1. **What is difference between ordered list and definition list?**

**ANS: Ordered List:**

The other popular kind of list is the ordered list. This type of list can be used when the order of items to list is important. It could be that you want to list steps to how to cook a recipe; for this kind of list you could use an ordered list because each step can be emphasized numerically. An ordered list can be created with different styles: Arabic numbers, lowercase or uppercase letters, or lowercase or uppercase Roman numerals. The following shows examples for each of these styles:

**Example:**

**<ol>**

**<li>Beetroot</li>**

**<li>Ginger</li>**

**<li>Potato</li>**

**<li>Radish</li>**

**</ol>**

**This will result in:**

1. Beetroot
2. Ginger
3. Potato
4. Radish

**The type Attribute**

You can use type attribute for <ol> tag to specify the type of numbering you like. By default it is a number. Following are the possible options:

**Example:**

**<ol type="I">**

**<li>Beetroot</li>**

**<li>Ginger</li>**

**<li>Potato</li>**

**<li>Radish</li>**

**</ol>**

**This will result in:**

1. Beetroot
2. Ginger
3. Potato
4. Radish

**Definition List:**

HTML and XHTML support a list style which is called definition lists where entries are listed like in a dictionary or encyclopedia. The definition list is the ideal way to present a glossary, list of terms, or other name/value list.

Definition List makes use of following three tags.

    <dl> - Defines the start of the list

    <dt> - A term

    <dd> - Term definition

    </dl> - Defines the end of the list

1. **What is XSL-FO? List the steps to create a XSL-FO document.**

**ANS: This chapter reviews the roles of the following Recommendations in the XML family and an International Standard in the SGML family, and overviews contexts in which XSLFO is used.**

**Extensible Markup Language (XML)**

**We use XML to express information hierarchically in a sequence of characters according to a vocabulary of element types and their attributes. Using various Recommendations and other industry standards, we can formally describe the makeup and constraints of this vocabulary in different ways to validate the content against our desired document model.**

**Two vocabularies specified in separate W3C Recommendations provide for the two distinct styling processes of transforming and rendering XML instances. The Extensible Stylesheet Language Transformations (XSLT) is a templating markup language used to express how a processor creates a transformed result from an instance of XML information.**

**The Extensible Stylesheet Language Formatting Objects (XSLFO) is a pagination markup language describing a rendering vocabulary capturing the semantics of formatting information for paginated presentation. Formally named Extensible Stylesheet Language (XSL), this Recommendation normatively incorporates the entire XSLT Recommendation by reference and, historically, used to be defined together in a single W3C draft Recommendation.**

**Steps to create XSL-FO document:**

**Know the structure of the XML document**

First, the information about the structure of XML source documents is required. XSLT

processor can transform XML document into XSL-FO without a DTD. But the information

described in the DTD such as types of elements, contents of elements, appearing order of

elements and values of properties are necessary for developing a stylesheet.

**Specify a printing form**

This is the printing form as a final output, in other words the output specification. XSL is

a formatting specification. Printing forms has various range of specifications such as sizes

and layouts of printing paper, layouts of head and body, deciding whether or not to output

index, table of contents, and so on.

**Apply a printing form to formatting objects**

After determining the specification of printing, you have to know what XSL formatting

objects and properties are applied in order to print in this style. It is better to practice how

to specify by referring to a simple stylesheet.

**Develop an XSL stylesheet**

Put the instructions to the stylesheet in order to transform XML source documents into the

target printing form. Map the XML source document to XSL formatting objects that can

generate the output specification. The stylesheet have the similar aspect as the general

programming languages, while it may be difficult if you do not understand the feature of

the XSL.

1. **Explain comparison between XML and Relational database.**

**ANS**: As XML's use increases, it is becoming a common requirement to view traditional relational data as XML. One way of doing this is to provide a virtual XML view over the relational database. This view is provided through the mechanism of annotating an XSD schema that describes the desired XML shape. These annotations contain relational metadata used to translate queries of the XML view into SQL queries, and to transform the SQL query results into the desired XML shape. In this article, we discuss the merits of choosing an XML data model over the more traditional relational data model. We also examine the mapping process and discuss the task of translating between the XML and relational query domains. Finally, we provide an example of mapping an XSD schema to a sample database and querying the virtual XML View.

Traditional relational database systems are excellent mechanisms for storing normalized, highly structured data. However, it is not always easy to query or manipulate this data model. Database developers have traditionally used the mechanism of relational views and stored procedures to abstract away the complexities of accessing the underlying data—in other words, providing a logical relational view. However, as the logical data model becomes increasingly semistructured (data with some variance), providing a relational view to program against becomes increasingly difficult.

The results would be a highly normalized view of customers that would require users to join several tables to get the desired Customer objects. For most experienced DBAs and SQL users, this probably wouldn't be a problem. However, for most application developers, it would probably be easier to program against a logical Customer object without exposing the details of the underlying relational storage mechanism.

<CustomerList>

    <Customer LastName="Jones" FirstName="Jeff">

        <WorkPhone>5556767</WorkPhone>

        <CellPhone>5558888</CellPhone>

        <HomePhone>5553232</HomePhone>

</Customer>

<Customer LastName="Smith" FirstName="Jane">

        <CellPhone>5552444</CellPhone>

        <Pager>5558989</Pager>

</Customer>

</CustomerList>

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Id | Name | Gender | Age | Salary |
| 1 | Kumar | Male | 28 | 20000 |
| 2 | Goal | Male | 30 | 30000 |

1. **Differentiate between Synchronous vs. Asynchronous Data Retrieval with example.**

**ANS: Synchronous –** Script stops and waits for the server to send back a reply before continuing. There are some situations where Synchronous Ajax is mandatory.

In standard Web applications, the interaction between the customer and the server is synchronous. This means that one has to happen after the other. If a customer clicks a link, the request is sent to the server, which then sends the results back.

Because of the danger of a request getting lost and hanging the browser, synchronous javascript isn’t recommended for anything outside of (onbefore)unload event handlers, but if you need to hear back from the server before you can allow the user to navigate away from the page, synchronous Javascript isn’t just your best option.

**Asynchronous –** Where the script allows the page to continue to be processed and will handle the reply if and when it arrives. If anything goes wrong in the request and/or transfer of the file, your program still has the ability to recognize the problem and recover from it.

Processing asynchronously avoids the delay while the retrieval from the server is taking place because your visitor can continue to interact with the web page and the requested information will be processed with the response updating the page as and when it arrives

1. **Explain any five new features of HTML5.**

**ANS:**

**Multimedia Support in HTML5:**

First and foremost feature is the support of multimedia in HTML5. Yes HTML5 supports both audio and video files to be played in a browser. Below is the sample syntax to play audio and video files respectively.

<video src="Video file path" controls>

        Text to be displayed when the browser doesn't support the video goes here.

</video>

<audio src="location of the audio" controls>

        Text to be displayed when the browser doesn't support the audio goes here.

</audio>

**Canvas in HTML5:**

Canvas is a rectangular area, which allows pixel level operations like drawing a line, box, circle, performing graphics, etc. Now HTML5 offers support for canvas areas. Shown below is a sample code.

<span class="user" data-domain="Technology" data-language="C#">

</span>

**Editable Contents in HTML5:**

This is a nice feature, which allows the end users to edit the HTML control's content. This kind of feature allows the developers to build web pages that include sections like notes, HTML editor etc. All you need to do is add an attribute named contentEditable="true" to the HTML control.

<p contenteditable="true">Click here to edit this content!</p>

**HTML5 Autofocus and Placeholder Attributes:**

The autofocus feature is achieved by adding the autofocus attribute. This allows the control to have the focus automatically on page load.

<button id="SubmitButton" autofocus></button>

Placeholder is a feature supported for input fields by adding the attribute placeholder. This feature displays the value provided for the placeholder attribute, like a water mark, until the focus is moved to the input control.

<input type="email" name="EmailTextBox"

      placeholder="Please enter email here..."/>

**Required Field and Range Validators in HTML5:**

Required field functionality is achieved using the attribute named "required" on the input controls. This makes sure that the form will not be posted until the value is entered for the input control.

<input type="email" name="EmailTextBox" required/>

Range validation is achieved through specifying the min and max values for the input control along with a valid type. This makes sure that the form will not be posted until the value entered for the input control is within the range.

<input type="number" name="NumberOfItems" min="10" max="50" required/>

1. **Explain the following: a) Window event attributes. b) Form event attributes.**

        ANS:

**Window Event Attributes:** were set off/happens for the window.

These events should be placed inside the <body>.

These attributes can be added with all the elements to denote an event action.

* onafterprint
* onbeforeprint
* onbeforeunload
* onerror
* onhashchange
* onload
* onmessage
* onoffline
* ononline
* onpagehide
* onpageshow
* onpopstate
* onresize
* onstorage
* onunload

**Form Events:**

Events triggered by actions inside a HTML form (applies to almost all HTML elements, but is most used in form elements):

* onblur
* onchange
* oncontextmenu
* onfocus
* oninput
* oninvalid
* onreset
* onsearch
* onselect
* onsubmit