UNIT 2

Web Page Designing

HTML

- HTML is a **markup** language for **describing** web documents (web pages).
- HTML stands for **H**yper **T**ext **M**arkup **L**anguage
- A markup language is a set of **markup tags**
- HTML documents are described by **HTML tags**
- Each HTML tag **describes** different document content

HTML tags

- Keywords surrounded by angle brackets eg. <html>
- Normally comes in pairs of starting and ending tags eg. <title> </title>
- Tags are interpreted by the browser but not displayed.

HTML document

- Describes a web page.
- Contains HTML tags and plaintext
- Example home.html
- Browser reads an html document and displays them as web pages.

A small HTML document:

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>

<h1>My First Heading</h1>
<p>My first paragraph.</p>

</body>
</html>
```

Formating

- BGCOLOR

Background color. Eg: BGCOLOR=“#C0C0C0”

Gives silver color to the background.

- LEFTMARGIN

States the size of the left margin. Specified number of spaces are left at the left side of the page and then the content starts. Eg: LEFTMARGIN=“50”

- TOPMARGIN

States the size of the top margin. Specified number of spaces are left at the top of the page and then the content starts. Eg: TOPMARGIN=“50”

- ALIGN

Left, right or centre alignment

Formatting Tags

- <b>

Defines a text bold.

- <big>

Defines a big text.

- <i>

Italics text

- <small>

Small text

- <sup>

Superscripted text

- <sub>

subscripted text

-

Line break

- <p>

paragraph

NESTED ELEMENTS

```
<!DOCTYPE html>
<html>
<body>

<h1>My First Heading</h1>
<p>My first paragraph.</p>

</body>
</html>
```

PREFORMATTING

```
<html>
<pre>
 This is preformatting.

 This is preformatting.

 This is preformatting.

 This is preformatting.
</pre>
</html>
```

List

Ordered List: numbered list

```
<OL TYPE=A>
<LI>Apples
<LI>Peaches
<LI>Oranges
</OL>
```

TYPE Numbering Style

- 1:1, 2,…
- A:a, b,…
- A: A, B,…
- I: i, ii,…
- I: I,II,..

Unordered list: Bulleted list

```
<UL TYPE=”square” >
<LI> COMPUTER CONCEPTS
<LI> MS-WINDOWS
<LI>MS-EXCEL
</UL>
```

Type may be square, disc, and circle.

Short Quotations

- The HTML **<q>** element defines a short quotation.
- Browsers usually insert quotation marks around the <q> element.
- **Example**

WWF's goal is to: <q>Build a future where people live in harmony with nature.</q>

HTML Comments

- Comment tags <!-- and --> are used to insert comments in HTML.

HTML Styles - CSS

CSS stands for **C**ascading **S**tyle **S**heets

Styling can be added to HTML elements in 3 ways:

- Inline - using a **style attribute** in HTML elements
- Internal - using a **<style> element** in the HTML <head> section
- External - using one or more **external CSS files**

Inline Styling (Inline CSS)

- **Inline styling** is useful for applying a unique style to a single HTML element:
- Inline styling uses the **style attribute**.
- This inline styling changes the text color of a single heading:
- **Example**

```
<h1 style="color:blue">This is a Blue Heading</h1>
```

Internal Styling (Internal CSS)

- An internal style sheet can be used to define a common style for all HTML elements on a page.
- **Internal styling** is defined in the **<head>** section of an HTML page, using a **<style>** element:

```
<style>
body {background-color:lightgrey}
h1   {color:blue}
p    {color:green}
</style>
</head>
<body>
```

External style sheet

- External style sheet are ideal when the style is applied to many pages.
- With external style sheets, you can change the look of an entire web site by changing one file.
- **External styles** are defined in an external CSS file, and then linked to in the **<head>** section of an HTML page:

```
<head>
  <link rel="stylesheet" href="styles.css">
</head>
```

CSS SYNTAX

- A CSS rule set consists of a selector and a declaration block:
- The selector points to the HTML element you want to style.
- The declaration block contains one or more declarations separated by semicolons.
- Each declaration includes a property name and a value, separated by a colon.

CSS Comments

- Comments are used to explain your code, and may help you when you edit the source code at a later date. Comments are ignored by browsers.
- A CSS comment starts with /* and ends with */. Comments can also span multiple lines:
- **Example**

```
p {
   color: red;
   /* This is a single-line comment */
   text-align: center;
}
```

CSS selectors

- CSS selectors allow you to select and manipulate HTML elements.
- CSS selectors are used to "find" (or select) HTML elements based on their id, class, type, attribute, and more.

Element Selector

- The element selector selects elements based on the element name.
- You can select all <p> elements on a page like this: (all <p> elements will be center-aligned, with a red text color)
- **Example**

```
p {
   text-align: center;
   color: red;
}
```

ID Selector

- The id selector uses the id attribute of an HTML element to select a specific element.
- An id should be unique within a page, so the id selector is used if you want to select a single, unique element.
- To select an element with a specific id, write a hash character, followed by the id of the element.
- **Example**

```
#para1 {
   text-align: center;
   color: red;
}
```

CSS Background

The background-color property specifies the background color of an element.
The background color of a page is set like this:

Example

```
body {
    background-color: #b0c4de;
}
```

Background Image

The background-image property specifies an image to use as the background of an element.

By default, the image is repeated so it covers the entire element.
The background image for a page can be set like this:

Example

```
body {
    background-image: url("paper.gif");
}
```

TEXT FORMATTING

Text Color

The color property is used to set the color of the text.

With CSS, a color is most often specified by:

- a HEX value - like "#ff0000"
- an RGB value - like "rgb(255,0,0)"
- a color name - like "red"

```
body {
    color: red;
}

h1 {
    color: #ff0000;
}

p.ex {
    color: rgb(255,0,0);
}
```

Text Alignment

The text-align property is used to set the horizontal alignment of a text.

Text can be centered, or aligned to the left or right, or justified.

When text-align is set to "justify", each line is stretched so that every line has equal width, and the left and right margins are straight (like in magazines and newspapers).

```
h1 {
    text-align: center;
}

p.date {
    text-align: right;
}

p.main {
    text-align: justify;
}
```

Text-decoration

The text-decoration property is used to set or remove decorations from text.

The text-decoration property is mostly used to remove underlines from links for design purposes.

```
a{
   text-decoration:none;
}
```

Text Transformation

The text-transform property is used to specify uppercase and lowercase letters in a text. It can be used to turn everything into uppercase or lowercase letters, or capitalize the first letter of each word.

```
p.uppercase {
   text-transform: uppercase;
}

p.lowercase {
   text-transform: lowercase;
}

p.capitalize {
   text-transform: capitalize;
}
```

Tables

Table is a collection of rows and columns.
A. Table is created by <table>…… </table> having following attributes:
<table ALIGN = “…….”
BORDER = “……..”
CELLPADDING = “……..”
CELLSPACING = “…….”
WIDTH = “…….” >

_____

_____
</table>

Align may be center, left, right.
Border may be 1,2,3, and so on.
Space between content and inside of a cell is called **cellpadding**. It may be 1,2,3 and so on.
Space between cells in a table is called **cellspacing.** It may be 1,2,3 and so on.

<tr>…..</tr> is used to add a row in a table.

<td>……</td> is used to add table data with following attributes

<td ALIGN = “…….”

COLSPAN = “……..”

ROWSPAN = “……..” >

_____

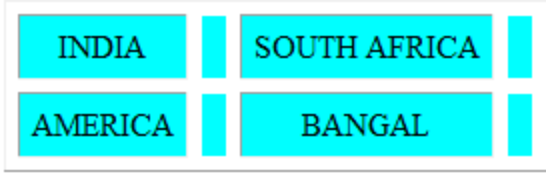
_____

</td>

- Align may be center, left, right.
- **Colspan:** Width of the cell in terms of numbers of columns when a cell occupies more than one column. It may 1,2,3 and so on.
- **Rowspan:** Height of the cell in terms of numbers of rows when a cell occupies more than one row. It may be 1,2,3 and so on.

Example: HTML code for creating two rows and two columns.

```
<html>
<body>
<table border = "1" cellpadding = "6" cellspacing ="6">
<tr bgcolor = "cyan">
<td align = "center">INDIA <td>
<td align = "center">SOUTH AFRICA <td>
</tr>
<tr bgcolor = "cyan">
<td align = "center">AMERICA <td>
<td align = "center">BANGAL <td>
</tr>
</table>
</body> </html>
```



Images

A. The <IMG> tag is used to insert a GIF or JPEG graphic into the document. The attributes are:

<IMG SRC = "MyImage.gif" BORDER = “3” WIDTH = “100%”>

B. Image links:

<A HREF="http://www.microsoft.com/">

< IMG SRC = "MyImage.gif" Microsoft Homepage </A>

C. ALT attributes: Alternative attribute is used when the required image is not supported by the browser.

<IMG SRC = “ABC.GIF” **ALT** = “A Simple Image”>

Frames

Frames are the division of window either horizontally or vertically. Frames enable the user to divide a page into number of rectangular regions/ window of various sizes.

Example:

```
<html>
<FRAMESET Rows = "30%,*" >
<Frame Src="Header.html" >
<FRAMESET Cols = "25%,*">
<Frame Src="Index.html" >
<Frame Src="Details.html" >
</FRAMESET>
</FRAMESET>
</html>
```

Forms

Form is a group of various elements just like examination form. A form film is a data-entry field on a page. A user supplies information into a field either by typing text or by selecting a value from a list of predicted values.

Application areas for forms:

Education sites

Online purchasing of orders

Collecting feedback about a website

Providing the interface for a chat session etc.

Creating a Form:

<form>……</form> tag is used to create a form. It has three parts.

A. Form header

B. Input fields/ elements

C. Action Buttons

Forms(cont..)

A. Form header: In header we have the following attributes.

I. Method: it specifies how the browser should communicate with the server.

It has two types:

GET: It is just getting or retrieving of data.

POST: It is used for insertion, updation and sending e-mail of data.

II. Action: It specifies what CGI script is used to process the data.

III. NAME: name of the form.

<form action = “ABC.html” method = “POST” name="Form1">

_____

_____

</form>

B. Action buttons: We design action buttons using input tags.

Forms(cont..)

C. Input fields/ elements:

I. Various Inputs :

<input TYPE= "….." NAME = "……." VALUE = "……." ALIGN= "……"
SIZE = "……." MAXLENGTH = "……">

Type may be text, button, checkbox, image, password, radio, reset, submit.

II. Multiple line text input:

<textarea rows = "…." Cols = "….." wrap = ".....">

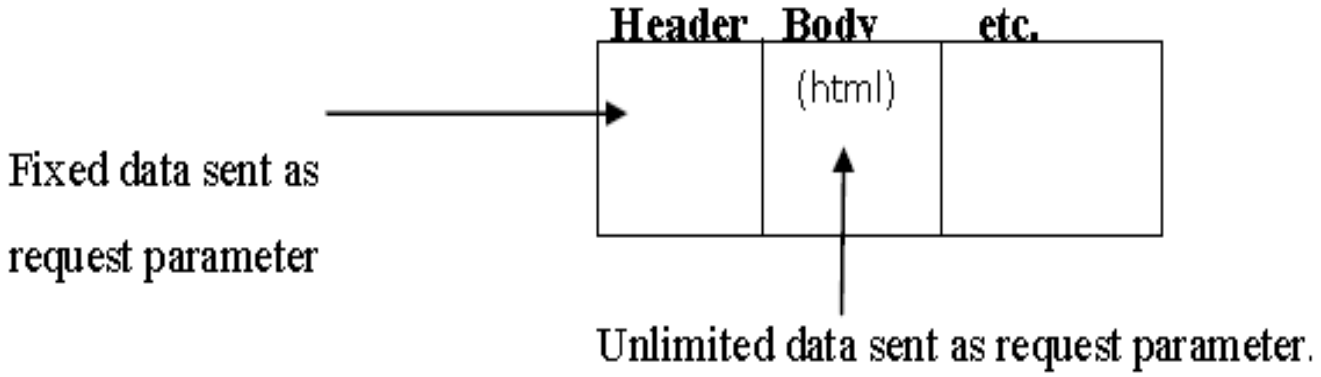
Rows identify height, cols identify width and wrap means word wrapping. The value of wrap may be OFF, VIRTUAL (means long lines) and PHYSICAL (means wrap points).

III. Pull down menus:

<select name = "list name" >
<Option> option1
<Option> option2
</select>

✓ **Difference between GET and POST method:**

GET	POST
1) It used for static resources.	1) It used for dynamic resources.
2) In this, request parameters are append to the URL and send as part of **request header.**	2) In this request parameters are sent as part of **request body.**
3) All parameters are shown in the address bar of browser.	3) Not shown.
4) Maximum data that can be sent as request parameter is determined by the size of header which has a fixed size.	4) Unlimited data can be sent as request parameter.



CSS

Cascading Style sheet: If you want to apply similar settings for all web pages in the website, This can be done by putting all the style rules in a style sheet file and then importing or linking it with your HTML document. This method of linking or importing is called **cascading style sheet (CSS).**
It must be saved with **.CSS** extension.

Style Sheet Properties:
I. Font Properties:
a. Font-family: denotes font
b. Font-size: denotes the size of the text.
c. Font-style: style of the text like normal, bold, italic, etc.
d. Font-weight: denotes the weight or darkness of the font. Ranges from 100 to 900 by increments of 100.

II. Text Properties:

a. Letter-spacing: space between letters.

b. Word-spacing: space between words.

c. Vertical-align: denotes the vertical positioning of the text and images with respect to the base line. The possible values include baseline, sub, super, top, text-top, middle percentage values etc.

d. Text-align: specifies the alignment of the text. The possible values are center, justify etc.

e. Text-indent: denotes margins.

f. Text-transform: denotes the transformation of text. The possible values are capitalize, uppercase, lowercase etc.

g. Text-decorate: denotes the text decoration. Values will be blink, linethrough,
over line, underline etc.

III. Color and Background Properties:

a. Color

b. Background-color

c. Background-image

d. Background-image.

Style Sheet Properties(cont..)

IV. Box Properties:

a. Margin properties: like 15 pt. etc.

b. Border properties:

Border-style: solid, double, groove, ridge etc.

Border-width: in terms of pt.

Border-color

V. Padding Properties: The space between an element's border and its content can

be specified four padding regions can be set using the padding-top, padding-right,

padding-bottom and padding-left properties.

CSS Example

```
<html>
<head>
<title>My document</title>
<link rel="stylesheet" type="text/css" href="style.css" />
</head>
<body>
<h1>My first stylesheet</h1>
</body>
</html>
```

style.css:

```
body {
background-color: #FF0000;
}
```

XML

- XML is the case sensitive markup Language.
- With XML you can store information about your document and pieces of your document.
- You can use that information as criteria for displaying page but also for validating digital signatures, sharing data across systems, processing data for other applications and much more.
- **HTML is limited in:**

  Intelligence: How will data knows itself?

  Adaption: How will data changes in response to changing time?

  Maintenance: How easily data is cared for?

XML (cont..)

- **Reasons for picking up XML:**
  Easy and simple.
  Easy to maintain.
  Contains only data and markup.

- **Features of XML:**
  XML can be used with existing protocol.
  Support a wide variety of applications.
  Compatible with SGML.
  It is easy to write programs that process XML document.
  XML documents are reasonable and clear to person.

- **Extensible means:**
  Reusable
  Create your own data type derived from standard data types.
  Reference of your code in same document.

Document type definition

- A Document Type Definition (DTD) defines the structure and the legal elements and attributes of an XML document.
- A DTD can be declared inside an XML document or in an external file.
- With a DTD, independent groups of people can agree to use a standard DTD for interchanging data.
- Your application can use a standard DTD to verify that the data you receive from the outside world is valid.
- You can also use a DTD to verify your own data.
- Once a DTD is created and a document written based on that DTD, the document is then compared to the DTD.
- If your XML document follows the rules listed in the DTD, then the document is said to be *valid*.

XML document with an internal DTD

```
<?xml version="1.0"?>
<!DOCTYPE note [
<!ELEMENT note (to,from,heading,body)>
<!ELEMENT to (#PCDATA)>
<!ELEMENT from (#PCDATA)>
<!ELEMENT heading (#PCDATA)>
<!ELEMENT body (#PCDATA)>
]>
<note>
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend</body>
</note>
```

XML document with a reference to an external DTD

```
<?xml version="1.0"?>
<!DOCTYPE note SYSTEM "note.dtd">
<note>
  <to>Tove</to>
  <from>Jani</from>
  <heading>Reminder</heading>
  <body>Don't forget me this weekend!</body>
</note>
```

And here is the file "note.dtd", which contains the DTD:

```
<!ELEMENT note (to,from,heading,body)>
<!ELEMENT to (#PCDATA)>
<!ELEMENT from (#PCDATA)>
<!ELEMENT heading (#PCDATA)>
<!ELEMENT body (#PCDATA)>
```

XML schema

a. XML Schema is an XML-based alternative to DTD.

b. It describes the structure of an XML document.

c. It is also referred to as XML Schema definition (XSD).

d. It contains:

- Defines elements, attributes that can appear in a document.
- Defines child elements, order and number of child elements.
- Defines weather an element is empty or can include text.
- Defines data types for elements and attributes.
- Defines default and fixed values for elements and attributes.

e. XML Schemas are successors of DTD:

- XML schemas are extensible for future purpose.
- Are richer and more powerful than DTD.
- Are written in XML.
- It supports data types, namespaces.

Example:

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="URL/XMLSchema"
targetNamespace="URL"
xmlns="URL"
elementFormDefault="qualified">
<xs:element name="note">
<xs:complexType>
<xs:sequence>
<xs:element name="to" type="xs:string"/>
<xs:element name="from" type="xs:string"/>
<xs:element name="heading" type="xs:string"/>
<xs:element name="body" type="xs:string"/>
</xs:sequence>
</xs:complexType>
</xs:element> </xs:schema>
```

.

xmlns:xs="URL/XMLSchema" the elements and data types used in the schema come from the "URL,XMLSchema".

targetNamespace="URL" indicates that the elements defined by this schema

xmlns="URL" indicates that the default namespace is URL.

elementFormDefault="qualified" > indicates that any elements used by the
XML instance document which were declared in the schema must be namespace qualified

Object Models

- Viewing XML using the **“XML data source object”.**
- Data source objects are used for what Microsoft calls data binding. Data
- **Binding** is microsoft’s way of bringing data manipulation to the browser (Client) away from the server.
- Normally, if you want a new view on the data, you resubmit a query to the server.
- The server performs the necessary calculations and sends a new HTML page together with the data to the client.
- There are two types of object models:

  A. DOM (Dynamic Object Model)

  B. SAX (Simple API for XML)

Using XML Processors: DOM and SAX

DOM (Dynamic Object Model):

I. DOM is a platform and language independent object model for representing HTML or XML and related formats.

II. DOM supports navigation in any direction (e.g. parent and previous sibling), hence the DOM is likely to be best suited for applications where the document must be accessed repeatedly or out of sequence order.

III. Levels of DOM:

a) Core DOM: standard model for any structured document.
b) XML DOM: standard model for XML document.
c) HTML DOM: standard model for HTML documents.

IV. DOM Nodes: According to the DOM, everything in an XML document is a node.

a) The entire document is a document node.
b) Every XML element is an element node.
c) The text in the XML elements are text node.
d) Every attribute is an attribute node.
e) Comments are comment nodes.

Example:

<? Xml version = “1.0” ?>

<bookstore>

<book category = “cooking”>

<title lang = “en”> Sweet Disses </title>

<author> kshitij </author>

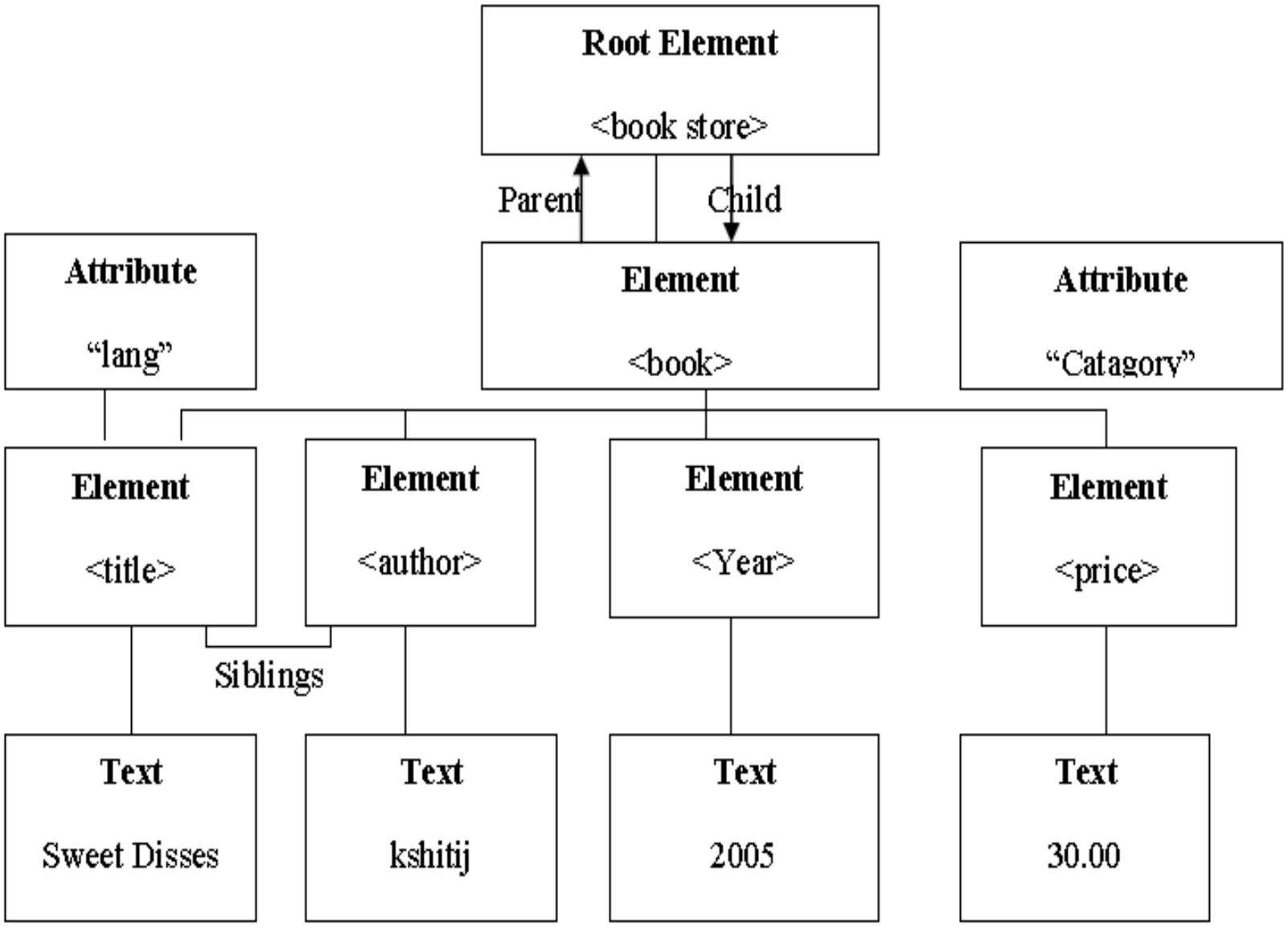
<year> 2005 </year>

<price> 30.00 <price>

</book>

</bookstore>

View DOM Node Tree:



SAX (Simple API for XML):

SAX is a serial access parser API for XML. SAX provides a mechanism for reading data from an XML document. It is a popular alternative to the DOM.

SAX parser for XML processing: A parser which implements SAX (SAX parser) functions as a stream parser, with an event-driven API. The user defines a number of call back methods that will be called when event occur during parsing. The SAX events include:

a) XML text nodes.

b) XML element nodes.

c) XML processing instructions.

d) XML comments.

SAX parsing is unidirectional. Previously data cannot be re-read without starting the parsing operation again.

Example:

```
<?xml version="1.0"?>
<RootElement param="value">
<FirstElement> Some Text </FirstElement>
<SecondElement param2="something">
Pre-Text <Inline>Inlined text</Inline> Post-text.
</SecondElement>
</RootElement>
```

THE XML document, when passed through a SAX parser, will generate a sequence of events like the following:

i. XML processing instruction, named XML, with attribute version = “1.0”.
ii. XML Element start, named *RootElement*, with an attribute *param* equal to "value"
iii. XML Element start, named *FirstElement*
iv. XML Text node, with data equal to "Some Text" (note: text processing, with regard to spaces, can be changed)
v. XML Element end, named *FirstElement*
vi. XML Element start, named *SecondElement*, with an attribute *param2* equal to "something"
vii. XML Text node, with data equal to "Pre-Text"
viii. XML Element start, named *Inline*
ix. XML Text node, with data equal to "Inlined text"
x. XML Element end, named *Inline*
xi. XML Text node, with data equal to "Post-text."
xii. XML Element end, named *SecondElement*
xiii. XML Element end, named *RootElement*

Introduction to Java Script

- JavaScript is the programming language of HTML and the Web.

- JavaScript is one of the **3 languages** all web developers **must** learn:

  1. **HTML** to define the content of web pages
  2. **CSS** to specify the layout of web pages
  3. **JavaScript** to program the behavior of web pages

Dynamic HTML

- Dynamic HTML is a collective term for a combination of Hypertext Markup Language (HTML) tags and options that can make Web pages more animated and interactive than previous versions of HTML. Much of dynamic HTML is specified in HTML 4.0. Simple examples of dynamic HTML capabilities include having the color of a text heading change when a user passes a mouse over it and allowing a user to "drag and drop" an image to another place on a Web page. Dynamic HTML can allow Web documents to look and act like desktop applications or multimedia productions.

- **The Concepts and Features in Dynamic HTML**

An object-oriented view of a Web page and its elements

Cascading style sheets and the layering of content

Programming that can address all or most page elements

Dynamic fonts