**1.What is XML?**

* Extensible Markup Language (XML) is the universal language for data on the Web
* XML is a technology which allows us to create our own markup language.
* XML documents are universally accepted as a standard way of representing information in platform and language independent manner.
* XML is universal standard for information interchange.
* XML documents can be created in any language and can be used in any language.

**2.What is the difference between XML and HTML?**

XML is no way clashes with HTML, since they are for two different purposes.

|  |  |
| --- | --- |
| **HTML** | **XML** |
| HTML is for displaying purpose. | whereas  XML is for data representation. |
| HTML is used to mark up text so it can be displayed to users. | XML is used to mark up data so it can be processed by computers. |
| HTML describes both structure (e.g. <p>, <h2>, <em>) and appearance (e.g. <br>, <font>, <i>) | XML describes only content, or “meaning” |
| HTML uses a fixed, unchangeable set of tags | In XML, you make up your own tags |

**3.What are the benefits of XML?**

There are many benefits of using XML on the Web :

* **Simplicity**- Information coded in XML is easy to read and understand, plus it can be processed easily by computers.
* **Openness**- XML is a W3C standard, endorsed by software industry market leaders.
* E**xtensibility** - There is no fixed set of tags. New tags can be created as they are needed.
* **Self-description**- In traditional databases, data records require schemas set up by the database administrator. XML documents can be stored without such definitions, because they contain meta data in the form of tags and attributes.
* **Contains machine-readable context information-** Tags, attributes and element structure provide context information that can be used to interpret the meaning of content, opening up new possibilities for highly efficient search engines, intelligent data mining, agents, etc.
* **Separates content  from presentation**- XML tags describe meaning not presentation. The motto of HTML is: "I know how it looks", whereas the motto of XML is: "I know what it means, and you tell me how it should look." The look and feel of an XML document can be controlled by XSL style sheets, allowing the look of a document to be changed without touching the content of the document. Multiple views or presentations of the same content are easily rendered.
* **Supports multilingual documents and Unicode**-This is important for the internationalization of applications.
* **Facilitates the comparison and aggregation of data** - The tree structure of XML documents allows documents to be compared and aggregated efficiently element by element.
* **Can embed multiple data types** - XML documents can contain any possible data type - from multimedia data (image, sound, video) to active components (Java applets, ActiveX).
* **Can embed existing data** - Mapping existing data structures like file systems or relational databases to XML is simple. XML supports multiple data formats and can cover all existing data structures and .
* **Provides a 'one-server view' for distributed data** - XML documents can consist of nested elements that are distributed over multiple remote servers. XML is currently the most sophisticated format for distributed data - the World Wide Web can be seen as one huge XML database.

|  |
| --- |
|  |

**4.What is a well-formed XML document?**

If a document is syntactically correct it can be called as well-formed XML documents. A well-formed document conforms to XML's basic rules of syntax:

* Every open tag must be closed.
* The open tag must exactly match the closing tag: XML is case-sensitive.
* All elements must be embedded within a single root element.
* Child tags must be closed before parent tags.
* A well-formed document has correct XML tag syntax, but the elements might be invalid for the specified document type.

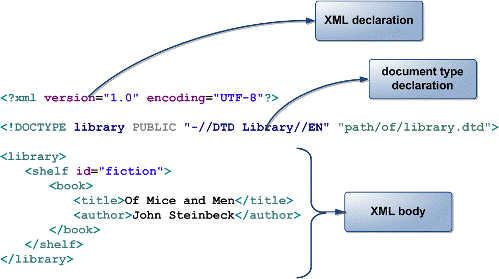
**5.What is a valid XML document?**

If a document is structurally correct then it can be called as valid XML documents. A valid document conforms to the predefined rules of a specific type of document:

* These rules can be written by the author of the XML document or by someone else.
* The rules determine the type of data that each part of a document can contain.

**Note**:*Valid XML document is implicitly well-formed, but well-formed may not be valid*

**6.What is the structure of XML document ?**



**Figure 1: XML Structure**

**7.What is a Processing Instruction in XML?**

A ProcessingIntruction is the information which we would like to give to application. Through a ProcessingInstruction an application would get idea about how to process the document. A ProcessingInstruction can appear anywhere and any no. of times in a document.

**8.How does the XML structure is defined?**

XML document will have a structure which has to be defined before we can create the documents and work with them. The structural rules can be defined using many available technologies, but the following are popular way of doing so-

* Document Type Definition (DTD)
* Schema

**9.What is DTD?**

A **Document Type Definition** (DTD) defines the legal building blocks of an XML document. It defines rules for a specific type of document, including:

* Names of elements, and how and where they can be used
* The order of elements
* Proper nesting and containment of elements
* Element attributes

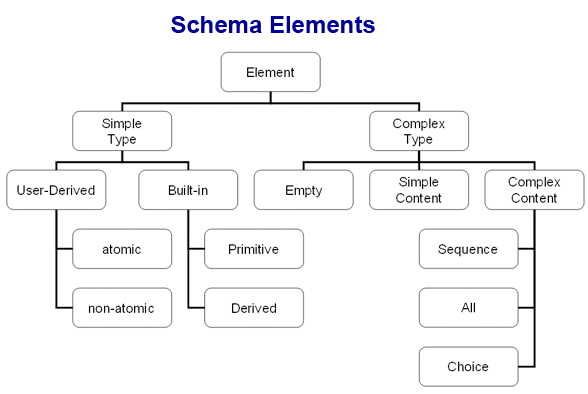
To apply a DTD to an XML document, you can:

* Include the DTD's element definitions within the XML document itself.
* Provide the DTD as a separate file, whose name you reference in the XML document.

**10.What is XML Schema?**

An XML **Schema** describes the structure of an XML instance document by defining what each element must or may contain.XML Schema is expressed in the form of a separate XML file.

* XML Schema provides much more control on element and attribute datatypes.
* Some datatypes are predefined and new ones can be created.
* <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">  
   <xsd:element name="test">  
   <xsd:complexType>



**Figure 2: XML Schema**

**11.What are differences between DTDs and Schema?**

|  |  |
| --- | --- |
| **Schema** | **DTD** |
| Schema document is an XML document i.e., the structure of an XML document is specified by another XML document. | DTDs follow SGML syntax. |
| Schema supports variety of dataTypes similar to programming language. | In DTD everything is treated as text. |
| In Schema,  It is possible to inherit and create relationship among elements. | This is not possible in DTD without invalidating existing documents. |
| In Schema, It is possible to group elements and attributes so that they can be treated as single logical unit. | Grouping of elements and attributes is not possible in DTD. |
| In Schemas, it is possible to specify an upper limit for the number of occurrences of an element | It is not possible to specify an upper limit of an element in DTDs |

|  |
| --- |
|  |

**12.What is a Complex Element?**

A complex element is an XML element that contains other elements and/or attributes.  
There are four kinds of complex elements:

* empty elements
* elements that contain only other elements
* elements that contain only text
* elements that contain both other elements and text

**13.What is a Simple Element?**

A simple element is an XML element that can contain only text.

* A simple element cannot have attributes
* A simple element cannot contain other elements
* A simple element cannot be empty
* However, the text can be of many different types, and may have various restrictions applied to it

|  |
| --- |
|  |

**14.What are namespaces? Why are they important?**

A simple element is an XML element that can contain only text.

* Namespaces are a simple and straightforward way to distinguish names used in XML documents, no matter where they come from.
* XML namespaces are used for providing uniquely named elements and attributes in an XML instance
* They allow developers to qualify uniquely the element names and relationships and make these names recognizable, to avoid name collisions on elements that have the same name but are defined in different vocabularies.
* They allow tags from multiple namespaces to be mixed, which is essential if data is coming from multiple sources.

**Example:** a bookstore may define the <TITLE> tag to mean the title of a book, contained only within the <BOOK> element. A directory of people, however, might define <TITLE> to indicate a person's position, for instance: <TITLE>President</TITLE>. Namespaces help define this distinction clearly.  
  
**Note:** *a) Every namespace has a unique name which is a string. To maintain the uniqueness among namespaces a IRL is most preferred approach, since URLs are unique.*

         b) *Except for no-namespace Schemas, every XML Schema uses at least two namespaces:  
                1.the target namespace.  
                2. The XMLSchema namespace (http://w3.org/2001/XMLSchema)*

**15.What are the ways to use namespaces?**

There are two ways to use namespaces:

* Declare a default namespace
* Associate a prefix with a namespace, then use the prefix in the XML to refer to the namespace

**16.What is the relevance of ElementFormDefault attribute in the Schema?**

ElementFormDefault indicates whether or not locally declared elements must be qualified by the target namespace in an instance document. ElementFormDefault attribute in the Schema has the following relevance:

* **Qualified**: Each and every element of the Schema must be qualified with the namespace in the instance document.
* **Unqualified**: means only globally declared elements must be qualified with there namespace and not the local elements.

**17.What is XML parser?**

An XML parser is a piece of software which can do following:

* Check for well-formedness
* Validate the document
* Allows us to read, create or modify existing XML documents

**Note**: *Parser is piece of software provided by vendors. An XML parser is built in Java runtime from JDK 1.4 onwards*

|  |
| --- |
|  |

**18.What is DOM?**

The Document Object Model (**DOM**) is a platform and language-independent standard object model for representing XML and related formats. DOM is standard API which is not specific to any programming language. DOM represents an XML document as a tree model. The tree model makes the XML document hierarchal by nature. Each and every construct of the XML document is represented as a node in the tree.

**19.What is SAX?**

SAX-Simple API for XML processing. SAX provides a mechanism for reading data from an XML document. It is a popular alternative to the Document Object Model (DOM).SAX provides an event based processing approach unlike DOM which is tree based.

**20.What are the interfaces of SAX?**

The interfaces of SAX are:

* **DocumentHandler**- is used for getting event notification relating to a document.
* **DTDHandler**- is implemented to get the notifications related to declarations in DTD like entities and notations
* **EntityResolver**- is used for reading external entities.
* **ErrorHandler**- is used for handling error related notifications.

**21.What is the difference between SAX parser and DOM parser?**

|  |  |
| --- | --- |
| **SAX** | **DOM** |
| A SAX parser takes the occurrences of components of an input document as events (i.e., event based processing), and tells the client what it reads as it reads through the input document. | A DOM parser creates a tree structure in memory from an input document and then waits for requests from client. |
| No navigation possible (top to bottom only once) | Whereas, we can navigate the DOM tree in any direction, any no. of times. |
| We cannot modify the document content in SAX | We can modify the document content in DOM |
| A SAX parser serves the client application always only with pieces of the document at any given time. | A DOM parser always serves the client application with the entire document no matter how much is actually needed by the client. |
| A SAX parser, however, is much more space efficient in case of a big input document | A DOM parser is space inefficient when the document is huge. |
| Use SAX parser when   * Input document is too big for available memory. * When only a part of the document is to be read and we create the data structures of our own. * If you use SAX, you are using much less memory and performing much less dynamic memory allocation. | Use DOM when   * Your application has to access various parts of the document and using your own structure is just as complicated as the DOM tree. * Your application has to change the tree very frequently and data has to be stored for a significant amount of time. |

|  |
| --- |
|  |

**22.What is a CDATA section in XML?**

CDATA Sections are used to escape blocks of text containing characters which would otherwise be recognized as markup. All tags and entity references are ignored by an XML processor that treats them just like any character data. CDATA blocks have been provided as a convenience measure when you want to include large blocks of special characters as character data, but you do not want to have to use entity references all the time.

**23.What is XSL?**

e**X**tensible **S**tylesheet **L**anguage(**XSL**)  deals with most displaying the contents of XML documents.XSL consists of three parts:

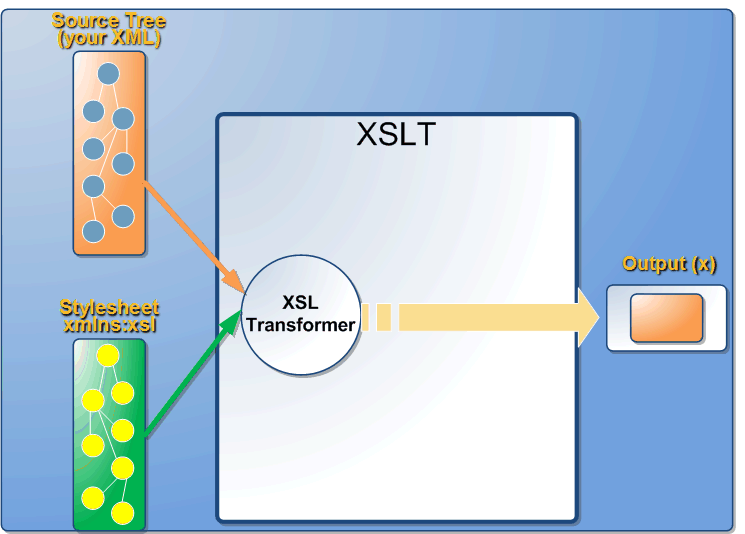
* XSLT - a language for transforming XML documents
* XPath - a language for navigating in XML documents
* XSL-FO - a language for formatting XML documents

**24.How is XSL different from Cascading Style Sheets? Why is a new Stylesheet language needed?**

XSL is compatible with CSS and is designed to handle the new capabilities of XML that CSS can't handle. XSL is derived from Document Style Semantics and Specification Language (DSSSL), a complex Stylesheet language with roots in the SGML community. The syntax of XSL is quite different from CSS, which could be used to display simple XML data but isn't general enough to handle all the possibilities generated by XML. XSL adds the capability to handle these possibilities. For instance, CSS cannot add new items or generated text (for instance, to assign a purchase order number) or add a footer (such as an order confirmation). XSL allows for these capabilities.

**25.What is XSLT?**

eXtensible Stylesheet Language Transformation (XSLT) deals with transformation of one XML document into XHTML documents or to other XML documents. XSLT uses XPath for traversing an XML document and arriving at a particular node.



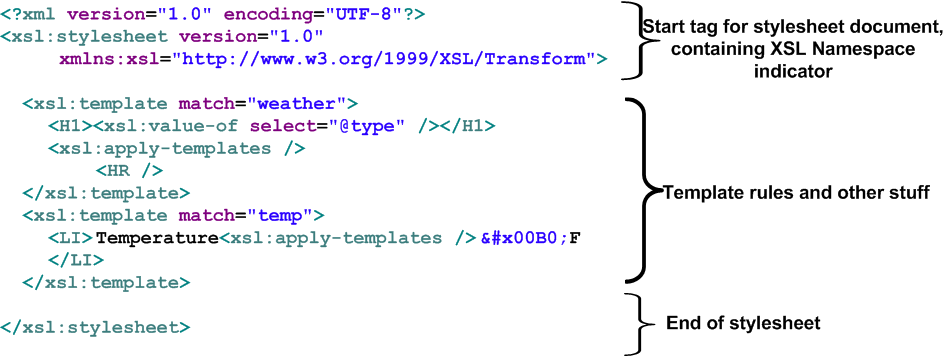
**Figure 3: XSLT**

**26.What is the role of XSL transformer?**

An XSL transformer will transform in the following way:

* The source tree is obtained by parsing in a normal XML style
* The transformation is now applied to the source with the help of information available in Stylesheet.

**27.What is the structure of XSLT?**



**Figure 4: XSLT Structure**

|  |
| --- |
|  |

**28.What is XSL template?**

Template specifies transformation rules. A Stylesheet document can be made up of at least one template, which acts as an entry point. Every template uniquely identifies a particular node in the source tree.

**29.What is XPath?**

XPath is an expression language used for addressing parts of an XML document.  XPath is used to navigate through elements and attributes in an XML document.

**30.What is XSL-FO?**

XSL-FO   deals with formatting XML data. This can be used for generating output in a particular format like XML to PDF, XML to DOC, etc.

**31.How XSL-FO Works (or) How would you produce PDF output using XSL’s?**

