Web techunit-2 by brevilearning yt compressed

B.tech (Dr. A.P.J. Abdul Kalam Technical University)

Web Technology

UNIT - 2 [One Shot]

Topics to be covered :

1. HTML : Lists, Tables, Images, Forms, CSS
2. HTML : Frames, DTD
3. XML : DTD, Object models.
4. XML schemes, presenting and using XML.
5. XML Processors : DOM and SAX
6. Dynamic HTML

HTML frames :

- It is a technique for dividing a webpage layout into multiple sections.
- Each section, called a frame, could display a separate HTML document, allowing for a more structured and compartmentalized experience.

Core Elements :

- <frameset> : This element defines the overall layout of the frames within the browser window.

  It replaces the standard <body> tag in a frameset document.

- <frame> : This element defines an individual frame within the <frameset>.

Layout and Attributes :

- Frames can be arranged horizontally using the 'rows' attribute of the <frameset> tag.
- Vertical arrangements are acheived using the 'cols' attribute.
- Additional attributes for <frame> tags control aspects like scrolling, margins, borders, and frame resizing.

* Disadvantage :

- Can cause problems for users with screen readers and make navigation challenging.
- Search engine struggles to index content within frames effectively.
- Less responsiveness

* Alternative of Frames :

⇒ CSS : - Cascading Style Sheet layouts offers a more flexible and accessible approach to website layout.

- It is more responsive across devices even with multisectional layouts.

HTML DTD / XML DTD :

* HTML DTD [Document Type Definition]

- Defines the structure of HTML doc
- No longer used in modern HTML5 development.
- Specified legal elements like <h1>, <p>, <img> and their attributes like 'href', 'align', etc.
- It provided a way to ensure valid HTML structure, promoting consistency and inter-operability between browsers.
- Not very flexible or expressive. Couldn't handle complex data types or data validation.

* XML DTD [Document Type Definition]

- Defines the structure and legal elements / attributes of XML documents.
- Similar to HTML DTD but more powerful. Defines elements, their content models and attributes with data type restrictions.
- Enforces data structure and helps with validation.

- Promotes data integrity and easier processing by applications.
- Can become complex for large scale data definitions.

★ Key differences b/w HTML and XML Document Type Definitions.

HTML DTD	XML DTD
• Defines structure of Html document.	• Defines the structure of XML document.
• Limited use	• Widely used
• Lower flexibility	• More flexible
• Limited datatypes.	• Can define data types.
• Basic structure validation.	• Validation of structure and data.

LIKE, SHARE & SUBSCRIBE

XML object model [DOM]:

- It is a collection of objects that you use to access and manipulate the data stored in an XML document.
- It is modeled as a tree, where each element is considered as a node.
- Each node contains the actual data in the document. Each element node can have child nodes, parent node, and sibling nodes as well.

★ Functionalities:

- Access data: allows the retrieval of elements, attributes and text content.
- Manipulation: allows to manipulate the structure of XML document. You can add, remove or edit elements, attributes and text nodes.

- **Navigation:** You can navigate the tree structure of the DOM to find the specific element or traverse entire document

* **Advantages:**

- Provides structured way to work with XML data.
- Enables to read, write, modify and create complex XML structures programmatically.

* **Disadvantages:**

- Memory consuming for large XML document.
- Traversing and manipulating DOM can be slower than raw XML texts for simple tasks.

XML schemas:

- An XML schema is a formal description that defines the legal building blocks and structure of an XML document.
- It acts like a blueprint that defines how elements and attributes can be arranged and what kind of data they can contain.

Why to use XML schemas?

- **Validation:** Validating XML document against XSD schema, errors and inconsistancies can be detected early in the development process.
- **Interoperability:** By defining a common schema, different applications can exchange XML data easily.
- **Documentation:** XSD acts as the formal documentation for

the XML data, making the clear representation and validation of elements, attributes and data types.

- **Reusability:** Schemas can be reused across multiple XML documents, increasing consistency and defining the structure.
- **Extensibility:** Schema offers creating custom data types or extending existing ones, making it adaptable to complex data needs.

★ **How XML Schemas are used?**

- XML Schemas are typically created using dedicated schema languages like XSD (XML Schema Definition).

Workflow:

1. **Define the Schema:** An XSD document is created, specifying the elements, attributes, data types, and constraints for the XML data.
2. **Validate XML documents:** XML documents are validated using an XML validator to ensure that document contains defined structure and data types.
3. **Processing XML data:** Applications that work with XML data can utilize the schema information to understand the structure, data types, and expected values, leading to more efficient and reliable processing.

Presenting and using XML:

- **XML:** · An XML is a document that is not designed to be human readable.
  · It stores data in a structured format using elements and attributes.
- **XSL sheet:** · This is a special set of instructions that

takes the XML data and transform it into a human readable format, like a web-page.

- XSL stylesheet reads the XML document and uses patterns to identify the data it needs.
- It then arranges this data according to formatting rules specified in the Stylesheet.

Advantages of XSL Stylesheet:

- **Presentable data:** XML data is transformed into a user-friendly format, making it easy to understand and handle.
- **Flexibility:** XSL stylesheet allows a wide range of formatting options, enabling you to customize the presentation of the data to your needs.
- **Reusable styles:** Same stylesheet can be applied to multiple XML document, saving time and effort in creating consistent presentation.
- **Ease updation:** XSL keeps the XML data separate from the presentation (XSLT) making it easier to maintain and update each part independently.

---

e.g:

(book.xml):

```
<?xml version="1.0" encoding="UTF-8"?>
<books>
  <book>
    <title> Krishna </title>
    <author> Vishnu </author>
  </book>
</books>
```

(book.xsl):

```
<xsl:stylesheet version="1.0" xmlns:xsl="link">
  <xsl:template match="/">
    <html>
      <body>
```

```
<h1> My Books </h1>
<ul>
 <xsl: for-each select = "books/book">
  <li>
   <xsl: value-of select = "title" />
    - by <xsl: value-of select = "author" />
  </li>
 </xsl: for-each>
</ul>
</body>
</html>
</xsl: template>
</xsl: stylesheet>
```

→ We use XSLT processor to apply this stylesheet to XML :

Result:

```
<! DOCTYPE html>
<html>
<body>
 <h1> My Books </h1>
 <ul>
  <li> Krishna - by Vishnu. </li>
 </ul>
</body>
</html>
```

XML processors :

- XML processors are the tools that helps you to work with XML data in programming languages.
- Two common XML processors are DOM (document object model) and SAX (simple API for XML).

i) DOM [Document Object Model] :

- Builds a complete in-memory representation of entire XML document as a tree structure.
- Each element, attribute, and piece of text becomes a node in the tree. We can access and manipulate these nodes using programming language functions.

* Advantages :

- Easier to navigate and modify the entire document structure.

- Good for random access to any part of the document.
- Well suited for smaller to medium-sized XML file.

★ Disadvantages:

- Requires loading of entire document in memory, which can be resource-intensive for large XML files.
- Slower for processing large documents compared to SAX.

ii) SAX [Simple API for XML]:

- SAX parses the XML document event-by-event. It calls specific functions (event handlers) you define as it encounters elements, attributes, text and other parts of the document.
- Doesn't build a complete in-memory representation. It processes the data sequentially as it reads the file.

★ Advantages:

- Lower memory usage as it doesn't need to store the entire document in memory.
- Faster processing speed for large XML files compared to DOM.

★ Disadvantages:

- More complex to work with document structure manipulation.
- Difficult to navigate back to previous parts of the document for changes.

Dynamic HTML:

- DHTML refers to a combination of technologies that allows for the creation of interactive and animated webpages.

- It is the combination of HTML, CSS, JS and DOM.

* HTML : Provides the foundation or structure of the web page, including elements like headings, paragraphs, images and forms.

* CSS (Cascading StyleSheet) : Defines the presentation of the HTML content, controls layout, fonts, colors and background.

* Java Script : - A scripting language that enables interactivity by manipulating the HTML and CSS after the page has loaded.

- JS allows for dynamic changes to the content and style based on user actions or other events.

* DOM (Document Object Model) :

- Represents the HTML document structure as a tree of objects.

- JS can interact with these objects to modify content, styles, and even add or remove elements entirely.

* Common Uses of DHTML :

- Interactive elements : Creating things like dropdown menus, image sliders that respond to user interaction.

- Animations : Implementing smooth transitions, moving elements, or other visual effects to enhance user experience.

- Real-time updates : Updating content without reloading the entire page, such as live chat features, etc.

- Data validation : Checking user i/p in forms for errors before submission.

Advantages :

- Enhanced user experience.
- Reduced server load.
- Greater flexibility.

Thankyou!! {@brevilearning}

ARFAT Date Page