GROW TECH SKILLS

Web Technology Fundamentals

Module Number: 5

Module Name: Introduction to XML

AIM:

To provide the basic concept of creating the website using HTML,CSS & XML

Objectives:

- Measures taken for Creating a Website
- XML Validation document
- Creating interface between the xml document and the client program
- Document storage and processing.

Outcome:

- Fundamentals of XML
- Understanding XML Schema
- Understanding XML Document Object Model
- Accessing and Manipulating XML document **Contents**

1. Why XML?

2. Fundamentals of XML

3. Data Type Definitions

4. XML Schema

5. XML Namespaces

6. XML Parsers

7. XML Document Object Model

Extensible Markup Language (XML)

Use case

1. Consider a business organization that does business at a global scale with clients and businesses all over the world.

2. There is data exchange between these businesses all the time.

3. Different businesses may use different data formats.

4. B2B data hubs often standardise on XML as their data exchange format.

5. XML enables an easy exchange of data.

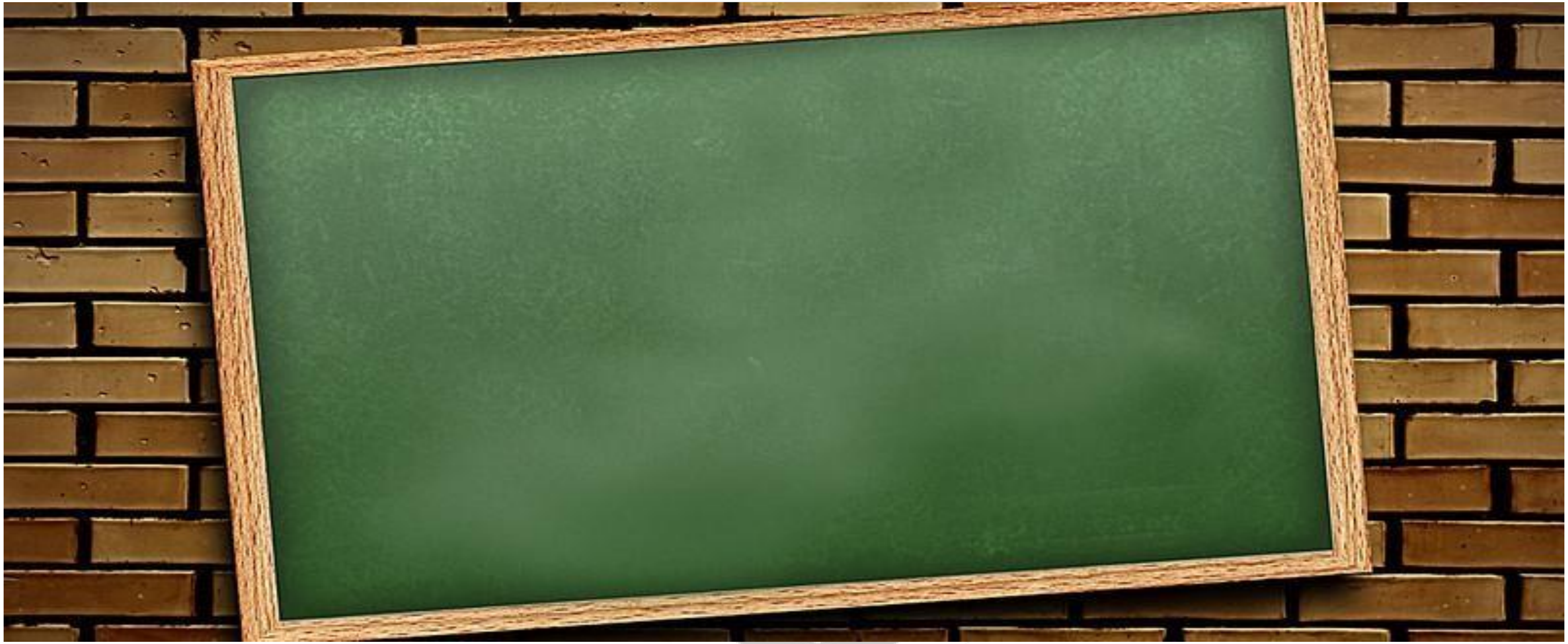
GROW TECH SKILLS

Q & A

What is Extensible Markup language?

A markup language that allows us to create our own tags.

Example of a user defined tag



Why XML?

1. XML documents are easy to create.

2. XML is a Meta-Markup Language. We can makeup tags as we go.

3. XML describes structure and semantics, not formatting.

4. Self describing data.

5. We can interchange data among applications.

6. Suitable for large and complex documents because the data is structured.

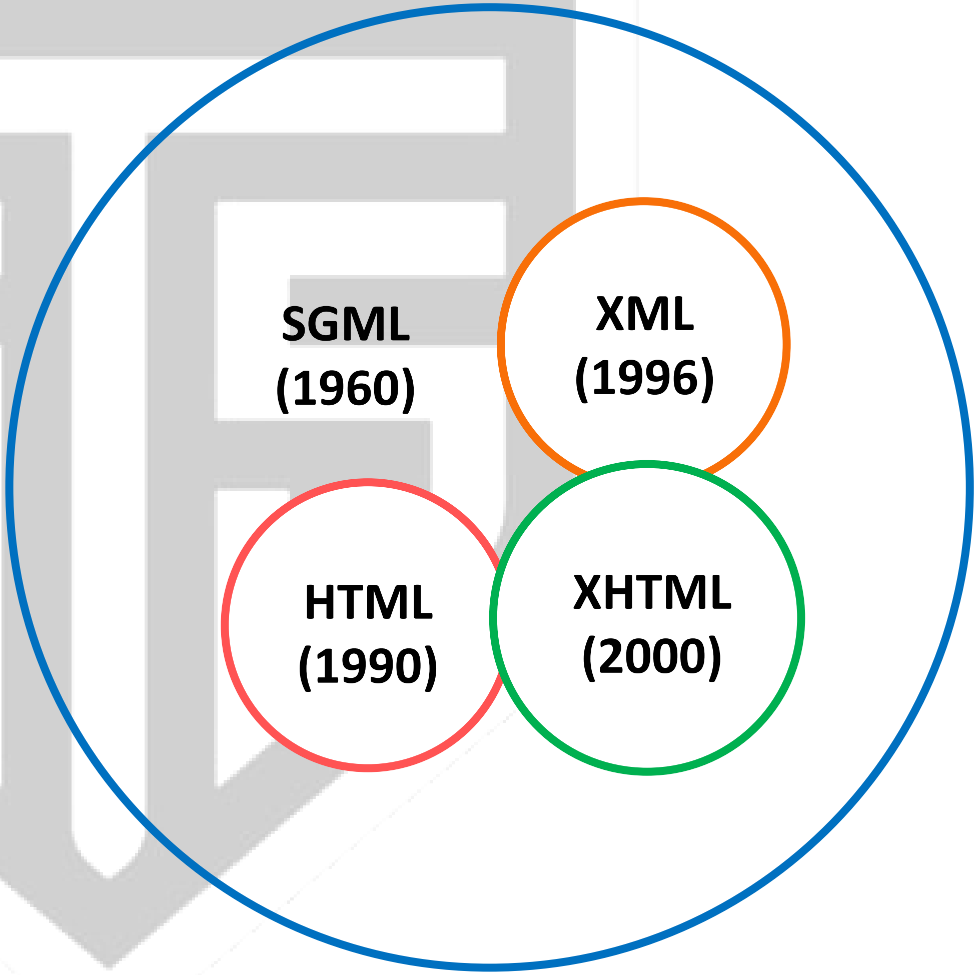
7. XML can be used with existing Web protocols (such as HTTP and MIME) and mechanisms (such as URLs), and it does not impose any additional requirements.

XML-a subset of SGML

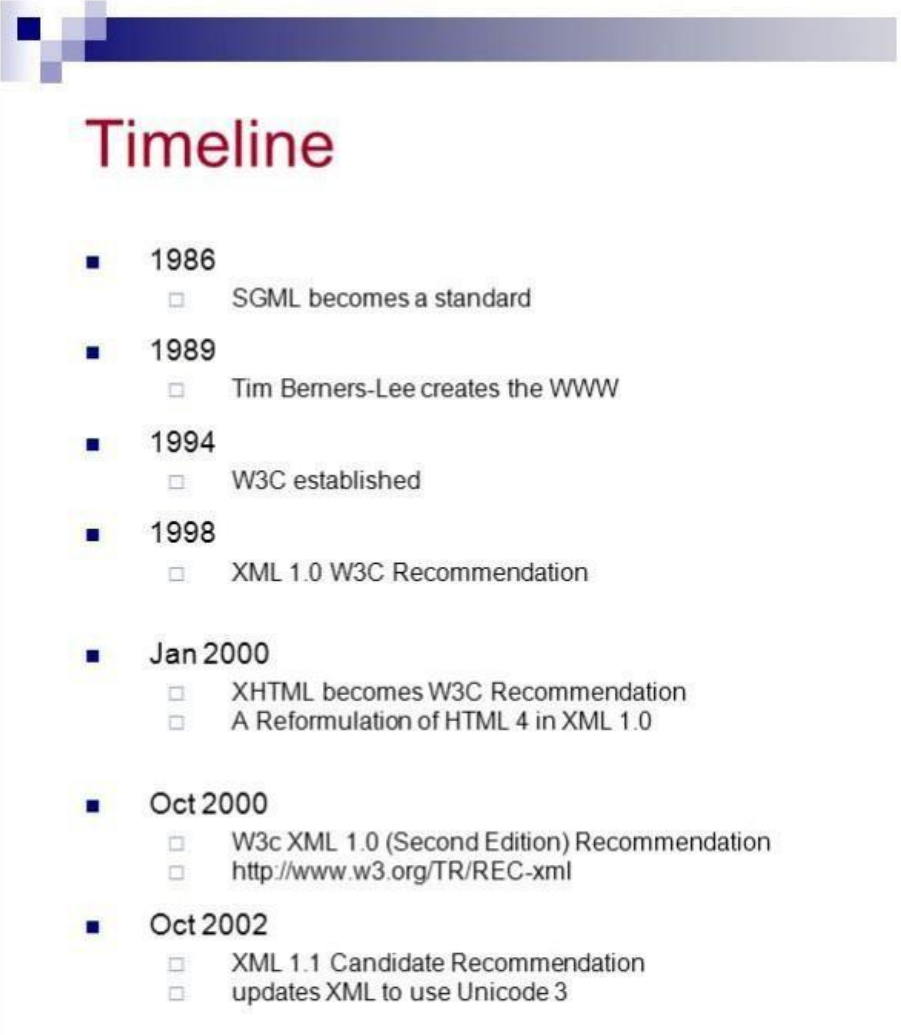
SGML – Standard Generalized Markup Language

HTML – Hyper Text Markup Language

XML – Extensible Markup Language



XML Timeline



Timeline

- 1986
  - □ SGML becomes a standard
- 1989
  - □ Tim Berners-Lee creates the WWW
- 1994
  - □ W3C established
- 1998
  - □ XML 1.0 W3C Recommendation
- Jan 2000
  - □ XHTML becomes W3C Recommendation
  - □ A Reformulation of HTML 4 in XML 1.0
- Oct 2000
  - □ W3c XML 1.0 (Second Edition) Recommendation
  - □ http://www.w3.org/TR/REC-xml
- Oct 2002
  - □ XML 1.1 Candidate Recommendation
  - □ updates XML to use Unicode 3

HTML vs. XML

	HTML	XML
Purpose	HTML was designed to “display the data” with focus on how data looks.	XML was designed to “describe the data”.
Markup	HTML is a markup or presentation language.	XML is a framework for defining markup languages.
Case sensitive	HTML is not case sensitive.	XML is case sensitive.
Structure	HTML has a set of predefined tags.	XML allows the user to define custom tags.
Tags	In HTML, it is not mandatory to close a tag.	In XML every tag must be closed.
Behavior	HTML is static since it is used to display the data.	XML is dynamic. It is used to transport and store the data.

Whitespaces	HTML does not preserve whitespaces.	XML preserves whitespaces.

HTML vs. XML

HTML CodeXML Code

```
<p> Mr. John Smith <address>                              </name>
          </br><name>                                     <street>
  94, coles street,<title>                                  94 Coles Street
             </br>Mr.                                     </street>
            Bangalore   560005 </title>                   <city>Bangalore
      </p><first-name>                                    </city>
                                  John                    <postal-code>
                                 </first-name>              560005
                                 <last-name>              </postal-code>
                                  Smith                  </address>
                                              </last-name>
```

Self Assessment Questions

1. Extensible Mark-up Language is a subset of ____________________.

   a) Standard Generalized Mark-up Language

   b) Hyper Text Mark-up Language

   c) Super Generalized Mark-up Language

   d) Extensible Hyper Text Mark-up Language

Answer: a)

2. XML is suitable for large and complex applications. State true or false.

Answer: False.

3. Select the option that is not true about XML.

   a) XML is a framework for defining markup languages.

   b) XML has a set of pre defined tags.

   c) XML preserves whitespaces.

   d) XML was designed to describe data.

Answer: b)

4. XML was made a W3C recommendation in the year __________________.

   a) 1998

   b) 2000

Answer: a)

Fundamentals of XML

XML Syntax

Syntax:Example:

```
<xml prolog><?xml version="1.0"      encoding="UTF-
                                                      8"?>
<root>
                                                     <refbooks>
 <child>
<subchild>.....</subchild><book             category=”web”>
</child><author> Adrian W. West
                                                     </author>
                              <year>2016</year>
      </root><title> Practical Web                        Design for
                                          Absolute       Beginners
                                                     </title>
                                     <publisher> Apress    </publisher>
                                                     </book>
```

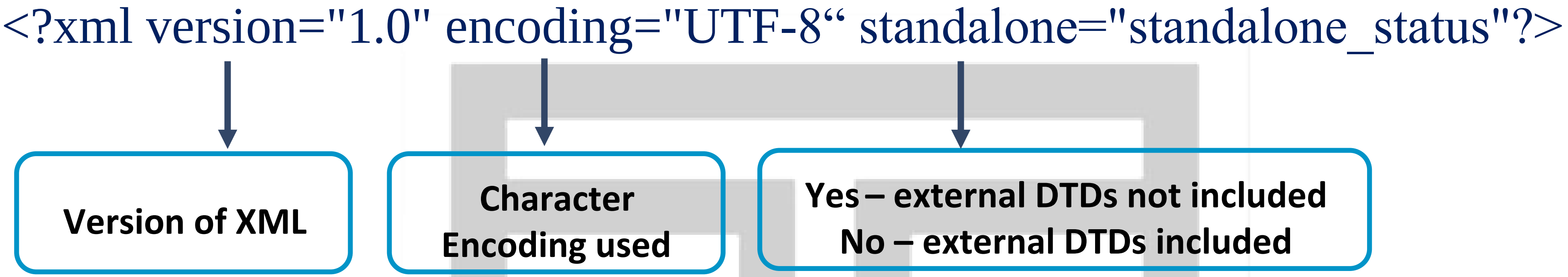
XML Prolog

The **Document Prolog** comes at the top of the document, before the root element.

This section contains:

1. **XML Declaration**
2. **Document Type Declaration (DTD)**

XML Declaration:



<?xml version="1.0" encoding="UTF-8“ standalone="standalone_status"?>

Version of XML

Character Encoding used

Yes – external DTDs not included
No – external DTDs included

XML Syntax Rules

The documents which follow these syntax rules are said to be “well formed” documents.

1. The XML declaration must be the first line in the document.
2. All XML elements must have a closing tag. <book> .. </book>.
3. XML tags are case sensitive. <Book> and <book> are not same.
4. XML elements must be properly nested.
5. XML attribute values must be quoted.

6. Characters like ‘<‘ which are already used for a specific purpose, if used as part of the data must be replaced with entity references.

7. White spaces are preserved in XML unlike HTML where white spaces are ignored.

   Example: XML: Hello World!   **HTML:** Hello World!

XML Character Entities

XML has predefined character entities such as:

- **&** (Ampersand) − **&amp;**
- ‘ (Single quote) − **&apos;**
- **“** (Double quote) − **&quot;**
- > (Greater than) − **&gt;**
- < (Less than) − **&lt;**

Whenever, any of these symbols need to be included in the XML document data, their corresponding character entities are used.

For Example, The < and > symbols are used to define a tag, such as, <book>. Therefore, if we want to include a statement like x > y in out data we should replace > with **&gt – x &gt y.**

XML Elements

Syntax:

<element-name attribute1 attribute2> ....content </element-name>

Example: <book> The Alchemist </book>

Attribute Syntax:

name = “value”

Example: <book category = “fiction”>

Empty Element Syntax:

<name attribute1 attribute2.../>

Example: </br>

XML Elements Rules

- An element name can contain any alphanumeric characters. The only punctuation mark allowed in names are the hyphen (-), under-score (_) and period (.).
- Names are case sensitive. For example, Address, address, and ADDRESS are different names.
- Start and end tags of an element must be identical.
- An element, which is a container, can contain text or other elements. Such as,

Element with text:

```
<book> Harry Potter </book>
```

Element with other element:

```
<book>
        <title> Harry Potter </title>
</book>
```

Element Attributes

Attribute Types:

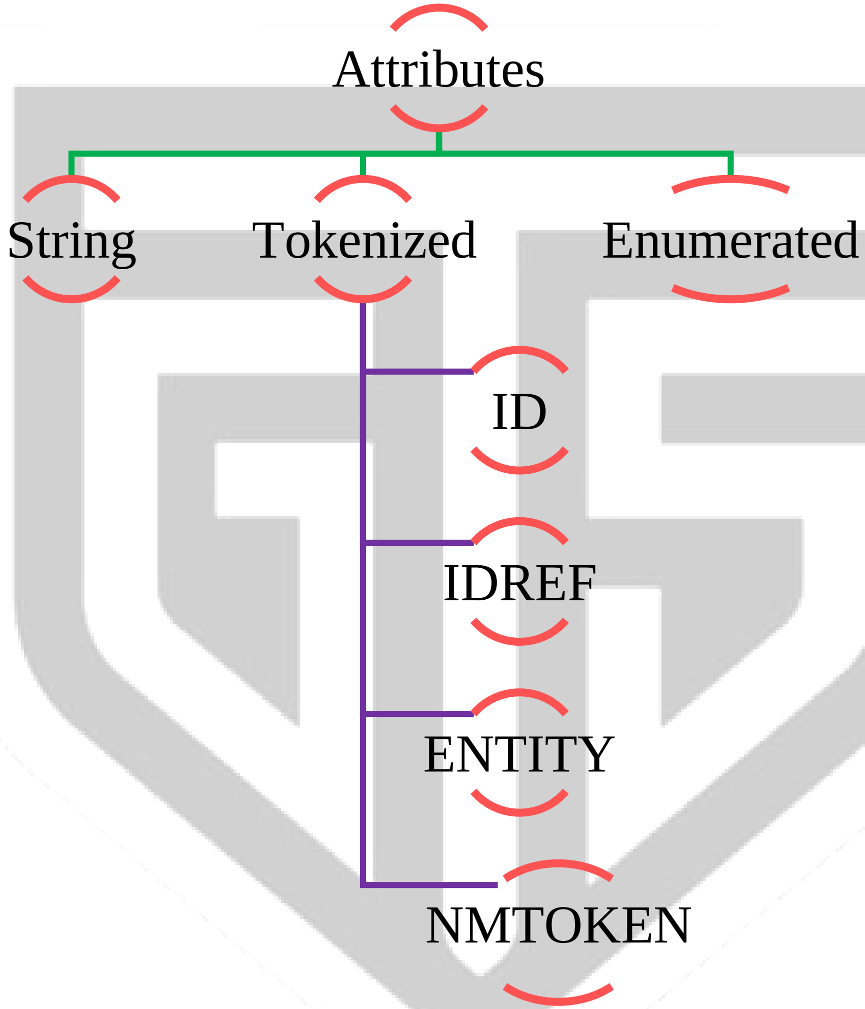
- StringType
- TokenizedType
- EnumeratedType

Rules:

1. The attribute value has to be in double (“) or single (‘) quotes.

2. Every attribute in an element must be unique.
3. Attribute can be of different types like String type (takes literal string as value), tokenized type (example, ID used to specify the element as unique) and enumerated type (predefined list of values in the declaration).

GROW TECH SKILLS



Attributes

String

Tokenized

Enumerated

ID

IDREF

ENTITY

NMTOKEN

XML comments

Syntax:

```
<!-- Comment -- >
```

For Example: <!—This is my first xml document -- >

Rules:

1. Comments cannot appear before the xml declaration.

2. Comments cannot break an element. For example,

```
<!—
    <book>
-->
```

3. This would result in the <book> element to have no start tag since it has become a part of the comment.

3. You cannot place a comment inside the xml element declaration. For example,

```
<book> <!—this is a book element -->
    Let us C
</book>
```

4. Finally, comments cannot be nested.

CDATA section

- CDATA stands for Character Data.
- CDATA is a Special element, the entire content of which is interpreted as character data, even if it appears to be markup.
- Begins with <![CDATA[and ends with]]> .
- Example:

```
<?xml version="1.0"?>
<text>
<![CDATA[ "x < y" & that "y < z" that means x < z ]]>
</text>
```

Here, ("x < y" & that "y < z" that means x < z) will be considered as data as is, without parsing.

Internal Entities

Internal Entities:

Internal entities allow us to define a large piece of data as an entity. In the actual xml document we can then use the entity name instead of the original text.

```
<!ENTITY name = “content”>
```

Example:

```
<!ENTITY amprights "All rights reserved. No part of this book, may be reproduced or passed on in any form, by any means (electronic, photocopying, or otherwise) without the prior consent of the publishers.">
```

Every time &amprights is used in the document it will be replaced by the corresponding text in the output.

External Entities

External Entities:

Also called **Binary entities,** external entities are mappings to an external file. Binary entities can define many types of data, such as, image, audio and video, the XML processor must be told how to interpret a given binary entity.

For example, consider this:

```
<!ENTITY entityName SYSTEM “filename” NDATA JPEG>
```

Here **SYSTEM** is a keyword that indicates that what follows is a filename. Following the filename is the **NDATA** keyword which indicates that this is a binary entity. And **JPEG** is the file type.

Know more

XML Fundamentals

Explains the fundamentals of XML.

This Video explain about fundamentals of XML

Topic	URL
XML Tutorial	https://www.w3schools.com/xml/

XML Fundamentals	https://www.youtube.com/watch?v=6Ob2h9_SfK8

Limitations of a Data Type Definition (DTD)

- Requires learning a **whole new syntax.**

- **Only one Data Type Definition** per xml document is permitted.

- An **internal** Data Type Definition overrides any **external** Data Type Definition.

- Defines only the **basic data types.**

- Not object oriented hence no support for **inheritance.**

- The **Document Object Model** used to read, edit and parse xml documents **cannot be used** for Data Type Definition.

Self Assessment Questions

1. DTD stands for ______________________.

Answer: Data Type Definition.

2. A DTD allows us to

   a) validate the contents of an XML document by describing the rules that need to be applied to the data in it.

   b) Define new data types

Answer: a)

3. Write the syntax for defining a PUBLIC DTD.**Answer: <!DOCTYPE root element PUBLIC**

   “URL”>

4. A well formed XML document:

   a) Follows the specified DTD rules.

   b) Is syntactically correct.

Answer: b)

5. Only one Data Type Definition per xml document is permitted. State true or false.

Answer: True.

6. An external DTD overrides internal DTD. State true or false.

Answer: False.

XML Schema

An alternative to DTD but with XML Syntax.

XML Schema Definition (XSD)

XML Schema Definition (XSD) is currently the standard schema language. An xml schema contains the top level schema element, it should include the namespace:

http://www.w3.org/2001/XMLSchema-instance

The **purpose** of an **XML Schema** is to define the legal building blocks of an **XML** document: the elements and attributes that can appear in a document, the number of (and order of) child elements and the data types for elements and attributes.

Here is an Example:

```
<?xml version=”1.0”?>
    <hello
    xmlns:xsi=”http://www.w3.org/2001/XMLSchema-instance”
    xsi:noNamespaceSchemaLocation=hello.xsd”>

        Hello World!

    </hello>
```

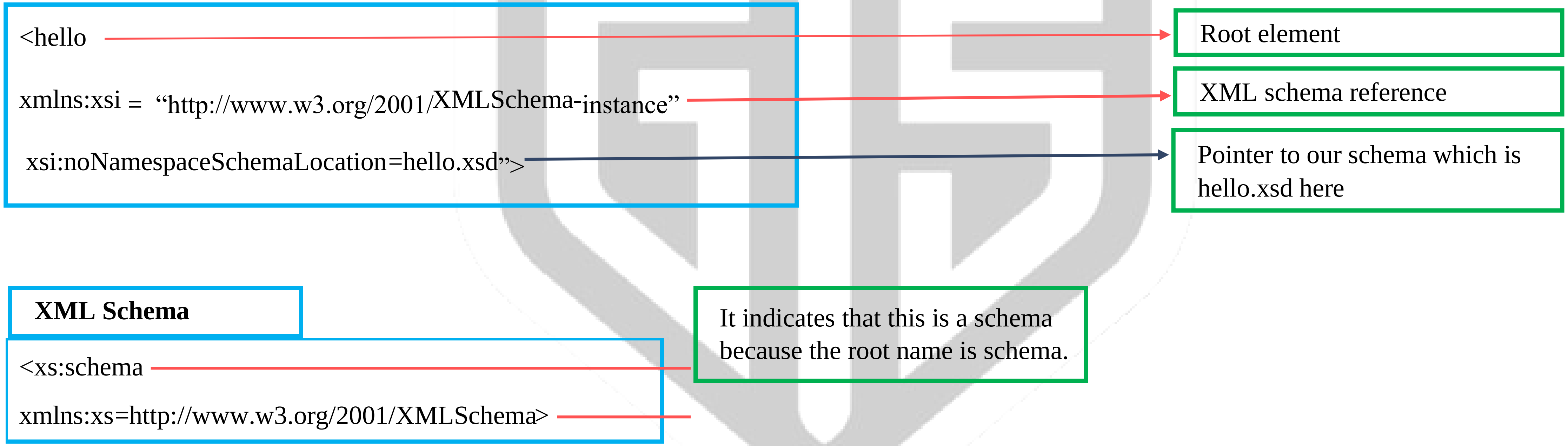
The code for hello.xsd will be:

```
<?xml version=”1.0”?>
```

```
<xsd:schema xmlns:xsd=http://www.w3.org/2001/XMLSchema>
<xsd:element name = “hello” type = “xsd:string”/>
</xsd:schema>
```

Let us understand the example

XML document



```
<hello
xmlns:xsi = “http://www.w3.org/2001/XMLSchema-instance”
xsi:noNamespaceSchemaLocation=hello.xsd”>
```

Root element

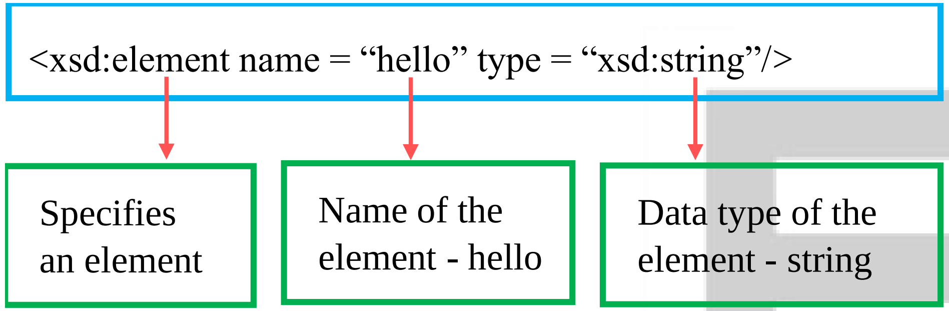
XML schema reference

Pointer to our schema which is hello.xsd here

XML Schema

```
<xs:schema
xmlns:xs=http://www.w3.org/2001/XMLSchema>
```

It indicates that this is a schema because the root name is schema.



```
<xsd:element name = “hello” type = “xsd:string”/>
```

Specifies an element

Name of the element - hello

Data type of the element - string

xs is the standard namespace and the schema definition for this namespace is present at the URI http://www.w3.org/2001/XMLSchema

Know more

XML Schema

Explains the XML schema and its advantages over DTDs.

This Video link explain the advantages of XML over DTDs

Topic	URL

XML Scheme advantages	http://www.billioncodes.com/en/ajwt/xml/933-cleary-explain-the-advantages-ofxml-schemas-over-dtds
XML advantages over DTDs	https://www.youtube.com/watch?v=vIW1XcsakV0

Self Assessment Questions

1. XML schema uses the same syntax as XML. State true or false.

Answer: True.

2. XML schema defines: (Select multiple)

   a) The elements and attributes that can appear in the document.

   b) Data types for elements and attributes.

c) The root element.

d) The order and number of child elements.

Answer: a), b), d).

3. Write the namespace an XML schema must include.

Answer: http://www.w3.org/2001/XMLSchema-instance

What is a Namespace?

XML Namespaces

XML namespaces provide us more information about elements in the form of qualifiers, thus removing any ambiguities.

For example:

```
<book_issue>
    <book_info>
        <number> 1000 </number>
        <title> Practical Web Design for Absolute Beginners </title>
        <author> Adrian W. West </author>
        <publisher> Apress </publisher>
    </book_info>
    <member_info>
        <number> 2222 </number>
        <title> Mr. </title>
        <author> Atul Bajaj</author>
```

```
                <issue_date> 12-dec-2017 </issue_date>
            </member_info>
        </book_issue>
```

XML Namespace Prefix

The prefix that we add to an element to make it unambiguous is called as a namespace prefix.

For example:

```
<book_issue xmlns:book=“urn:bookDetails” xmlns:member=“urn:memberDetails”>
    <book:book_info>
        <book:number> 1000 </number>
        <book:title> Practical Web Design for Absolute Beginners </title>
        <book;author> Adrian W. West </author>
        <book:publisher> Apress </publisher>
    </book:book_info>
    <member:member_info>
        <member:number> 2222 </number>
        <member:title> Mr. </title>
        <member:author> Atul Bajaj</author>
        <member:issue_date> 12-dec-2017 </issue_date>
    </member:member_info>
</member:book_issue>
```

XML Namespace URI

A **namespace prefix** needs to be tied to a **Uniform Resource Identifier (URI).** The URI has a name. We are free to use a URI name of our choice.

For example:

```
<book_issue xmlns:book=“urn:bookDetails” xmlns:member=“urn:memberDetails”>
```

- BOOK_NAME is the root element of our XML document.

- To this root element, we have added two namespace prefixes: book and member. In effect, we inform the XML parser that it may see elements and attributes prefixed by book or member.

- The declarations **urn:bookDetails** and **urn:memberDetails** simply indicates a longer name for the namespace prefix, called as namespace URI. This has nothing to do with the actual elements in the XML document.

XML Default Namespace

Example: Default namespace for XHTML is,

```
<html xmlns = http://www.w3.org/1999/xhtml>
```

For example:

```
<BOOK_ISSUE xmlns = “http://www.dummysite.com/myURL–book”
xmlns:member = “http://www.dummysite.com/myURL–member”>
```

Here,

”http://www.dummysite.com/myURL–book” and http://www.dummysite.com/myURL–member are dummy URLs. We can specify any URL.

Know more

XML Namespaces

🗎 Explains how namespaces are used in XML to resolve the ambiguity of element names.

▶ This Video Links explain how namespaces are used to resolve the element names

Topic	URL

XML namespaces	https://www.w3schools.com/xml/xml_namespaces.asp
XML namespaces to resolve conflict	https://www.youtube.com/watch?v=pzYf_sTtmOQ

Self Assessment Questions

1. XML namespaces are used for providing uniquely named elements and attributes in an XML document.
   State true or false.

Answer: True.

2. A _______________ is used to resolve ambiguity between elements of the same name.a)

   Namespace prefix

   b) Namespace tag

Answer: a)

3. A namespace prefix needs to be tied to a Uniform Resource Identifier. State true or false.

Answer: True.

4. A namespace prefix is:

   a) System defined

   b) User defined

Answer: b)

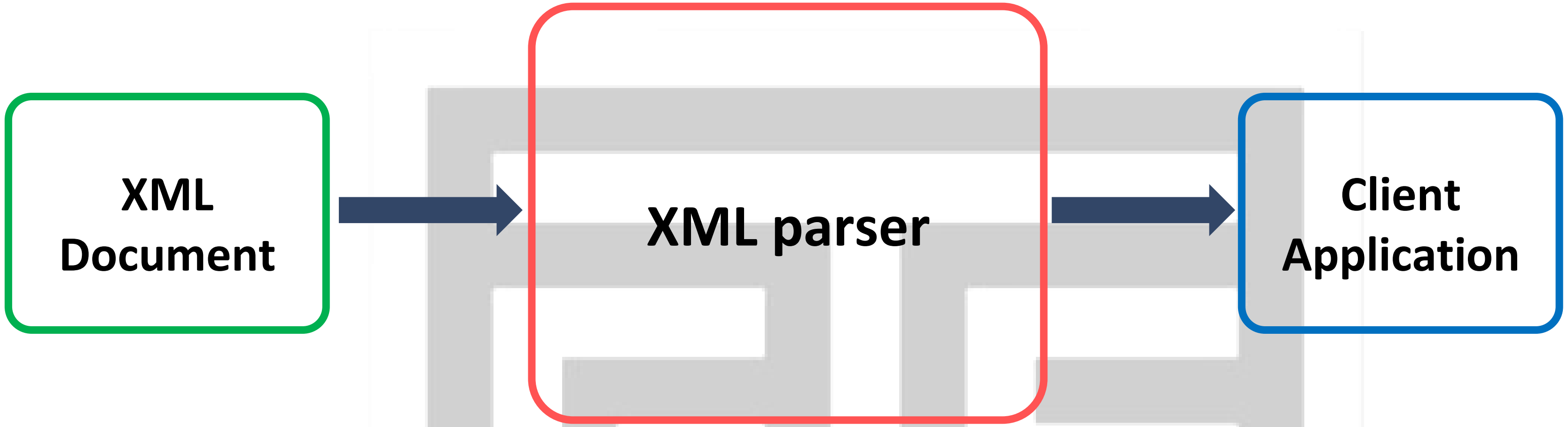
5. Write the default namespace for XHTML.

Answer: <html xmlns = http://www.w3.org/1999/xhtml>.

What is a Parser?

XML Parser

A software that reads, interprets and additionally may also validate an xml document and acts as an interface between the xml document and the client program is called an **‘xml parser’.**



XML Document

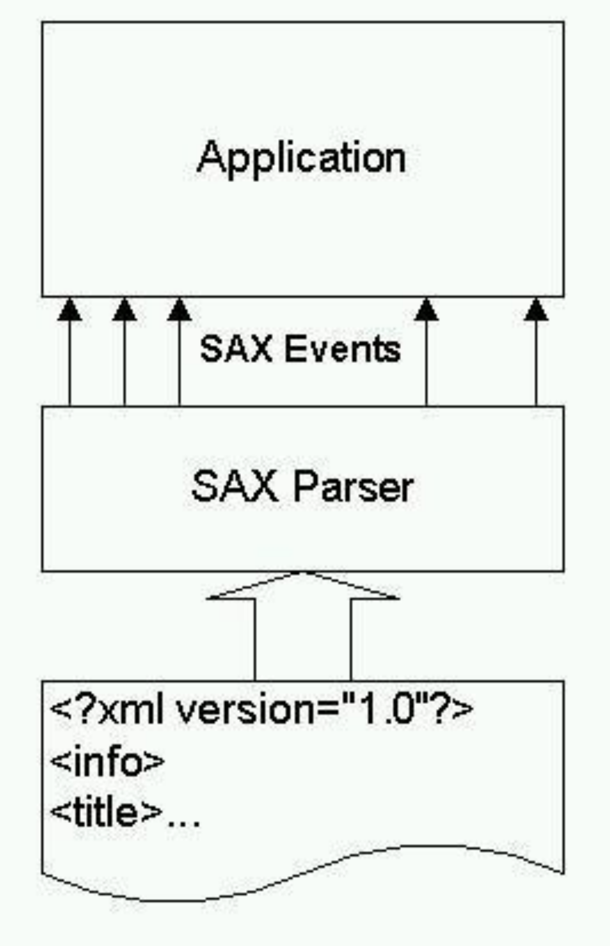
XML parser

Client Application

The two main xml parsers are:

- **SAX** (Simple API for XML) – Event based parser.
- **DOM** (Document Object Model) – Object based parser.

Simple API for XML Parser



What does Simple API for XML do?

SAX or Simple API for XML is an event based parser.

1. Reads an XML document from top to bottom, recognizing the tokens that make up a well-formed XML document.

2. Tokens are processed in the same order that they appear in the document.

3. Reports the application program about the nature of tokens that the parser has encountered as they occur.

4. The application program provides an "event" handler that must be registered with the parser.

5. As the tokens are identified, callback methods in the handler are invoked with the relevant information.

SAX Parser Advantages and Disadvantages

Advantages:

- Analysis begins immediately.
- The processed data is not stored in the memory.
- It is easier to process large XML documents.
- The application need not process the entire document if not required.

Disadvantages:

- Backward navigation is not possible.
- Modifying the XML document is not possible.
- Using SAX parser we cannot build the XML document.

Know more

XML Parsers

Explains event based and object based XML parser.

This Video explain event and Object based XML parser

Topic	URL
XML Parser	https://www.w3schools.com/xml/xml_parser.asp
XML parser Object based and event based	https://www.youtube.com/watch?v=aLxrk1vK0ZU

Self Assessment Questions

1. An XML parser is:

a) A software that validates the XML document.

b) A software that reads, interprets and may additionally validate an XML document.

Answer: b)

2. An XML parser acts as an interface between the xml document and the client program. State true or false.

Answer: True.

3. SAX stands for ____________________________.

Answer: Simple API for XML.

4. SAX is an:

   a) Event based parser.

   b) Object based parser.

Answer: a)

5. SAX parser processes the XML document in a sequential order from top to bottom. State true or false.

Answer: True.

6. Select the incorrect option:

   a) Using SAX we cannot build or modify the XML document.

b) Using SAX backward navigation is possible.

c) SAX is suitable for large XML documents.

d) SAX does not store the processed data in memory.

Answer: a)

XML DOM

a standard for accessing and manipulating XML documents.

XML DOM

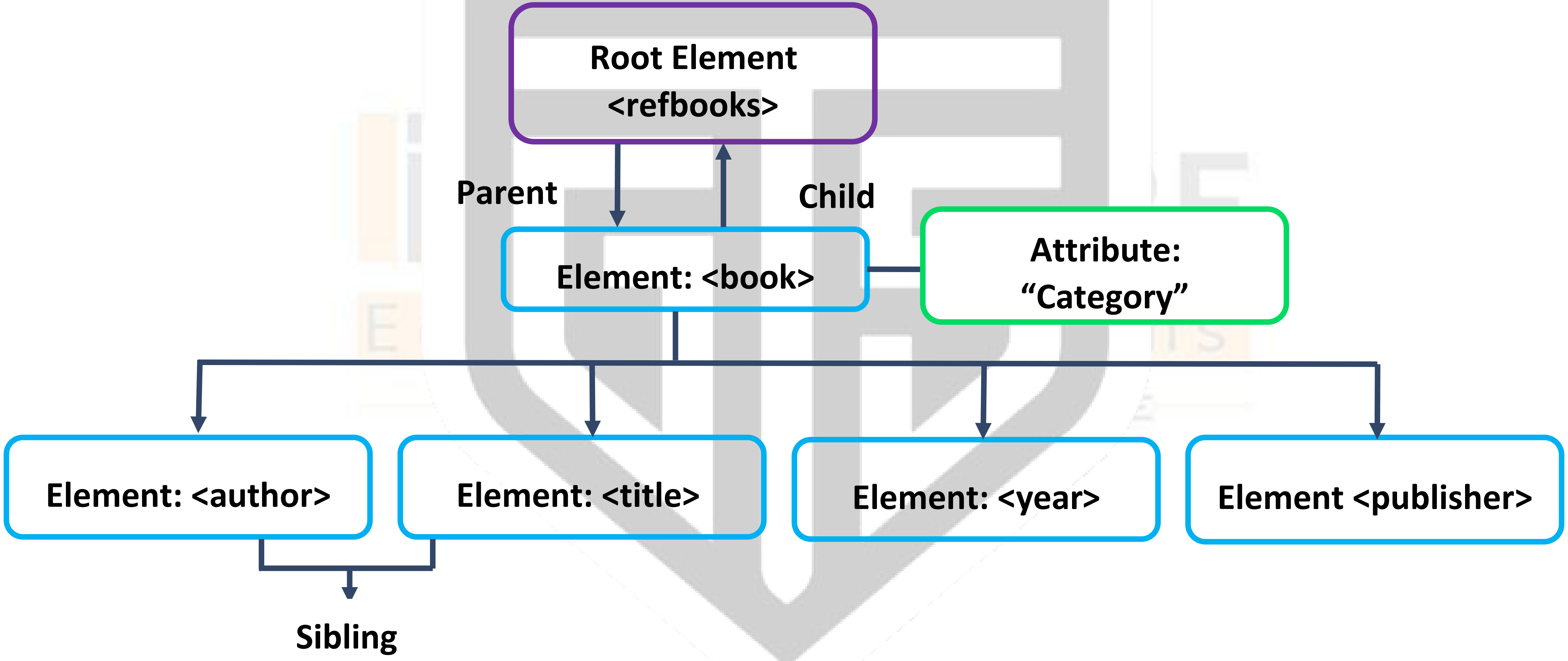
- The XML DOM makes a tree-structure view for an XML document.
- We can access all elements through the DOM tree.
- We can modify or delete their content and also create new elements. The elements, their content (text and attributes) are all known as nodes.

Introduction to XML P1-XMLB

Consider this code:

```
<refbooks>
    <book category=”web”>
        <author> Adrian W. West </author>
        <year>2016</year>
        <title> Practical Web Design for Absolute Beginners </title>
        <publisher> Apress </publisher>
    </book>
</refbooks>
```

XML DOM Tree


Root Element
<refbooks>
Parent
Child
Element: <book>
Attribute:
“Category”
Element: <author>
Element: <title>
Element: <year>
Element <publisher>
Sibling

DOM Parser Advantages and Disadvantages

Advantages:

- Simple to modify and extract data.
- Easy to use.
- Random access of elements is possible.

Disadvantages:

- Stores the entire document in memory.
- Construction of the DOM tree is time consuming.
- It can be impractical for large documents.

Introduction to XML P1-XMLB

Advantages of XML

1. It supports Unicode.
2. Common data structures can be represented.
3. Self-documenting format.
4. Document storage and processing.
5. It is based on international standards.
6. It can be updated incrementally.
7. It allows validation.
8. Hierarchical structure of documents.
9. Platform-independent.
10. Forward and backward compatibility.

Introduction to XML

Know more

XML Document Object Model

Explains the XML Document Object Model.

This Video explains about XML Document Object Model

Topic	URL
XML DOM Tutorial	https://www.w3schools.com/xml/dom_intro.asp
XML DOM using in data	https://www.youtube.com/watch?v=Nb4cO5BZ9Fc

Self Assessment Questions

1. DOM stands for ____________________.

Answer: Document Object Model.

2. The XML DOM creates the _________________ from the XML document.

   a) A DOM tree

   b) A DOM event list

Answer: a)

3. A DOM tree allows:
   a) to traverse the XML document only.
   b) Modify, add or delete elements in the XML document.

Answer: b)

4. DOM parser is ideal for large documents. State true or false.

Answer: False.

Introduction to XML

5. DOM parser stores the entire document in memory. State true or false.

Answer: True.

Introduction to XML P1-XMLB

Assignment

Introduction to XML

Assignment

Description about assignment.

I. Create an XML document for information about hotels in Bangalore. The document should contain the given data:

1. **ID –** 1, 2, 3, 4.

2. **Name –** Lemon Tree Premier, Royal Orchid Central, The Chevron Hotel, Shelton Grand Hotel.

3. **Stars –** 4, 4, 3, 3

4. **Facilities –** Internet, Gym and Pickup. Internet and Gym.
   Internet and Parking.
   Internet and Parking.

Introduction to XML

5. Address - 2/1, Saint Johns Road, Ulsoor, Bengaluru-42

   47/1, Dickenson Road, Manipal Centre, Bengaluru -42

   147,Infantry Road, Bengaluru – 1

   73, Entrance Church Street, M.G. Road, Bengaluru – 1

6. Distance from center – 4.1 Km, 0.8 Km, 3.3 Km, 2.9 Km.

7. Available – True, False, True, False.

II. Write an external DTD or XML Schema considering the given facts:

1. Stars range from 3 to 5.
2. Facilities can have at least one or more of the followings: Internet, Gym, Parking, Pick-up.

3. Distance can be empty or have one value.
4. Available can be either true or false.