**Structuring Documents for the Web**

We need to add structure to a document so that web browser can present the web page to the people who visit your site in a way they will understand.

Imagine an article that contains a headline (or title) and several paragraphs of text; if you wanted to put this article on the web, you need to add structure to the words in the document so that the browser known which words are the headline and where each paragraph starts and ends.

**A web of structured documents:**

Take a moment to look at the printed information we see every day and how it compares to what we see on the web.

Every day you come across all kinds of printed documents – newspapers, train time tables, bank forms. You can think of the web as being a sea of documents that all link together. Similar to the printed documents that you meet every day in life.

**Introducing HTML and XHTML:**

XHTML:

* Stands for Extensible Hyper Text Markup Language
* Similar to HTML
* Stricter than HTML

Major differences from HTML:

* DOCTYPE is mandatory
* <html>, <head>, <title> and <body> are mandatory
* Elements must be properly nested and be closed.

Example:

<title>……………...</title>

<body>……………. </body>

* All the elements should be in lowercase and one root element (root element are called elements.) In html, the root element is the <html> element.

HTML:

* Stands for Hypertext Markup Language
* It is a tag-based language used to create web pages
* Created by Berners – Lee in the late 1991
* Current version is HTML 5.3
* Developed with an intent of defining the structure of documents like headings, paragraphs, lists etc.
* HTML is sent from server to client’s browser whenever HTTP request is sent to server

**Basic Syntax/Skeleton:**

<html>

<head>

<title> This is my first document </title>

</head>

<body>

<h1> This is a heading </h1>

<p> This is paragraph </p>

</body>

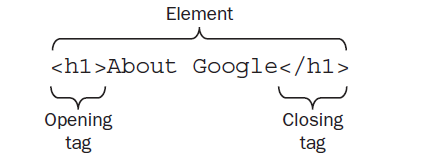
</html>

**Difference between HTML and XHTML:**

|  |  |  |
| --- | --- | --- |
| Basis | HTML | XHTML |
| Stands for | Hyper Text Markup Language | Extensible Hyper Text Markup Language |
| Introduction | HTML is the main markup language for creating web page and other information that can be distinguished in web browser. | XHTML is a family of XML markup language that mirror or extend versions of the widely used HTML, the language in which web pages are written. |
| File extension | .html, .htm | .xhtml, .xht, .xml, .html, .htm |
| Internet media type | text/html | application/xhtml + xml |
| Type of format | Document file format | Markup Language |
| Extended From | SGML | XML, HTML |
| Application | Application of standard Generalized Markup Language | Application of XML |
| Function | Webpages are written in HTML | Extended version of HTML that is stricter and XML-based |
| Nature | Flexible framework requiring lenient HTML specific parser | Restrictive subset of XML and needs to be parsed with standard XML parser |
| Origin | HTML was created by Tim Berners-Lee | (W3C) World Wide Web Consortium Recommendation in 2000. |
| Version | Current version is HTML 5.3 | XML 1.1 |

**Tags and Elements:**

|  |  |
| --- | --- |
| Tags | Elements |
| A tag is made up of left & right-angle bracket and letters & numbers between those brackets. | Whereas elements are the opening and closing tags plus anything between the two tags. |



**Q. Why HTML is called markup language?**

Hypertext means machine readable text and Markup means to structure it in a specific format. So, HTML is called Hypertext Markup Language because it is a language that allows users to organize, improve the appearance and link text with data on the internet.

HTML defines the structure and layout of a web document by using a variety of tags and attributes. The HTML syntax is based on a list of tags that describe the page format and what is displayed on the web page.

Further, it is called markup language because it marks the information by tagging them.

Example:

<p> Hello </p>

Here, it is marking the word “hello” by <p> paragraph tag. So, that browser will know how to display it.

**DOCTYPE:**

In HTML, the DOCTYPE declaration (short for "Document Type Declaration") is an instruction to the web browser about which version of HTML the document is written in.

The DOCTYPE declaration is placed at the beginning of an HTML document, before the <html> tag. Its purpose is to tell the browser how to interpret the document, as different versions of HTML may have different rules for syntax and features.

***For example, the DOCTYPE declaration for HTML5 is:***

<!DOCTYPE html>

This tells the browser that the document is written in HTML5 and should be rendered accordingly.

**Prior to HTML5,** the DOCTYPE declaration was more complex and included information about the document type, version, and DTD (Document Type Definition).

***For example, the DOCTYPE declaration for HTML 4.01 Transitional was:***

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

This DOCTYPE declaration specifies the DTD (loose.dtd) and the version of HTML (HTML 4.01 Transitional). The DTD defines the rules for the syntax and structure of the document, and tells the browser how to interpret it.

In HTML5, the DOCTYPE declaration is simplified to just <!DOCTYPE html>, as the HTML5 specification includes a built-in default DTD and does not require a separate reference to an external DTD.

**Separating Heads from Bodies:**

Whenever you write a web page in HTML, the whole of the page is contained between the opening <html> and closing </html> tags, just as it was in the last example. Inside the <html> element, there are two main parts to the page:

* **The** <head> **element**: Often referred to as the head of the page, this contains information about the page (this is not the main content of the page). For example, it might contain a title and a description of the page, or instructions on where a browser can find CSS rules that explain how the document should look. It consists of the opening <head> tag, the closing </head> tag, and everything in between.
* **The** <body> **element**: Often referred to as the body of the page, this contains the information you actually see in the main browser window. It consists of the opening <body> tag, closing </body> tag, and everything in between.

Together, the <html>, <head>, and <body> elements make up the skeleton of an XHTML document — they are the foundation upon which every web page is built.

Example:

<head>

<title> Popular website: Google </title>

</head>

Nested HTML Elements:

HTML elements can be nested (elements can contain elements). All HTML documents consist of nested HTML elements.

Example:

<html>

<body>

<p> Web programming is <b>easy</b> </p>

</body>

</html>

**Note:** The <head> element contains information about the document, which is not displayed within the main page itself. The < body > element holds the actual content of the page that is viewed in your browser.

Empty HTML Elements:

HTML elements with no content are called empty elements. <br> is an empty element without a closing tag (the <br> tag defines a line break)

Example:

<p> This is a <br> paragraph with a line break </p>

HTML is not case sensitive:

HTML tags are not case sensitive. <P> means same as <p>.

**Learning from Others by Viewing Their Source Code:**

When HTML first came out, a lot of people learned how to create pages by using a very handy feature that you’ll find in most common browsers — the ability to look at the source code that made the page.

If you go to the View menu in your browser, and then look for an option that says View Source or Page Source, you should be able to see the code that created the page.

If you want to see how the author of a page achieved something on a page, this can be a very handy technique.

You can view or inspect any web page source code through following command: ctrl + u or ctrl + shift + I in google chrome.

**Attributes Tell Us About Elements:**

Attributes are used to say something about the element that carries them, and they always appear on the opening tag of the element that carries them. It provides additional information about HTML elements.

Example:

<a href="https://www.google.com"> Click here to visit google website </a>

All attribute is made up of two parts: a name and a value:

* The name is the property of the element that you want to set. In above example, the <p> element carries an attribute whose name is title, which you can use to indicate where the link should take you.
* The value is what you want the value of property to be. In this example, the value was the URL of the site that the link should take you to, so the value of the “href” attribute is <https://www.google.com>
* The value of the attribute should always be put in double quotation marks, and it is separated from the name by the equal sign. If you wanted the link to open in new window, you could add a target attribute to the opening <a> tag as well, and give it a value of “\_blank”:

<a href=”<https://www.google.com>” target=”\_blank”> Click here to visit google website </a>

**Elements for Marking Up Text:**

You now know that an HTML page (also sometimes referred to as an HTML document) is made up of elements that describe how its content is structured. Each element describes what you will find between its opening and closing tags. The opening tags can also carry attributes that tell you more about that particular element.

You’ ve already seen that every XHTML document starts off using the <html>, <head>, and <body> elements. You will come back to look at these elements in more detail near the end of the chapter, at which point you will also meet some attributes called universal attributes (so called because they can appear on every element). But I want to get you building pages as quickly as possible, so you’ re going to spend most of the remaining part of this chapter learning the different elements you can use to describe the structure of text:

* The six levels of headings: <h1>, <h2>, <h3>, <h4>, <h5>, and <h6>
* Paragraphs <p>, preformatted sections <pre>, line Breaks <br>, and addresses <address>
* Presentational elements: <b>, <i>, <u>, <s>, <tt>, <sup>, <sub>, <strike>, <big>, <small>, and <hr>
* Phrase elements such as <em>, <strong>, <abbr>, <acronym>, <dfn>, <blockquote>, <q>, <cite>, <code>, <kbd>, <var>, <samp>, and <address>
* Lists such as unordered lists using <ul> and <li>, ordered lists using <ol> and <li>, and definition lists using <dl>, <dt>, and <dd>
* Editing elements such as <ins> and <del>
* Grouping elements: <div> and <span>

That may sound like a lot of elements, but I hope you’ ll be surprised at how quickly we can move through them.