Unit Content

- HTML, DOM, Basic Structure of HTML Document,
- Markup Tags, Working with Lists and Tables,
- Working with Hyperlink, Image Handling and Frames,
- HTML form for use Inputs, Various new form elements,
- XML Syntax, Attributes and Namespace, HTTP Request ,
- Parser DOM, Xpath, XSLT, Xquerry, Xlink,
- Validator, DTD and XML Schema.

Unit Objective

HTML Objectives:

- Understand the basic structure of an HTML document.
- Learn and apply HTML tags to create well-formed web pages.
- Format text and create headings, paragraphs, lists, and links.
- Incorporate images, videos, and other media into web pages.
- Construct tables to organize data.
- Implement forms to collect user input.
- Apply CSS styles to enhance the visual appearance of web pages.
- Understand HTML semantics and the proper use of semantic elements.

Unit Objective

XML Objectives:

•Understand the basic syntax and structure of XML documents.
•Create well-formed XML documents using appropriate tags and nesting.
•Define and use XML namespaces for proper document organization.
•Validate XML documents against Document Type Definitions (DTD) or XML Schema.
•Transform XML data using XSLT (eXtensible Stylesheet Language Transformations).
•Query and extract data from XML documents using XPath.
•Understand the purpose and usage of XML parsers and processors.
•Interact with XML data through programming languages and APIs.

Topic Objective

Introduction to HTML: This objective covers the fundamental aspects of HTML, including document structure, text formatting, linking to other web pages, adding images, and establishing a solid understanding of HTML syntax and elements. By achieving this objective, you'll have a strong foundation in HTML that will enable you to create static web pages effectively.

Introduction to XML: This objective encompasses the essential aspects of XML, including document structure, syntax rules, the use of namespaces for organizing data, transforming XML with XSLT, extracting information with XPath, and integrating XML within various programming environments. By achieving this objective, you'll have a solid grasp of XML and its applications for data storage, interchange, and manipulation.

HTML

HTML5 is the latest version of the Hypertext Markup Language (HTML), which is the standard language used for creating and structuring web pages.

It introduced several new features and improvements over its predecessor, HTML4, to enhance the functionality and interactivity of web content.

Here are some of the basics of HTML5.

HTML

HTML, or HyperText Markup Language, is the standard markup language used to create web pages and structure the content within them.

It consists of a set of elements or tags that define the structure and presentation of web documents.

HTML documents are the building blocks of the World Wide Web, and they can be rendered by web browsers to display text, images, links, forms, multimedia, and interactive elements.

Key Aspects of HTML

Structure and Semantics: HTML provides a structured way to organize content on a web page. It uses a system of tags, each with a specific purpose, to define headings, paragraphs, lists, images, and more.

This structural clarity is essential for both human readers and search engines to understand the content and context of a webpage.

Cross-Browser Compatibility: HTML is a universal standard supported by all modern web browsers.
When you create a webpage using HTML, you can be reasonably confident that it will display consistently across different browsers and devices, ensuring a consistent user experience.

Key Aspects of HTML

Platform Independence: HTML is platform-independent, which means it can be used on various operating systems and devices, including computers, smartphones, tablets, and more.
This makes it a versatile choice for creating content accessible to a broad audience.

Accessibility: HTML supports accessibility features, such as semantic elements and attributes, that help make web content more accessible to people with disabilities.
These features improve screen reader compatibility and ensure that content is navigable and understandable by all users.

Key Aspects of HTML

Integration with Other Technologies: HTML can be seamlessly integrated with other web technologies like CSS (Cascading Style Sheets) for styling and JavaScript for adding interactivity and dynamic behavior to web pages.

This trio of technologies (HTML, CSS, and JavaScript) forms the foundation of modern web development.

Rich Media Support: HTML5, the latest version of HTML, introduced native support for multimedia elements like audio and video, reducing the need for third-party plugins like Flash.

This makes it easier to include multimedia content directly in web pages.

HTML- DOM

The DOM (Document Object Model) in HTML5 is a programming interface that represents and manipulates the structure and content of web documents, such as HTML and XML documents.
It provides a way for programs and scripts to interact with the content of a web page, allowing developers to dynamically change, add, or delete elements and content on a webpage using JavaScript or other scripting languages.

Here are some key aspects of the DOM in HTML5:

Key Aspects of the DOM

Tree Structure: The DOM represents an HTML document as a tree structure, with each HTML element as a node in the tree.
The top node is the document itself, and it branches out to include all the elements on the page, including elements like headings, paragraphs, images, and more.

Accessibility: The DOM provides a way to access and manipulate the content and structure of an HTML document.
You can access elements by their tag name, class, ID, or other attributes, making it easy to modify the page dynamically based on user interactions or other events.

Key Aspects of the DOM

Methods and Properties: Each node in the DOM tree has associated methods and properties that can be used to manipulate it.
For example, you can change the content of an element using the innerHTML property, modify attributes like src or href, and create, delete, or move elements using various methods.

Event Handling: HTML5 DOM allows you to attach event listeners to elements. Event listeners enable you to respond to user interactions, such as clicks, keypresses, and mouse movements, and trigger specific actions or functions in response.

Key Aspects of the DOM

Traversal: You can traverse the DOM tree using methods like **querySelector**, **querySelectorAll**, **getElementsByTagName**, and more. These methods allow you to find specific elements within the DOM based on their attributes or structure.

Dynamic Updates: HTML5 and the DOM make it easy to update a web page dynamically without requiring a full page reload. You can add or remove elements, change their content, or modify their appearance in response to user actions or data from external sources.

Cross-Browser Compatibility: HTML5 and the DOM are designed to work consistently across different web browsers, ensuring that your web applications function properly for users on various platforms.

DOM

Here's a simple example of how you might use the DOM to change the text of a paragraph element with JavaScript:

```
<!DOCTYPE html>
<html>
<head>
   <title>DOM Example</title>
</head>
<body>
   <p id="myParagraph">This is a paragraph.</p>

 <script>
      // Get the paragraph element by its ID
      var paragraph = document.getElementById("myParagraph");

      // Change the text content of the paragraph
      paragraph.textContent = "This is a modified paragraph.";
   </script>
</body>
</html>
```

Understand the structure of an HTML page

1. **Document Type Declaration**: At the very beginning of an HTML document, you need to include the document type declaration, which tells the browser that the document is written in HTML. In HTML5, the doctype declaration is simply **<!DOCTYPE html>.**

2. HTML Tag: The HTML tag **(<html>)** serves as the root element of the HTML document. It encapsulates the entire content of the page.

3. Head Section: The **<head>** element contains metadata and other non-visible information about the document. It does not represent the visible content of the page. Some common elements within the <head> section include:

<meta>: Defines metadata such as character encoding, viewport settings.
<title>: Specifies the title of the page, which appears in the browser's title bar or tab.
<link>: Used to link external CSS stylesheets or other resources.
<script>: Used to include JavaScript code or link to external JavaScript files.

Cont…

4. **Body Section:** The **<body>** element represents the main content of the web page. It contains all the visible elements that are displayed in the browser window. Common elements within the <body> section include:

<header>: Represents the introductory content at the top of the page, often containing logos, navigation menus, or headings.
<nav>: Represents a section containing navigation links.
<main>: Represents the main content area of the page.
<article>: Represents a self-contained composition within a document, such as a blog post or a news article.
<section>: Represents a standalone section within a document, typically grouped by a common theme.
<aside>: Represents content that is tangentially related to the main content, such as sidebars or pull quotes.
<footer>: Represents the footer section at the bottom of the page, often containing copyright information, links, etc.

HTML Tags

Tags can be defined as the instructions which are being directly embedded in the text of an HTML document.
The types of tags used in the HTML document are responsible to tell a web browser to do something (follow the instruction) instead of just displaying text.

In an HTML document, all tag names are differentiated from other simple text. The tag names are enclosed in between angle brackets or a 'less than' and a 'greater than' symbol, **(<) and (>).**

Three Types of Tags in HTML

- Paired Tags and Unpaired Tags
- Self- Closing Tags
- Utility Based
    1. Formatting Tags
    2. Structured Tags
    3. Control Tags

HTML Tags

1. Paired Tags

An HTML tag is known as a paired tag when the tag consists of an opening tag and a closing tag as its companion tag.
An HTML Paired tag starts with an opening tag: the tag name enclosed inside the angle brackets; for example, a paragraph opening tag is written as ' '.
The content follows the opening tag, which ends with an ending tag: the tag name starting with a forward slash; for example, an ending paragraph tag is written as ' '.
The first tag can be referred to as the 'Opening Tag', and the second tag can be called Closing Tag.

Example :

 This text is a paragraph .

HTML Tags

Unpaired Tags

An HTML tag is called an unpaired tag when the tag only has an opening tag and does not have a closing tag or a companion tag. The Unpaired HTML tag does not require a closing tag; an opening tag is sufficient in this type. Unpaired tags are sometimes also named as Standalone Tags or Singular Tags since they do not require a companion tag.

Example:

```
<p> This is a paragraph </p>
<hr>
<i>
 <b> This is a bold and italicized text </b>
</i>
```

HTML Tags

2. Self-Closing Tags

Self-Closing Tags are those HTML tags that do not have a partner tag, where the first tag is the only necessary tag that is valid for the formatting.

The main and important information is contained WITHIN the element as its attribute. An image tag is a classic example of a self-closing tag. Let's see it in action below:

Example:

```
<img src="a.jpg" alt="This is an alternate text">
```

Note: In the older versions, the self-closing tags use a 'forward slash' before the ending or closing 'greater than' sign/symbol, as written below:

```
<img src=”a.jpg” alt=”This is an alternate text” />
```

HTML Tags

3. Utility-Based Tags

The HTML tags can be widely differentiated on the basis of their utility, that is, on the basis of the purpose they serve.

We can divide them basically into three categories as discussed below:

Formatting Tags

The HTML tags that help us in the formatting of the texts like the size of the text, font styles, making a text bold, etc.

This is done using tags like <font>, <b>, <u>, etc. Tables, divisions, and span tags are also those tags that help format a web page or document and set the layout of the page.

Below is a small program using divisions for formatting the page along with some other formatting tags.

HTML Tags

```
<body>
<div class="container">
<div class="row">
<div class="col-25">
<label for="email"><b>Name</b></label>
</div>
<div class="col-35">
<input type="text" placeholder="First" name="fname" required>
</div>
<div class="col-35">
<input type="text" placeholder="Last" name="lname" required>
</div>
</div>
</div>
</body>
```

HTML Tags

Structure Tags

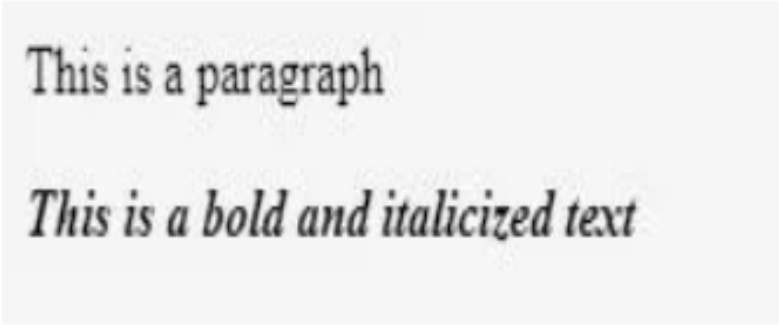
The HTML tags that help in structuring the HTML document are called Structure Tags. Description, head, html, title, body, etc., form the group of the page structure tags.

The structure tags only assist in creating or forming the basic html page from the root; that is, they do not affect or has any hand in the formatting of texts.
So a basic HTML program is the basic group of structural tags:

HTML Tags

Example:

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>Types of Tags Demo</title>
</head>
<body>
<p> This is a paragraph </p>
<i><b> This is a bold and italicized text
</b></i>
</body>
</html>
```



HTML Tags

Control Tags

Another category of tags that can be created is 'Control Tags'. The Script tags, radio buttons or checkboxes, the Form tags, etc., forms the control tags.

These are the tags that are used in managing content or managing scripts or libraries that are external. All the form tags, drop-down lists, input text boxes, etc., are used in interacting with the visitor or the user.

The above distinction of the HTML tags is based on the type of tags and their utility.

The HTML tags can also be simply divided based on basic categories like Basic HTML Root Tags, Formatting tags, Audio and Video Tags, Form and Input Tags, Frame Tags, Link Tags, List Tags, Table Tags, Style Tags, Meta Tags, etc.

How Many HTML Tags Are There?

According to Mozilla Developer Network(MDN), there are 142 HTML tags. However, only 115 are functional with the other 27 being deprecated or obsolete.
Each of the HTML tags has different functionalities across semantic purposes that help define the structure of HTML documents designed to be displayed in a web browser.

There are 142 HTML tags within the latest HTML version, HTML 5.2. Only 115 tags are deemed functional and usable within any HTML code.

These tags indicate the beginning and end of an HTML element that serves as a node within the DOM to define the structure of a webpage.

Cont…

5. HTML Elements: Within the body section, you can use various HTML elements to structure and format your content.

Some commonly used elements include headings (<h1>, <h2>, etc.),
paragraphs (<p>),
lists (<ul>, <ol>, <li>),
images (<img>),
links (<a>),
and more.

New Semantic elements in HTML 5

HTML5 introduced several new semantic elements that provide a more structured and meaningful way to represent content on web pages.

<header>: Represents the introductory content or a group of navigational links for a section or the whole document.

<nav>: Defines a section containing navigation links.

<main>: Represents the main content of a document. It should be unique and not nested within other structural elements like <article>, <section>, or <aside>.

<article>: Represents a self-contained composition that can be independently distributed or reused, such as a blog post, a news story, or a forum post.

<section>: Defines a thematic grouping of content within a document, such as chapters, tabbed content, or a group of related items.

Cont…

<aside>: Represents a section of content that is tangentially related to the main content, such as sidebars, pull quotes, or advertising.

<figure> and **<figcaption>:** <figure> represents self-contained content like images, videos, or illustrations, while <figcaption> provides a caption or description for the content within the <figure> element.

<footer>: Defines the footer section of a document or a section. It typically contains information about the author, copyright data, links to related documents, or contact information.

<time>: Represents a specific date, time, or duration. It can be used to mark up events, publication dates, or timestamps.

<mark>: Highlights a portion of text to indicate its relevance or importance within the context.

Tables

Tables in HTML5 are used to present tabular data, where information is organized in rows and columns. HTML5 introduced new attributes and elements to enhance the accessibility and semantic structure of tables.

Table Structure:

<table>: Defines the container for the entire table.
<thead>, <tbody>, and **<tfoot>:** These elements group the table's header, body, and footer sections, respectively. They provide structural organization to the table, but their usage is optional.

Table Headers:

<th>: Represents a table header cell. It is used to define the headers of columns or rows.
<tr>: Defines a table row.
<th scope="">: This attribute specifies the scope of a header cell. It can take values like "row", "col", "rowgroup", or "colgroup" to indicate the scope of the header.

Cont…

Table Data Cells:

<td>: Represents a table data cell. It is used to define the content within a table row.
Table Captions:

<caption>: Defines a caption for the table. It should be placed immediately after the opening <table> tag.

Table Spanning:

colspan attribute: Specifies the number of columns a table cell should span.

rowspan attribute: Specifies the number of rows a table cell should span.

Cont…

Table Headers and ID References:

<th id="">: Assigns a unique ID to a table header cell.

<td headers="">: Specifies the ID(s) of the header cell(s) that apply to a data cell. This association helps assistive technologies understand the relationships between headers and data cells.

HTML Lists

HTML lists allow web developers to group a set of related items in lists.

An unordered HTML list:

- Item
- Item
- Item
- Item

An ordered HTML list:

1. First item
2. Second item
3. Third item
4. Fourth item

Cont…

Unordered HTML List

```
<ul>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ul>
```

Ordered HTML List

```
<ol>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>
```

Cont…

HTML Description Lists

```
<dl>
  <dt>Coffee</dt>
  <dd>- black hot drink</dd>
  <dt>Milk</dt>
  <dd>- white cold drink</dd>
</dl>
```

Daily Quiz

1: Which tag is used to define the main heading of a webpage?

a) <header>
b) <h1>
c) <title>
d) <main>
Answer: b) <h1>

2: Which tag is used to create a paragraph in HTML?

a) <p>
b) <h2>
c)
d)
Answer: a) <p>

3: Which attribute is used to specify the URL of an image in the <img> tag?

a) src
b) alt
c) href
d) link
Answer: a) src

Daily Quiz

4: Which tag is used to create a numbered list in HTML?
a) <ol>
b) <ul>
c) <li>
d) <dl>
Answer: a) <ol>

5: Which attribute is used to provide alternative text for an image in the <img> tag?
a) src
b) alt
c) title
d) link
Answer: b) alt

6: Which tag is used to create a horizontal rule (a horizontal line) in HTML?
a) <hr>
b)
c) <line>
d) <rule>
Answer: a) <hr>

HTML Links

<a href="*url*">*link text*</a>

By default, links will appear as follows in all browsers:

- An unvisited link is underlined and blue
- A visited link is underlined and purple
- An active link is underlined and red

Cont…

HTML Links - The target Attribute

The target attribute specifies where to open the linked document.

The target attribute can have one of the following values:

_self - Default. Opens the document in the same window/tab as it was clicked
_blank - Opens the document in a new window or tab
_parent - Opens the document in the parent frame
_top - Opens the document in the full body of the window

```
<a href="https://www.niet.co.in//" target="_blank">Visit NIET!</a>
```

Cont…

Absolute URLs vs. Relative URLs

Both examples above are using an absolute URL (a full web address) in the href attribute.

A local link (a link to a page within the same website) is specified with a relative URL (without the "https://www" part):

```
<h2>Absolute URLs</h2>
<p><a href="https://www.niet.co.in/">NIET</a></p>

<h2>Relative URLs</h2>
<p><a href="html_images.asp">HTML Images</a></p>
```

HTML Images

HTML images are defined with the <img> tag.
The source file (src), alternative text (alt), width, and height are provided as attributes:

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

<h2>HTML Images</h2>
<p>HTML images are defined with the img tag:</p>

<img
src="https://i.pinimg.com/236x/da/e3/46/dae346dee0d6d9716583d3d475561bb7.jpg"
alt="reddit.com" width="104" height="142">

</body>
</html>
```

Topic Objective

Students were able to learn:

- Basics of HTML
- Concepts of HTML Tags
- Concepts of Tables
- Concepts of HTML Lists
- Concepts of HTML Links
- Concepts of HTML Images

Working with HTML Forms

An HTML form is used to collect user input. The user input is most often sent to a server for processing.

The HTML **<form>** element is used to create an HTML form for user input:

```
<form>
.
form elements
.
</form>
```

The **<form>** element is a container for different types of input elements, such as: **text fields, checkboxes, radio buttons, submit buttons, etc.**

The <input> Element

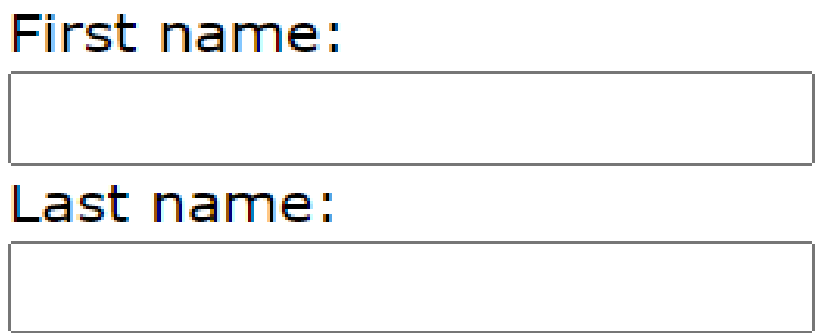
The HTML <input> element is the most used form element.
An <input> element can be displayed in many ways, depending on the type attribute.
Here are some examples:

Type	Description
<input type="text">	Displays a single-line text input field
<input type="radio">	Displays a radio button (for selecting one of many choices)
<input type="checkbox">	Displays a checkbox (for selecting zero or more of many choices)
<input type="submit">	Displays a submit button (for submitting the form)
<input type="button">	Displays a clickable button

Text Fields

The **<input type="text">** defines a single-line input field for text input.

```
<form>
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname"><br>
  <label for="lname">Last name:</label><br>
  <input type="text" id="lname" name="lname">
</form>
```



The <label> Element

- Notice the use of the **<label>** element in the example above.
- The **<label> tag** defines a label for many form elements.
- The <label> element is useful for screen-reader users, because the screen-reader will read out loud the label when the user focuses on the input element.
- The <label> element also helps users who have difficulty clicking on very small regions (such as radio buttons or checkboxes) - because when the user clicks the text within the <label> element, it toggles the radio button/checkbox.
- The for attribute of the <label> tag should be equal to the id attribute of the <input> element to bind them together.

Radio Buttons

The <input type="radio"> defines a radio button.
Radio buttons let a user select ONE of a limited number of choices.

```
<p>Choose your favorite Web language:</p>
<form>
  <input type="radio" id="html" name="fav_language" value="HTML">
  <label for="html">HTML</label><br>
  <input type="radio" id="css" name="fav_language" value="CSS">
  <label for="css">CSS</label><br>
  <input type="radio" id="javascript" name="fav_language" value="JS">
  <label for="javascript">JavaScript</label>
</form>
```

Choose your favorite Web language:

- ○ HTML
- ○ CSS
- ○ JavaScript

Checkboxes

The **<input type="checkbox">** defines a checkbox.
Checkboxes let a user select ZERO or MORE options of a limited number of choices.

```
<form>
  <input type="checkbox" id="vehicle1" name="vehicle1" value="Bike">
  <label for="vehicle1"> I have a bike</label><br>
  <input type="checkbox" id="vehicle2" name="vehicle2" value="Car">
  <label for="vehicle2"> I have a car</label><br>
  <input type="checkbox" id="vehicle3" name="vehicle3" value="Boat">
  <label for="vehicle3"> I have a boat</label>
</form>
```

This is how the HTML code above will be displayed in a browser:

☐ I have a bike
☐ I have a car
☐ I have a boat

The Submit Button

- **The <input type="submit">** defines a button for submitting the form data to a form-handler.
- The form-handler is typically a file on the server with a script for processing input data.
- The form-handler is specified in the form's action attribute.

```
<form action="/action_page.php">
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname" value="John"><br>
  <label for="lname">Last name:</label><br>
  <input type="text" id="lname" name="lname" value="Doe"><br><br>
  <input type="submit" value="Submit">
</form>
```

This is how the HTML code above will be displayed in a browser:

First name:

John

Last name:

Doe

Submit

The Submit Button

- **The <input type="submit">** defines a button for submitting the form data to a form-handler.
- The form-handler is typically a file on the server with a script for processing input data.
- The form-handler is specified in the form's action attribute.

```
<form action="/action_page.php">
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname" value="John"><br>
  <label for="lname">Last name:</label><br>
  <input type="text" id="lname" name="lname" value="Doe"><br><br>
  <input type="submit" value="Submit">
</form>
```

This is how the HTML code above will be displayed in a browser:

First name:

John

Last name:

Doe

Submit

The Name Attribute for <input>

Notice that each input field must have a name attribute to be submitted.

If the name attribute is omitted, the value of the input field will not be sent at all.

```
<form action="/action_page.php">
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" value="John"><br><br>
  <input type="submit" value="Submit">
</form>
```

HTML Input Types

Describing the different types for the HTML **<input>** element.

```
<input type="button">
<input type="checkbox">
<input type="color">
<input type="date">
<input type="datetime-local">
<input type="email">
<input type="file">
<input type="hidden">
<input type="image">
<input type="month">
<input type="number">
<input type="password">
<input type="radio">
<input type="range">
<input type="reset">
<input type="search">
<input type="submit">
<input type="tel">
<input type="text">
<input type="time">
<input type="url">
<input type="week">
```

HTML Input Attributes

Describing the different attributes for the HTML <input> element.

The value Attribute

The input **value** attribute specifies an initial value for an input field:

```
<form>
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname" value="John"><br>
  <label for="lname">Last name:</label><br>
  <input type="text" id="lname" name="lname" value="Doe">
</form>
```

Topic Objective

Students were able to learn:

- Working with HTML Forms
- Usage of various form tags
- Usage of various input elements
- Working with various semantic elements like Radio Buttons, CheckBoxes, Dropdown menus etc.

Daily Quiz

1: Which attribute is used to specify the URL that a link should navigate to?
a) src
b) link
c) href
d) url

Answer: c) href

2: How do you open a link in a new browser tab/window?
a) Using the new attribute
b) Using the target="_blank" attribute
c) Using the open attribute
d) Using the popup attribute

Answer: b) Using the target="_blank" attribute

3: How do you create an anchor (bookmark) within the same webpage?
a) Using the <a href="#bookmark"> tag
b) Using the <bookmark> tag
c) Using the <a href="anchor"> tag
d) Using the <a name="bookmark"> tag

Answer: d) Using the <a name="bookmark"> tag

Daily Quiz

4: Which attribute is used to specify the HTTP method for submitting a form?
a) action
b) method
c) submit
d) type
Answer: b) method

5: How do you create a text input field in a form?
a) <input type="text">
b) <textinput>
c) <input field="text">
d) <textfield>
Answer: a) <input type="text">

6: Which attribute is used to define a placeholder text in an input field?
a) value
b) text
c) placeholder
d) input
Answer: c) placeholder

Introduction to XML

XML, which stands for Extensible Markup Language, is a versatile and widely used markup language designed for storing and exchanging structured data. It provides a flexible way to describe and organize information in a hierarchical format.

XML uses tags to define elements within a document, similar to HTML. However, unlike HTML, which is primarily used for structuring web content, XML is not limited to any specific domain. It is a generic markup language that can be used for a variety of purposes, such as storing data, configuring settings, representing document structures, and more.

One of the key strengths of XML is its ability to define custom tags and document structures. This flexibility allows XML to adapt to different data formats and industry-specific needs. XML documents are typically human-readable and self-descriptive, making them easy to understand and interpret.

Introduction XML

XML Definition

- Xml (eXtensible Markup Language) is a mark up language.
- XML is designed to store and transport data.
- XML became a W3C Recommendation on February 10, 1998.
- XML is designed to be self-descriptive.
- XML is designed to carry data, not to display data.
- XML is platform independent and language independent.

Introduction XML(cont..)

Benefit and Feature Of XML

- XML separates data from HTML
- XML simplifies data sharing
- XML simplifies data transport
- XML simplifies Platform change

XML Syntax

- **Tags:** XML documents are made up of elements enclosed within start and end tags. Tags are used to define the structure and hierarchy of the document's content.
- Start tags begin with a less-than symbol (<), followed by the element name, and end with a greater-than symbol (>).
- End tags have a similar syntax, but also include a forward slash (/) before the element name.

```
<book>
  <title>XML Basics</title>
  <author>John Smith</author>
</book>
```

Cont…

Elements: Elements are the building blocks of an XML document. They represent individual pieces of information within the document. Each element has a start tag, an end tag, and may contain nested elements and/or text content.

Attributes: Attributes provide additional information about an element. They are placed within the start tag of an element and consist of a name-value pair. Attribute values are enclosed in quotation marks.

xml

```xml
<?xml version="1.0" encoding="UTF-8"?>
<bookstore>
  <book category="fiction">
    <title>Harry Potter and the Philosopher's Stone</title>
    <author>J.K. Rowling</author>
    <year>1997</year>
  </book>
  <book category="non-fiction">
    <title>The Power of Now</title>
    <author>Eckhart Tolle</author>
    <year>1997</year>
  </book>
</bookstore>
```

XML Namespaces

XML namespaces are used to avoid naming conflicts when different XML vocabularies or elements with the same name are combined. They provide a way to uniquely identify and differentiate elements and attributes in an XML document.

The concept of namespaces allows elements and attributes to be distinguished by associating them with a namespace prefix. A namespace is defined by a Uniform Resource Identifier (URI), which serves as its unique identifier. The URI can point to a web page, a schema definition, or any other resource that can be used to identify the namespace.

Namespace declarations are typically added to the root element of an XML document or to individual elements. They are declared using the xmlns attribute, where the namespace prefix is associated with its corresponding URI. The xmlns attribute without a prefix specifies the default namespace for elements that don't have a namespace prefix explicitly defined.

Cont…

Here's an example that demonstrates the use of namespaces in an XML document:

```
<?xml version="1.0" encoding="UTF-8"?>
<root xmlns:ns1="http://www.example.com/ns1"
xmlns:ns2="http://www.example.com/ns2">
<ns1:element1>This is element 1 from namespace ns1</ns1:element1>
<ns2:element2>This is element 2 from namespace ns2</ns2:element2>
</root>
```

In this example, two namespaces are declared: ns1 and ns2. The prefixes ns1 and ns2 are used to qualify the elements element1 and element2, respectively. By using different prefixes, the elements can be distinguished even if they have the same local name.

Cont…

Namespace prefixes are not mandatory and can be omitted, as shown in the following example:

```
<?xml version="1.0" encoding="UTF-8"?>
<root xmlns="http://www.example.com/ns1">
  <element1>This is element 1 from the default
namespace</element1>
  <ns2:element2
xmlns:ns2="http://www.example.com/ns2">This is element 2
from namespace ns2</ns2:element2>
</root>
```

In this case, the default namespace is set using xmlns without a prefix. The element element1 is associated with the default namespace, while element2 uses the ns2 prefix to indicate its namespace.

XML HttpRequest

All modern browsers have a built-in XMLHttpRequest object to request data from a server.

The XMLHttpRequest object can be used to request data from a web server.
The XMLHttpRequest object is **a developers dream**, because you can:

- Update a web page without reloading the page
- Request data from a server - after the page has loaded
- Receive data from a server - after the page has loaded
- Send data to a server - in the background

Introduction XML(cont..)

XML Validation

• A well formed XML document can be validated against DTD or Schema.

• A well-formed XML document is an XML document with correct syntax.

• It is very necessary to know about valid XML document before knowing XML validation.

• It must be well formed (satisfy all the basic syntax condition)

• It should be behave according to predefined DTD or XML schema

Introduction XML(cont..)

Rules for well formed XML

- It must begin with the XML declaration.
- It must have one unique root element.
- All start tags of XML documents must match end tags.
- XML tags are case sensitive.
- All elements must be closed.
- All elements must be properly nested.
- All attributes values must be quoted.

Introduction XML(cont..)

Example Of XML

```
<?xml version="1.0"?>
<employee>
  <firstname>vimal</firstname>
  <lastname>jaiswal</lastname>
<address>Gaziabad</address>
  <email>vima.jaiswal@gmail.com</email>
</employee>
```

Daily Quiz

1: What does XML stand for?
a) eXtensible Markup Language
b) Extra Markup Language
c) External Markup Language
d) Extended Markup Language
Answer: a) eXtensible Markup Language

2: Which character is used to indicate the start and end of an XML element?
a) <
b) >
c) /
d) &
Answer: a) <

3: Which is the correct syntax for defining an XML declaration?
a) <xml version="1.0">
b) <?xml version="1.0" ?>
c) <xml version="1.0" ?>
d) <?xml version="1.0">
Answer: b) <?xml version="1.0" ?>

Daily Quiz

4: How are attributes defined in XML?
a) Inside square brackets: [attribute="value"]
b) Inside angle brackets: <attribute="value">
c) Inside double quotes: attribute="value"
d) Inside parentheses: (attribute="value")
Answer: c) Inside double quotes: attribute="value"

5: Which symbol is used to separate the namespace prefix and element name in XML?
a) :
b) .
c) /
d) _
Answer: a) :

6: What is the purpose of XML Schema (XSD)?
a) To define the structure and data types of XML documents
b) To provide styling and presentation for XML documents
c) To define the display layout of XML documents
d) To transform XML documents into HTML
Answer: a) To define the structure and data types of XML documents

Introduction XML(cont..)

XML Document Object Model (DOM)

- It defines a standard way to access and manipulate documents.

- The Document Object Model (DOM) is a programming API for HTML and XML documents.

- It defines the logical structure of documents and the way a document is accessed and manipulated.

- XML DOM defines a standard way to access and manipulate XML documents.

Introduction XML(cont..)

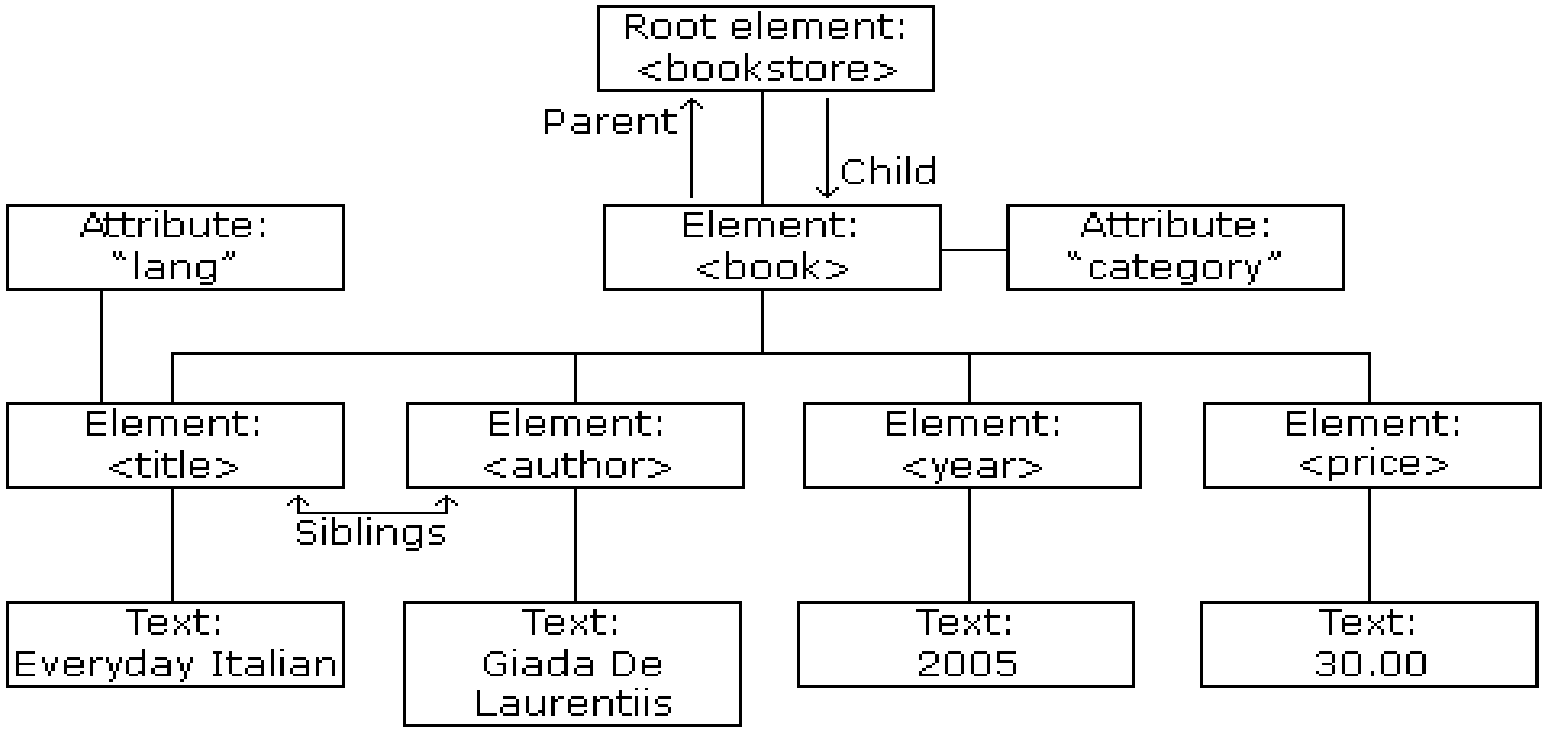
- **XML Document Object Model (DOM)(cont..)**

  - **The XML DOM is:**

    - A standard object model for XML
    - A standard programming interface for XML
    - Platform- and language-independent
    - A W3C standard

Introduction XML(cont..)

- **Example Of XML DOM**



Introduction XML(cont..)

- **Example Of XML DOM(cont..)**

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

<p id="demo"></p>

<script>
var xhttp = new XMLHttpRequest();
xhttp.onreadystatechange = function() {
    if (this.readyState == 4 && this.statu
s == 200) {
    myFunction(this); } }
```

Introduction XML(cont..)

- **Example Of XML DOM(cont..)**

```
xhttp.open("GET", "books.xml", true);
xhttp.send();

function myFunction(xml) {
   var xmlDoc = xml.responseXML;
   document.getElementById("demo").innerHTML =
   xmlDoc.getElementsByTagName("title")[0].childNodes[0].node
Value;
}
</script>
</body>
</html>
```

Topic Objective

Students were able to learn:

- Basics of XML
- XML syntax
- XML Document
- XML Namespace
- XML Validation
- Basics of Document Object Model.

Introduction XML(cont..)

- **Presenting and using XML**

• Presenting XML is a Java web application framework for presenting HTML, PDF, WML etc. in a device independent manner.

• It aims to achieve a complete separation of content and presentation.

• It supports various kinds of content including XML files, dynamic content, SQL result sets and flat files.

• Presenting XML may be used as a command line tool or as a framework for a servlet-based web application

Introduction XML(cont..)

XML Processors

• When a software program reads an XML document and takes actions accordingly, this is called *processing* the XML.

• Any program that can read and process XML documents is known as an *XML processor*.

• An XML processor reads the XML file and turns it into in-memory structures that the rest of the program can access.

•This is called a *parser*, and it is an important component of every XML processing program.

Introduction XML(cont..)

- **XML Parsers**

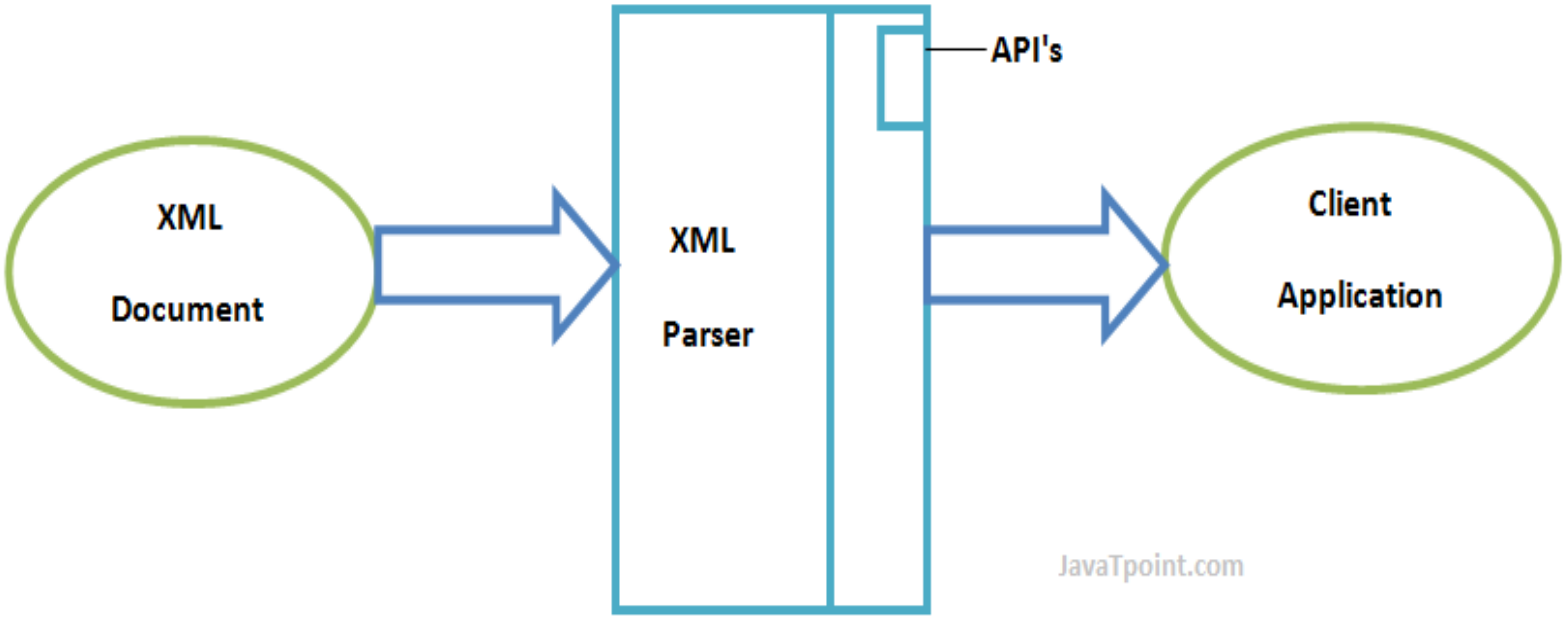
  - An XML parser is a software library or package that provides interfaces for client applications to work with an XML document.

  - The XML Parser is designed to read the XML and create a way for programs to use XML.

  - XML parser validates the document and check that the document is well formatted.

Introduction XML(cont..)

- **XML Parsers(cont..)**



Introduction XML(cont..)

- **DOM (Document Object Model) parser**

  - A DOM document is an object which contains all the information of an XML document.

  - It is composed like a tree structure.

  - The DOM Parser implements a DOM API.

  - This API is very simple to use.

Introduction XML(cont..)

- **DOM (Document Object Model) parser(cont..)**

Advantages

- It supports both read and write operations and the API is very simple to use.

- It is preferred when random access to widely separated parts of a document is required.

Disadvantages

- It consumes more memory because the whole XML document needs to loaded into memory.

- It is comparatively slower than other parsers.

Introduction XML(cont..)

- **(Simple API for XML) parser**

• This API is an event based API and less intuitive.

• It does not create any internal structure.

• Clients does not know what methods to call, they just overrides the methods of the API and place his own code inside method.

• It is an event based parser, that works like an event handler in Java.

Daily Quiz

1: What does DOM stand for in XML?
a) Document Object Model
b) Data Object Model
c) Document Oriented Model
d) Data Oriented Model
Answer: a) Document Object Model

2: Which programming languages can be used to manipulate XML using DOM?
a) JavaScript
b) Python
c) Java
d) All of the above
Answer: d) All of the above

3: What is the purpose of XML DOM?
a) To represent an XML document as a tree structure
b) To validate XML documents against a schema
c) To transform XML documents into HTML
d) To serialize XML documents into a string representation
Answer: a) To represent an XML document as a tree structure

Daily Quiz

4: What is an XML parser?
a) A software tool used to read and process XML documents
b) A programming language used to write XML documents
c) A markup language used to define XML schema
d) A web browser used to render XML documents
Answer: a) A software tool used to read and process XML documents

5: Which type of XML parser reads the entire XML document into memory before processing?
a) Event-based parser
b) Tree-based parser
c) SAX parser
d) DOM parser
Answer: d) DOM parser
6: Which type of XML parser processes XML documents sequentially and triggers events as it encounters different elements?
a) Event-based parser
b) Tree-based parser
c) SAX parser
d) DOM parser
Answer: c) SAX parser

Introduction XML(cont..)

- **(Simple API for XML) parser(cont..)**

Advantages

- It is simple and memory efficient.
- It is very fast and works for huge documents.

Disadvantages

- It is event-based so its API is less intuitive.
- Clients never know the full information because the data is broken into pieces.

Cont…

XML documents are well-formed when they adhere to specific syntax rules, such as properly nested tags and correct use of attributes. Additionally, XML documents can be validated against Document Type Definitions (DTDs) or XML Schemas to ensure their structure and content integrity.

XML is often used in conjunction with other technologies, such as XSLT (eXtensible Stylesheet Language Transformations) for transforming XML data into different formats, and XPath for querying and selecting specific parts of an XML document.

XML Display

Raw XML files can be viewed in all major browsers.
Don't expect XML files to be displayed as HTML pages.

Viewing XML Files:

```
<?xml version="1.0" encoding="UTF-8"?>
- <note>
  <to>Tove</to>
  <from>Jani</from>
  <heading>Reminder</heading>
  <body>Don't forget me this weekend!</body>
</note>
```

XML DISPLAY

Look at the XML file above in your browser: note.xml

Most browsers will display an XML document with color-coded elements.

Often a plus (+) or minus sign (-) to the left of the elements can be clicked to expand or collapse the element structure.

To view raw XML source, try to select "View Page Source" or "View Source" from the browser menu.

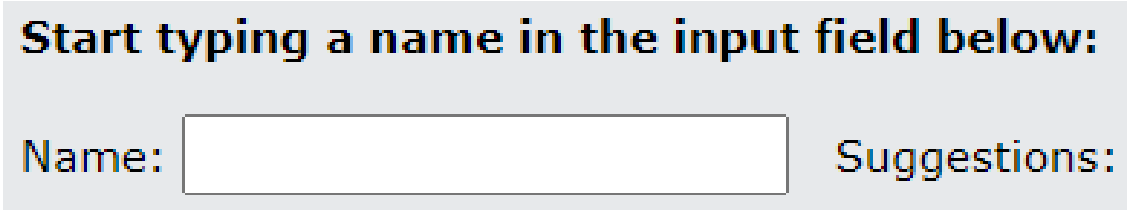
Note: In Safari 5 (and earlier), only the element text will be displayed. To view the raw XML, you must right click the page and select "View Source".

XML :HTTP Request

The XMLHttpRequest object can be used to request data from a web server.
The XMLHttpRequest object is **a developers dream**, because you can:

- Update a web page without reloading the page
- Request data from a server - after the page has loaded
- Receive data from a server - after the page has loaded
- Send data to a server - in the background

When you type a character in the input field below, an XMLHttpRequest is sent to the server, and some name suggestions are returned (from the server):



XML :HTTP Request

Sending an XMLHttpRequest:

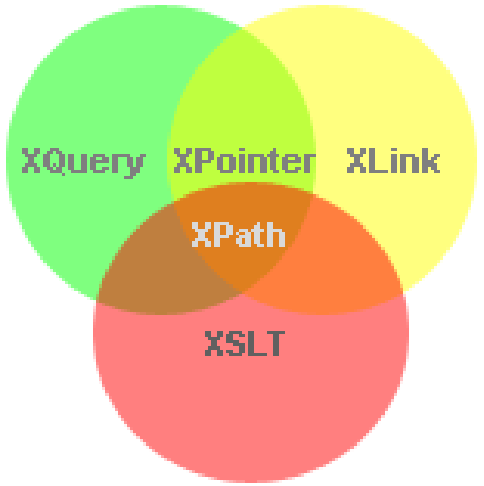
A common JavaScript syntax for using the XMLHttpRequest object looks much like this:

```
var xhttp = new XMLHttpRequest();
xhttp.onreadystatechange = function() {
    if (this.readyState == 4 && this.status == 200) {
       // Typical action to be performed when the document is ready:
       document.getElementById("demo").innerHTML = xhttp.responseText;
    }
};
xhttp.open("GET", "filename", true);
xhttp.send();
```

XML :XPath

XPath is a major element in the XSLT standard.
XPath can be used to navigate through elements and attributes in an XML document.

- XPath is a syntax for defining parts of an XML document
- XPath uses path expressions to navigate in XML documents
- XPath contains a library of standard functions
- XPath is a major element in XSLT and in XQuery
- XPath is a W3C recommendation



XML :Xpath (Cont..)

XPath Path Expressions

XPath uses path expressions to select nodes or node-sets in an XML document.
These path expressions look very much like the expressions you see when you work with a traditional computer file system.
XPath expressions can be used in JavaScript, Java, XML Schema, PHP, Python, C and C++, and lots of other languages.

XPath is Used in XSLT

XPath is a major element in the XSLT standard.
With XPath knowledge you will be able to take great advantage of XSL.

XML :Xpath (Cont..)

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

<bookstore>

<book category="cooking">
  <title lang="en">Everyday Italian</title>
  <author>Giada De Laurentiis</author>
  <year>2005</year>
  <price>30.00</price>
</book>

<book category="children">
  <title lang="en">Harry Potter</title>
  <author>J K. Rowling</author>
  <year>2005</year>
  <price>29.99</price>
</book>

<book category="web">
  <title lang="en">XQuery Kick Start</title>
  <author>James McGovern</author>
  <author>Per Bothner</author>
  <author>Kurt Cagle</author>
  <author>James Linn</author>
  <author>Vaidyanathan Nagarajan</author>
  <year>2003</year>
  <price>49.99</price>
</book>
<book category="web">
  <title lang="en">Learning XML</title>
  <author>Erik T. Ray</author>
  <year>2003</year>
  <price>39.95</price>
</book>
</bookstore>
```

XML :Xpath (Cont..)

In the table below we have listed some XPath expressions and the result of the expressions:

XPath Expression	Result
/bookstore/book[1]	Selects the first book element that is the child of the bookstore element
/bookstore/book[last()]	Selects the last book element that is the child of the bookstore element
/bookstore/book[last()-1]	Selects the last but one book element that is the child of the bookstore element
/bookstore/book[position()<3]	Selects the first two book elements that are children of the bookstore element
//title[@lang]	Selects all the title elements that have an attribute named lang
//title[@lang='en']	Selects all the title elements that have a "lang" attribute with a value of "en"

XML :Xpath (Cont..)

In the table below we have listed some XPath expressions and the result of the expressions:

XPath Expression	Result
/bookstore/book[price>35.00]	Selects all the book elements of the bookstore element that have a price element with a value greater than 35.00
/bookstore/book[price>35.00] /title	Selects all the title elements of the book elements of the bookstore element that have a price element with a value greater than 35.00

XML :XSLT

XSLT (eXtensible Stylesheet Language Transformations) is the recommended style sheet language for XML.

XSLT is far more sophisticated than CSS. With XSLT you can add/remove elements and attributes to or from the output file. You can also rearrange and sort elements, perform tests and make decisions about which elements to hide and display, and a lot more.

XSLT uses XPath to find information in an XML document.

XML :XSLT (Cont..)

```
<?xml version="1.0" encoding="UTF-8"?>
<breakfast_menu>
<food>
<name>Belgian Waffles</name>
<price>$5.95</price>
<description>Two of our famous Belgian
Waffles with plenty of real maple
syrup</description>
<calories>650</calories>
</food>
<food>
<name>Strawberry Belgian Waffles</name>
<price>$7.95</price>
<description>Light Belgian waffles covered
with strawberries and whipped
cream</description>
<calories>900</calories>
</food>
<food>
<name>Berry-Berry Belgian Waffles</name>
<price>$8.95</price>
<description>Light Belgian waffles covered with an
assortment of fresh berries and whipped
cream</description>
<calories>900</calories>
</food>
<food>
<name>French Toast</name>
<price>$4.50</price>
<description>Thick slices made from our homemade
sourdough bread</description>
<calories>600</calories>
</food>
<food>
<name>Homestyle Breakfast</name>
<price>$6.95</price>
<description>Two eggs, bacon or sausage, toast, and
our ever-popular hash browns</description>
<calories>950</calories>
</food>
</breakfast_menu>
```

XML :XSLT (Cont..)

Use XSLT to transform XML into HTML, before it is displayed in a browser:

```
<?xml version="1.0" encoding="UTF-8"?>
<html xsl:version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<body style="font-family:Arial;font-size:12pt;background-color:#EEEEEE">
<xsl:for-each select="breakfast_menu/food">
  <div style="background-color:teal;color:white;padding:4px">
    <span style="font-weight:bold"><xsl:value-of select="name"/> - </span>
    <xsl:value-of select="price"/>
    </div>
  <div style="margin-left:20px;margin-bottom:1em;font-size:10pt">
    <p>
    <xsl:value-of select="description"/>
    <span style="font-style:italic"> (<xsl:value-of select="calories"/> calories per
serving)</span>
    </p>
  </div>
</xsl:for-each>
</body>
</html>
```

Daily Quiz

1: What is XPath used for in XML?
a) To query and navigate XML documents
b) To transform XML into HTML
c) To define the structure and data types of XML documents
d) To validate XML documents against a schema
Answer: a) To query and navigate XML documents

2: Which symbol is used to select the root node in XPath?
a) .
b) /
c) //
d) *
Answer: b) /

3: Which XPath expression selects all elements with a specific tag name?
a) //elementName
b) /elementName
c) //*elementName
d) elementName
Answer: a) //elementName

Daily Quiz

4: What does XSLT stand for?
a) XML Style Language Transformation
b) eXtensible Stylesheet Language Transformation
c) XML Structure Language Transformation
d) eXtensible Style Language Translator
Answer: b) eXtensible Stylesheet Language Transformation

5: What is the primary purpose of XSLT?
a) To transform XML documents into different formats (e.g., HTML, PDF)
b) To define the structure and data types of XML documents
c) To query and navigate XML documents
d) To validate XML documents against a schema
Answer: a) To transform XML documents into different formats (e.g., HTML, PDF)

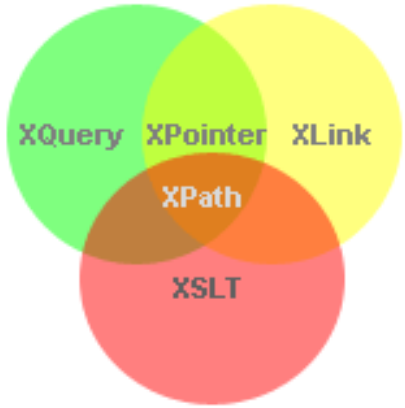
6: Which element is used in XSLT to match and select nodes for transformation?
a) <template>
b) <output>
c) <apply-templates>
d) <stylesheet>
Answer: c) <apply-templates>

XML :XQuery

•XQuery is to XML what SQL is to databases.
•XQuery was designed to query XML data.

```
for $x in doc("books.xml")/bookstore/book
where $x/price>30
order by $x/title
return $x/title
```

•XQuery is ***the*** language for querying XML data
•XQuery for XML is like SQL for databases
•XQuery is built on XPath expressions
•XQuery is supported by all major databases
•XQuery is a W3C Recommendation



XML :XQuery (Cont..)

XQuery is About Querying XML

XQuery is a language for finding and extracting elements and attributes from XML documents.

Here is an example of what XQuery could solve:

"Select all CD records with a price less than $10 from the CD collection stored in cd_catalog.xml"

XQuery and XPath

XQuery 1.0 and XPath 2.0 share the same data model and support the same functions and operators. If you have already studied XPath you will have no problems with understanding XQuery.

XQuery - Examples of Use

XQuery can be used to:

Extract information to use in a Web Service

Generate summary reports

Transform XML data to XHTML

Search Web documents for relevant information

XML :XLink

XLink is used to create hyperlinks in XML documents.

- XLink is used to create hyperlinks within XML documents
- Any element in an XML document can behave as a link
- With XLink, the links can be defined outside the linked files
- XLink is a W3C Recommendation

XLink Syntax

In HTML, the <a> element defines a hyperlink. However, this is not how it works in XML. In XML documents, you can use whatever element names you want - therefore it is impossible for browsers to predict what link elements will be called in XML documents.

Below is a simple example of how to use XLink to create links in an XML document:

XML :Xlink (Cont..)

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

<homepages xmlns:xlink="http://www.w3.org/1999/xlink">
  <homepage xlink:type="simple" xlink:href="https://www.w3schools.com">Visit
W3Schools</homepage>
  <homepage xlink:type="simple" xlink:href="http://www.w3.org">Visit W3C</homepage>
</homepages>
```

XML :Xlink (Cont..)

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

<bookstore xmlns:xlink="http://www.w3.org/1999/xlink">

<book title="Harry Potter">
  <description
  xlink:type="simple"
  xlink:href="/images/HPotter.gif"
  xlink:show="new">
  As his fifth year at Hogwarts School of Witchcraft and
  Wizardry approaches, 15-year-old Harry Potter is.....
  </description>
</book>
```

```
<book title="XQuery Kick Start">
  <description
  xlink:type="simple"
  xlink:href="/images/XQuery.gif"
  xlink:show="new">
  XQuery Kick Start delivers a concise introduction
  to the XQuery standard.....
  </description>
</book>

</bookstore>
```

XML :Xlink (Cont..)

Example explained:

The XLink namespace is declared at the top of the document (xmlns:xlink="http://www.w3.org/1999/xlink")

The xlink:type="simple" creates a simple "HTML-like" link

The xlink:href attribute specifies the URL to link to (in this case - an image)

The xlink:show="new" specifies that the link should open in a new window

Topic Objective

Students were able to learn:
XML HttpRequest
XML Parser
XML SAX
XML Xsl/Xslt
Xquerry
XPath

Daily Quiz

1: What is XQuery used for in XML?
a) To query and extract data from XML documents
b) To define the structure and data types of XML documents
c) To transform XML into different formats
d) To validate XML documents against a schema
Answer: a) To query and extract data from XML documents

2: Which programming language is XQuery most closely related to?
a) SQL (Structured Query Language)
b) JavaScript
c) Java
d) Python
Answer: a) SQL (Structured Query Language)

3: Which keyword is used in XQuery to select elements?
a) SELECT
b) FROM
c) WHERE
d) FOR
Answer: d) FOR

Daily Quiz

4: What is XLink used for in XML?
a) To define relationships between different XML documents
b) To transform XML into different formats
c) To validate XML documents against a schema
d) To query and extract data from XML documents
Answer: a) To define relationships between different XML documents

5: Which attribute is used in XLink to create a hyperlink?
a) href
b) link
c) target
d) url
Answer: a) href

6: Which element is used in XLink to define the start and end points of a link?
a) <link>
b) <source>
c) <locator>
d) <arc>
Answer: d) <arc>

XML :Validator

An XML document with correct syntax is called "Well Formed".

The syntax rules were described in the previous chapters:

- XML documents must have a root element
- XML elements must have a closing tag
- XML tags are case sensitive
- XML elements must be properly nested
- XML attribute values must be quoted

```
<?xml version="1.0" encoding="UTF-8"?>
<note>
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend!</body>
</note>
```

XML :Server

XML files are plain text files just like HTML files.
XML can easily be stored and generated by a standard web server.

Storing XML Files on the Server

XML files can be stored on an Internet server exactly the same way as HTML files.
Start Windows Notepad and write the following lines:

```
<?xml version="1.0" encoding="UTF-8"?>
<note>
  <from>Jani</from>
  <to>Tove</to>
  <message>Remember me this weekend</message>
</note>
```

XML DTD

XML DTD

• A DTD defines the legal elements of an XML document.

• XML schema is a XML based alternative to DTD.

• Actually DTD and XML schema both are used to form a well formed XML document.

• DTD stands for Document Type Definition.

• It defines the legal building blocks of an XML document.

Cont…

XML DTD Syntax

• The XML Document Type Declaration, commonly known as DTD, is a way to describe XML language precisely.

• DTDs check vocabulary and validity of the structure of XML documents against grammatical rules of appropriate XML language.

```
<!DOCTYPE element DTD identifier
[
  declaration1
  declaration2
  ........
]>
```

Cont…

Internal DTD Declaration XML

• A DTD is referred to as an internal DTD if elements are declared within the XML files.

• To refer it as internal DTD, *standalone* attribute in XML declaration must be set to **yes**.

• This means, the declaration works independent of external source.

(cont..)

Internal DTD Declaration XML Example

```
<?xml version="1.0" standalone="yes" ?>
<!DOCTYPE address [
  <!ELEMENT address (name,company,phone)>
  <!ELEMENT name (#PCDATA)>
  <!ELEMENT company (#PCDATA)>
  <!ELEMENT phone (#PCDATA)>

<address>
  <name>Devendra Kumar</name>
  <company>NIET</company>
  <phone>(011) 123-4567</phone>
</address>
```

(cont..)

External DTD

- In external DTD elements are declared outside the XML file.
- They are accessed by specifying the system attributes which may be either the legal *.dtd* file or a valid URL.
- To refer it as external DTD, *standalone* attribute in the XML declaration must be set as **no**.
- This means, declaration includes information from the external source.

(cont..)

External DTD Example

```
<?xml version="1.0" standalone="no" ?>
<!DOCTYPE address SYSTEM"address.dtd"><address>  <name>Anand
Varshney</name><company>NIET</company>  <phone>(011) 123-
4567</phone></address>
```

address.dtd

```
<!ELEMENT address (name,company,phone)><!ELEMENT
name (#PCDATA)><!ELEMENT company
(#PCDATA)><!ELEMENT phone (#PCDATA)>
```

XML: Schema

XML schema

- It is defined as an XML language.
- It uses namespaces to allow for reuses of existing definitions
- It supports a large number of built in data types and definition of derived data types
- XML Schema is commonly known as XML Schema Definition (XSD)

- **Syntax**

```
<xs:schema xmlns:xs="http://www.org/2001/XMLSchema">
```

(cont..)

- **XML schema Example**

```
<?xml version="1.0" encoding="UTF-8"?><xs:schema
xmlns:xs="http://www.w3.org/2001/XMLSchema"><xs:element
name="contact">    <xs:complexType>        <xs:sequence>
<xs:element name="name" type="xs:string" />          <xs:element
name="company" type="xs:string" />          <xs:element
name="phone" type="xs:int" />        </xs:sequence>
</xs:complexType></xs:element>
```

(cont..)

- **Ways to adefine XML schema elements**
  - Simple Type
  - Complex Type
  - Global Types
  - **XML Simple Types Example**

```
<xs:element name="phone_number" type="xs:int" />
```

(cont..)

- **XML Complex Types Example**

```
<xs:element
name="Address"><xs:complexType><xs:sequence>
<xs:element name="name" type="xs:string" />
<xs:element name="company" type="xs:string" />
<xs:element name="phone" type="xs:int" />
</xs:sequence>     </xs:complexType></xs:element>
```

(cont..)

- **XML Global Types Example**

```
<xs:element name="Address1">
<xs:complexType>
<xs:sequence>
<xs:element name="address"
type="AddressType" />
<xs:element name="phone1"
type="xs:int" />
</xs:sequence>
</xs:complexType></xs:element
> <xs:element
name="Address2">
<xs:complexType>
<xs:sequence>
<xs:element name="address"
type="AddressType" />
<xs:element name="phone2"
type="xs:int" />
</xs:sequence>
</xs:complexType></xs:element
>
```

Topic Objective

Students were able to learn:

XML Validator
XML Schema
XML DTDs
With their examples

Previous Topics: Recap

- The topic was focused on CSS concepts in XML which includes
  - Introduction
  - XML Examples
  - Syntax
  - Features
  - Benefits
- Validations and its rules with examples
- XML Schema with its syntax.
- XML DTD and its
  - Syntax
  - Types
  - Declaration