformatted Text Using the <pre> Element**

Sometimes you want your text to follow the exact format of how it is written in the XHTML document you don't want the text to wrap onto a new line

when it reaches the edge of the browser; you don't want it to ignore multiple spaces; and you want the line breaks where you put them.

Any text between the opening <pre> tag and the closing </pre> tag will preserve the formatting of the source document. You should be aware, however, that most browsers would display this text in a monospaced fontby default. (Courier is an example of a monospaced font, because each letter of the alphabet takes up the same width. In non-monospaced fonts, an*i* is usually narrower than an *m*.)

Two of the most common uses of the <pre> element are to display tabular data without the use of a table (in which case you must use the monospaced font or columns will not align correctly) and to represent computer source code.

For example, the following shows some JavaScript inside a <pre> element

```
<pre>
function testFunction(strText){
alert (strText)
}
</pre>
```

By using above code, the content of the <pre> element is displayed in the monospaced font.

- **Presentational Elements**

If you use a word processor, you are familiar with the ability to make text bold, italic, or underlined; these are just three of the ten options available to indicate how text can appear in HTML and XHTML. The elements are bold, italic, underlined, strong,em, teletype, larger, smaller, superscripted, and subscripted text.

Technically speaking, these elements affect only the presentation of a document, and the markup is of no other use, but they remain in both Transitional and Strict XHTML 1.0. As you will discuss now.

There are dedicated elements for indicating things like emphasis within a piece of text, and these will result in a similar presentation of the information.

All of the following presentational elements can carry the universal attributes and the UI event attributes.

- **The Element**

Anything that appears in a element is displayed in **bold**, like the word bold here:

The following word uses a bold typeface.

This does not necessarily mean the browser will use a bold face version of a font. Some browsers use an algorithm to take a font and make the lines thicker (giving it the appearance of being bold), while others (if they cannot find a boldface version of the font) may highlight or underline the text.

This element has the same effect as the element, and is used to indicate that its contents have strong emphasis.

- **The Element**

The content of an element is displayed in italicized text, like the word italic here:

The following word uses an italic typeface.

This does not necessarily mean the browser will look for an oblique or italicized version of the font. Most browsers use an algorithm to put the lines on a slant to simulate an italic font.

The element has the same effect as the element and which is used to indicate that its contents have emphasis.

- **The <u> Element (deprecated)**

The content of a <u> element is underlined with a simple line:The following word would be <u>underlined</u>

The <u> element is deprecated in HTML 4 and XHTML 1.0, although it is still supported by current browsers.

- **The <s> and <strike> Elements (deprecated)**

The content of an <s> or <strike> element is displayed with a strikethrough, which is a thin line through the text (<s> is just the abbreviated form of <strike>).

The following word would have a <s>strikethrough</s>.

Both the <s> and <strike> elements are deprecated in HTML 4.1 and Transitional XHTML 1.0, and were removed from Strict XHTML 1.0, although they are still supported by current browsers.

- **Special characters**

When making up text, certain characters or symbols (eg: less than <) may be difficult to embed directly into an XHTML document. Some keyboards do not provide these symbols, or the presence of these symbols may cause syntax errors.

Example: if x <10 then inceremnt by x by 1 <?P>

Result in a syntax error because it uses the less-than charater(<), which is reserved for start tags and end tags such as <P> and </P>. XHTML provides character entity references (in the form &code ;) for representing special character.

We could correct previous line by writing

Example : if x <10 then inceremnt by x by 1 <?P>

- **Grouping Elements with and **

The and elements allow you to group several elements to create sections or subsections of a page. On their own, they will not affect the appearance of a page, but they are commonly used with CSS to allow you to attach a style to a section of a page.

For example, you might want to put all of the footnotes on a page within a element to indicate that all of the elements within that element relate to the footnotes. You might then attach a style to this element so that they appear using a special set of style rules.

The element is used to group block-level elements:

Example:

<h2>Footnotes</h2>

1 The World Wide Web was invented by Tim Berners-Lee

2 The W3C is the World Wide Web Consortium who maintains

```
many Web standards</p>
</div>
```

The element, on the other hand, can be used to group inline elements only. So, if you had a part of a sentence or paragraph you wanted to group, you could use the element. Here you can see that I have added a element to indicate which content refers to an inventor. It contains both a bold element and some text:

Example:

```
<div class="footnotes">
<h2>Footnotes</h2>
<p><span class="inventor"><b>1</b> The World Wide Web was invented by Tim Berners Lee</span></p>
<p><b>2</b> The W3C is the World Wide Web Consortium who maintains many Web standards</p>
</div>
```

On its own, this would have no effect at all on how the document looks visually, but it does add extra meaning to the markup, which now groups the related elements. This grouping can either be used by a processing application, or can be used to attach special styles to these elements using CSS rules.

13.3.2 Images

Browsers tend to support three common bitmap graphics formats, and most graphics programs will save images in these formats:

- **GIF:** Graphics Interchange Format (pronounced either "gif" or "jif")
- **JPEG:** Joint Photographic Experts Group Format (pronounced "jay peg")
- **PNG:** Portable Network Graphics (pronounced "ping" or "pee en gee")

GIF (Graphic Interchange Format)

GIF images are created using a palette of up to 256 colors and each pixel of the image is one of these 256 colors. Every different GIF image can have a different palette of 256 colors selected from a range of over 16 million colors. The program that saves the image also selects the palette that will best represent the images.

GIFs do have another handy feature: you can specify one or more colors in

a GIF to represent a *transparent background* in parts of the image that are the specified colors, the background will be allowed to show through.

JPEG (Joint Photographic Experts Group Format)

The JPEG image format was developed as a standard for storing and *compressing* images such as photographs with wide ranges of colors. When you save a JPEG, you can usually specify by how much, if at all, you want to compress the image which depends upon the image quality you want.

The process of compressing a JPEG involves discarding color data that people would not normally perceive, such as small color changes. However, because the image format discards this data when the image is compressed, some of the data is lost and the original cannot be recreated from a compressed version – hence it is known as *lossy compression*.

PNG (Portable Network Graphics)

The PNG formats was designed for the same uses as GIF images, but while it was being created the designers decided to solve what they thought were some of the disadvantages with the GIF format. The result is two types of PNG. The 8-bit PNG has the same limitations as an 8-bit GIF only 256 colors, and when transparency is used each pixel is either on or off. Then there is the enhanced PNG-24, a 24-bit version, which has the following advantages:

- The number of colors available for use in an image is not restricted, and so any color can be included without losing any data.
- A map (like the lookup table that indicates the color of each pixel in GIFs) is used to provide different levels of transparency for every pixel, which allows for softer, anti-aliased edges.
- The approach of sampling one in eight lines was replaced with a two-dimensional sample which can display an image eight times faster than a GIF.
- PNG 24-bit files can contain gamma correction information to allow for slight differences in color between different monitors and platforms.

Furthermore, all PNGs tend to compress better than a GIF equivalent.

- **Adding Images Using the <img> Element**

Images are usually added to a site using the <img> element. It must carry the src attribute indicating the source of the image and an alt attribute whose value is an alternate description for the image in case it does not load or the user has a visual impairment.

For example, the following line would add the image called logo.gif into the page (in this case, the image lives in a directory called images, and this images directory resides inside the same directory that holds the XHTML file).

- **The height and width Attributes**

The height and width attributes specify the height and width of the image: height=“120” width=“180”

The values for these attributes are almost always shown in pixels.

- ***The border Attribute (deprecated)***

The border attribute specifies the width of the border around the image in pixels: border=“2”

If the attribute is not used, there will not be a border unless the image is used as a link, in which case you could specify border= “0”

- **<Object> Element**

The W3C introduced the <object> element into XHTML/HTML4 with the intention that it could be used to embed all media types into documents, not just graphics but also MP3 files, Flash movies, QuickTime movies, JavaScript objects, Java applets, and so on. It is even intended that in the long run the <object> element be used to include images in documents.

Before the <object> element was introduced, a range of elements was used to insert multimedia objects into pages, such as the <applet>, <embed> and <bgsound> elements, but these elements have been deprecated.

While we are used to browsers supporting GIFs, JPEGs, and, more recently, PNGs, the same cannot be said of including MP3 audio files, Flashmovies, QuickTime movies, or Java applications. Rather, in these cases, the <object> element is used to include some other kind of software that is used to play or load these files. For example:

- Flash movies are played with the Flash Player;
- Windows Media Files require Windows Media Player;
- MP3s can be played in various players including Flash Player, Windows Media Player, and QuickTime Player.

So when it comes to embedding audio, video, or Java/JavaScript programs on your web page, you not only need to have the file, but you also have to choose an application to embed into your page that will play/run the file.

Self Assessment Questions

6. Whenever you use the_____element, anything following it starts on the next line.
7. The_____element, can be used to group inline elements only.

13.4 Summary

- XHTML is the modern reformulation of HTML using the rules of XML.
- XHTML can be the latest version of HTML and that there are three different flavors of XHTML— in order to tell the browser which you are using, you can use a DOCTYPE declaration.
- You've seen how every XHTML document should contain at least the <html>, <head>, <title>, and <body> elements, and how the <html> element should carry a namespace identifier.
- You have covered following text format tags:
  - The six levels of headings: <h1>, <h2>, <h3>, <h4>, <h5>, and <h6>
  - Paragraphs <p>, preformatted sections <pre>, and line breaks
.
  - Presentational elements <b>, <i>, <u> , etc.
- Images are usually added to a site using the <img> element. It must carry the src attribute indicating the source of the image.

13.5 Terminal Questions

1. What is XHTML? Explain advantages of XHTML over HTML?
2. Describe page framing tags of XHTML?
3. Explain mandatory Head tag elements?
4. Explain how to display information in paragraph form and explain different text tags available in XHTML?
5. List and explain available image formats?

13.6 Answers

Self Assessment Questions

1. Extensible Hypertext Markup Language

2. Extensible Markup Language
3. Encoding
4. Title
5. Meta tags
6. br
7.

Terminal Questions

1. The rules of XML are more consistent than the rules of HTML.
   In HTML, it was common to specify where the browser placed an element in a web page or the fonts and colors used to display anelement. XHTML allows only a document's content and structure to appear in a valid XHTML document, and not it's formatting. For more details refer section 13.1. 2
2. The four main elements that form the basic structure of every document: <html>, <head>, <title>, and <body>. These four elements should appear in every XHTML document. For more detail refer section 13.2
3. Each <head> element should contain a <title> element indicating the title of the document, although it may also contain some more elements. For more detail refer section 13.2
4. Every document you create will contain some form of text, the elements you are about to meet are the fundamental building blocks of most pages. For more details refer section 13.3.1
5. Browsers tend to support three common bitmap graphics formats, and most graphics programs will save images by using those formats. For more details refer section 13.3.2

13.7 References

- Gosselin Don (2010) *principles of html xhtml and dhtml*. Eleventh edition.
- Hart Davis (2009). *HTML, XHTML and CSS quick step*s. McGraw-Hill Education.
- Gary Rebholz (2003). *How to use HTML and XHTML.* Preston Gralla Que Publishing, Eighth edition.