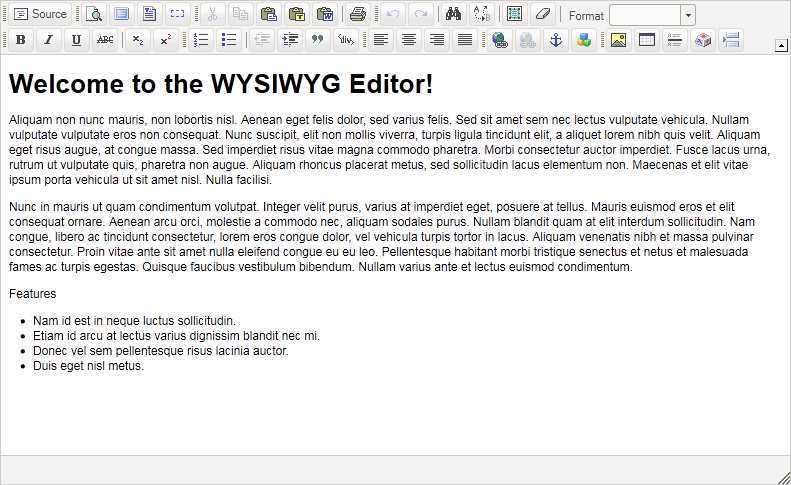
## **Unit - 4**

## **What is a WYSIWYG editor?**

A WYSIWYG (pronounced "wiz-ee-wig") editor or program is one that allows a developer to see what the end result will look like while the interface or document is being created.



WYSIWYG is an acronym for "what you see is what you get". A WYSIWYG editor can be contrasted with more traditional editors that require the developer to enter descriptive codes (or markup) and do not permit an immediate way to see the results of the markup. The first true WYSIWYG editor was a word processing program called Bravo. Invented by Charles Simonyi at the Xerox Palo Alto Research Center in the 1970s, it became the basis for Simonyi’s work at Microsoft and evolved into two other WYSIWYG applications called Word and Excel.

**Introduction to Dreamweaver**

Dreamweaver – what is it? Adobe Dreamweaver is a software application that allows you to create and develop Web sites. Dreamweaver is considered WYSIWYG (What You See Is What You Get), meaning that when you format your Web page, you see the results of the formatting instead of the mark-ups that are used for formatting. HTML is not WYSIWYG, whereas Microsoft Word is WYSIWYG. However, Dreamweaver allows you to hand code HTML as well. Dreamweaver also supports CSS and JavaScript as well as other languages including ASP and PHP. Dreamweaver makes it easy to upload your entire Web site to a Web server. You can also preview your site locally. Dreamweaver also lets you create templates for your Web site that you can use again and again by modifying certain unrestricted areas within the template. Then if you want to change one particular part of your Web site (the logo changes, a main link changes), you only have to modify the template for the changes to propagate throughout your Web site.

**Website Creation and maintenance**

Websites are created using a markup language called HTML. **Web** designers build webpages using HTML tags that define the content and metadata of each **page**. The layout and appearance of the elements within **a webpage** are typically defined using CSS, or cascading style sheets.

Website maintenance is the act of regularly checking your website for issues and mistakes and keeping it updated and relevant. This should be done on a consistent basis in order to keep your website healthy, encourage continued traffic growth, and strengthen your SEO and Google rankings.

Keeping a website well maintained and attractive is important to companies big and small in order to engage and retain customers. It’s easy for businesses, especially startups, to cut corners and let a few tasks slide.

**Web Hosting**

A web hosting service is a type of Internet hosting service that allows individuals and organizations to make their website accessible via the World Wide Web.

Web hosting is a service that allows organizations and individuals to post a website or web page onto the Internet. ... Websites are hosted, or stored, on special computers called servers. When Internet users want to view your website, all they need to do is type your website address or domain into their browser.

**XML**

XML stands for **E**xtensible **M**arkup **L**anguage. It is a text-based markup language derived from Standard Generalized Markup Language (SGML).

XML tags identify the data and are used to store and organize the data, rather than specifying how to display it like HTML tags, which are used to display the data. XML is not going to replace HTML in the near future, but it introduces new possibilities by adopting many successful features of HTML.

**Features of XML**

Extensible markup language takes the best features of standard generalized markup language and combines it with some features of hyper text mark up language and adds rest of the features from successful applications of both. Extensible markup language takes its major frame work from standard generalized mark up language.  
  
It is commonly called subset of standard generalized mark up language but in technical terms it is an application profile of standard generalized markup language where as hyper text mark up language uses standard generalized markup language and is an application of standard generalized markup language, extensible markup language is just standard generalized markup language on a very small scale. Extensible markup language inherits the use of web address from hyper text mark up language.

|  |  |  |
| --- | --- | --- |
| **No.** | **HTML** | **XML** |
| 1) | HTML is used **to display data** and focuses on how data looks. | XML is a software and hardware independent tool used **to transport and store data**. It focuses on what data is. |
| 2) | HTML is a **markup language** itself. | XML provides a **framework to define markup languages**. |
| 3) | HTML is **not case sensitive**. | XML is **case sensitive**. |
| 4) | HTML is a presentation language. | XML is neither a presentation language nor a programming language. |
| 5) | HTML **has its own predefined tags**. | You **can define tags according to your need**. |
| 6) | In HTML, it is **not necessary to use a closing tag**. | XML **makes it mandatory to use a closing tag**. |
| 7) | HTML is **static** because it is used to display data. | XML is **dynamic** because it is used to transport data. |
| 8) | HTML **does not preserve whitespaces**. | XML **preserve whitespaces**. |

## XML Naming Rules

XML elements must follow these naming rules:

* Element names are case-sensitive
* Element names must start with a letter or underscore
* Element names cannot start with the letters xml (or XML, or Xml, etc)
* Element names can contain letters, digits, hyphens, underscores, and periods
* Element names cannot contain spaces

Any name can be used, no words are reserved (except xml).

## The Building Blocks of XML Documents:

The XML documents, when viewed from a DTD point of view, are made up by the below-listed building blocks:

* Elements
* Attributes
* Entities
* PCDATA
* CDATA

## Elements:

The main building blocks of both XML and HTML documents are the elements. The text, or other elements, can be included in an element or it can be empty.

****Example: HTML elements:****

|  |
| --- |
| "body" and "table" |

****Example: XML elements:****

|  |
| --- |
| "note" and "message" |

****Example: empty HTML elements:****

|  |
| --- |
| "hr", "br" and "img" |

****Example:****

|  |
| --- |
| <body>hello world</body><message>how are you</message> |

## Attributes:

To provide extra information about elements, the Attributes are used. They are always placed inside the opening tag of an element and always come in name/value pairs.

****Example:****

|  |
| --- |
| <img src="book.gif" /> |

****Explanation:****

In the above example, the “img” element has additional information about a source file. Here, “img” is the name of the element, “src” is the name of the attribute and “book.gif” is the value of the attribute. The element is closed by a ” /” because it itself is empty.

## Entities:

In XML, special meanings are attached to some characters. For instance, the start of an XML tag is defined by the less-than sign (<). The HTML entity: “&nbsp;” is a “no-breaking-space” entity and is used to insert an extra space in a document in HTML. When a document is parsed by an XML parser, the entities are expanded.

****The predefined entities in XML:****

|  |  |
| --- | --- |
| ****Entity References**** | ****Character**** |
| &lt; | < |
| &gt; | > |
| &amp; | & |
| &quot; | “ |
| &apos; | ‘ |

## PCDATA:

The text data that will be parsed by the XML parser is also termed as Parsed Character Data (PCDATA). Usually, all the text in an XML document is parsed by the XML parsers. The text between the XML tags is also parsed if an XML element is parsed. The parser examines the text for entities and markup. The tags inside the text will be treated as markup. The entities will be expanded. The &, <, or > characters should not be included in a parsed character data. The &amp; &lt; and &gt; entities, respectively, are used to represent them.

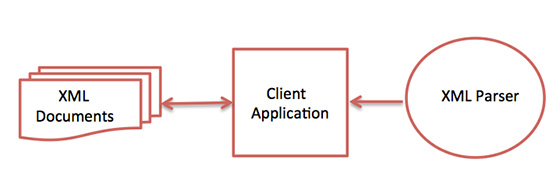
## CDATA:

The text data that should not be parsed by the XML parser is also termed as CDATA or Character Data. The tags inside the text will NOT be treated as markup. The entities will not be expanded.

**XML Parsar**

**XML parser** is a software library or a package that provides interface for client applications to work with XML documents. It checks for proper format of the XML document and may also validate the XML documents. Modern day browsers have built-in XML parsers.

Following diagram shows how XML parser interacts with XML document −



The goal of a parser is to transform XML into a readable code.

To ease the process of parsing, some commercial products are available that facilitate the breakdown of XML document and yield more reliable results.

Some commonly used parsers are listed below −

**MSXML (Microsoft Core XML Services)** − This is a standard set of XML tools from Microsoft that includes a parser.

**System.Xml.XmlDocument** − This class is part of .NET library, which contains a number of different classes related to working with XML.

**Java built-in parser** − The Java library has its own parser. The library is designed such that you can replace the built-in parser with an external implementation such as Xerces from Apache or Saxon.

**Saxon** − Saxon offers tools for parsing, transforming, and querying XML.

**Xerces** − Xerces is implemented in Java and is developed by the famous open source Apache Software Foundation.

**XML Example:**

**XML file :**  
Creating Books.xml as :-

//UTF-8: Universal Text Format-8bits

|  |
| --- |
| <?**xml** version="1.0" encoding="UTF-8"?>  <?**xml-stylesheet** type="text/css" href="Rule.css"?>  <**books**>      <**heading**>Welcome To GeeksforGeeks </**heading**>      <**book**>          <**title**>Title -: Web Programming</**title**>          <**author**>Author -: Chrisbates</**author**>          <**publisher**>Publisher -: Wiley</**publisher**>          <**edition**>Edition -: 3</**edition**>          <**price**> Price -: 300</**price**>      </**book**>      <**book**>          <**title**>Title -: Internet world-wide-web</**title**>          <**author**>Author -: Ditel</**author**>          <**publisher**>Publisher -: Pearson</**publisher**>          <**edition**>Edition -: 3</**edition**>          <**price**>Price -: 400</**price**>      </**book**>      <**book**>          <**title**>Title -: Computer Networks</**title**>          <**author**>Author -: Foruouzan</**author**>          <**publisher**>Publisher -: Mc Graw Hill</**publisher**>          <**edition**>Edition -: 5</**edition**>          <**price**>Price -: 700</**price**>      </**book**>      <**book**>          <**title**>Title -: DBMS Concepts</**title**>          <**author**>Author -: Navath</**author**>          <**publisher**>Publisher -: Oxford</**publisher**>          <**edition**>Edition -: 5</**edition**>          <**price**>Price -: 600</**price**>      </**book**>      <**book**>          <**title**>Title -: Linux Programming</**title**>          <**author**>Author -: Subhitab Das</**author**>          <**publisher**>Publisher -: Oxford</**publisher**>          <**edition**>Edition -: 8</**edition**>          <**price**>Price -: 300</**price**>      </**book**>  </**books**> |

**CSS FILE :**  
Creating Rule.css as:-

|  |
| --- |
| books {  **color**: white;       background-color : gray;  **width**: 100%;  }   heading {  **color**: green;       font-size : 40px;       background-color : powderblue;  }   heading, title, author, publisher, edition, price {       display : block;  }   title {       font-size : 25px;       font-weight : bold;  } |

