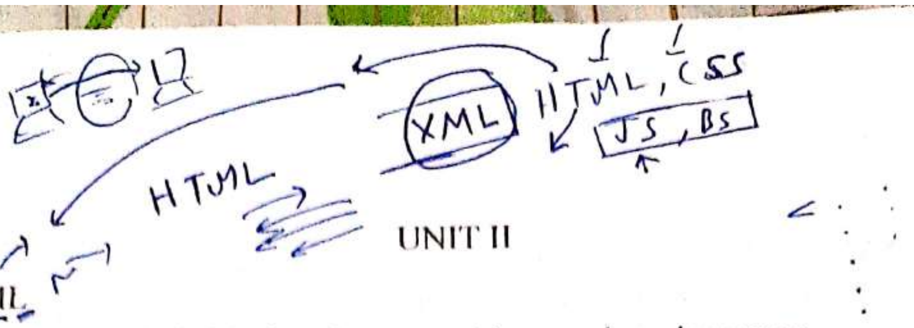
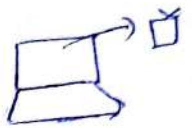


UNIT II

What is XML

- Xml (eXtensible Markup Language) is a mark up language.
- XML is designed to store and transport data.
- Xml was released in late 90's. it was created to provide an easy to use and store self describing data.
- XML became a W3C Recommendation on February 10, 1998.
- XML is not a replacement for HTML.
- XML is designed to be self-descriptive.
- XML is designed to carry data, not to display data.
- XML tags are not predefined. You must define your own tags.
- XML is platform independent and language independent.

XML



AD

XML

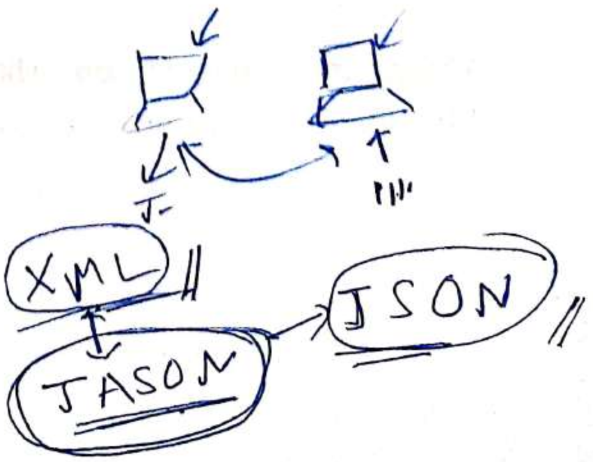
What is mark-up language

A **mark up language** is a modern system for highlight or underline a document. Students often underline or highlight a passage to revise easily, same in the sense of modern mark up language highlighting or underlining is replaced by tags.

Why xml

Platform Independent and Language Independent: The main benefit of xml is that you can use it to take data from a program like Microsoft SQL, convert it into XML then share that XML with other programs and platforms. You can communicate between two platforms which are generally very difficult.
The main thing which makes XML truly powerful is its international acceptance. Many corporation use XML interfaces for databases, programming, office application mobile phones and more. It is due to its platform independent feature.

Features and Advantages of XML



Education4Fun.com

XML is widely used in the era of web development. It is also used to simplify data storage and data sharing.

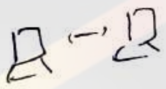
The main features or advantages of XML are given below.

1) XML separates data from HTML

If you need to display dynamic data in your HTML document, it will take a lot of work to edit the HTML each time the data changes.

With XML, data can be stored in separate XML files. This way you can focus on using HTML/CSS for display and layout, and be sure that changes in the underlying data will not require any changes to the HTML.
With a few lines of JavaScript code, you can read an external XML file and update the data content of your web page.

2) XML simplifies data sharing

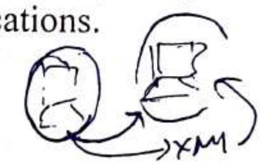


In the real world, computer systems and databases contain data in incompatible formats. XML data is stored in plain text format. This provides a software- and hardware-independent way of storing data.
This makes it much easier to create data that can be shared by different applications.

3) XML simplifies data transport

One of the most time-consuming challenges for developers is to exchange data between incompatible systems over the Internet.
Exchanging data as XML greatly reduces this complexity, since the data can be read by different incompatible applications.

4) *XML simplifies Platform change*



Upgrading to new systems (hardware or software platforms), is always time consuming. Large amounts of data must be converted and

Education4Fun.com

incompatible data is often lost.
XML data is stored in text format. This makes it easier to expand or upgrade to new operating systems, new applications, or new browsers, without losing data.

5) XML increases data availability//
Different applications can access your data, not only in HTML pages, but also from XML data sources.
With XML, your data can be available to all kinds of "reading machines" (Handheld computers, voice machines, news feeds, etc), and make it more available for blind people, or people with other disabilities.
6) XML can be used to create new internet languages A lot of new Internet languages are created with XML. Here are some examples:

- **XHTML**
- **WSDL** for describing available web services
- **WAP** and **WML** as markup languages for handheld devices
- **RSS** languages for news feeds
- **RDF** and **OWL** for describing resources and ontology
- **SMIL** for describing multimedia for the web.

XML Example

XML documents create a hierarchical structure looks like a tree so it is known as XML Tree that starts at "the root" and branches to "the leaves".
Syntax:

```
<root>
 <child>
  <subchild>     </subchild>
 </child>
</root>
```

Education4Fun.com

Example of XML Document

XML documents uses a self-describing and simple syntax:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<note>
 <to>Tove</to>
 <from>Jani</from>
 <heading>Reminder</heading>
 <body>Don't forget me this weekend!</body>
```

The next line describes the root element of the document (like saying: "this document is a note"):

```
<note>
```

(The next 4 lines describe 4 child elements of the root (to, from, heading, and body).

```
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend!</body>
```

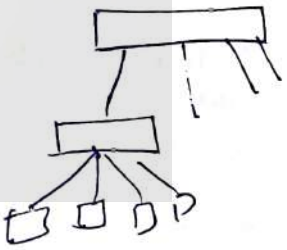
And finally the last line defines the end of the root element.

```
</note> )
```

XML: Books.xml

Attri

```
<bookstore>
<book category="COOKING">
<title lang="en">Everyday Italian</title>
<author>Giada De Laurentiis</author>
<year>2005</year>
```



Education4Fun.com

```
<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">Learning XML</title>
<author>Erik T. Ray</author>
<year>2003</year>
<price>39.95</price>
</book>
</bookstore>
```

XML: Emails.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<emails>
<email>
 <to>Vimal</to>
 <from>Sonoo</from>
 <heading>Hello</heading>
 <body>Hello brother, how are you!</body>
</email>
<email>
 <to>Peter</to>
 <from>Jack</from>
 <heading>Birth day wish</heading>
 <body>Happy birth day Tom!</body>
```

Education4Fun.com

```
</email>
<email>
 <to>James</to>
 <from>Jaclin</from>
 <heading>Morning walk</heading>
 <body>Please start morning walk to stay fit!</body>
 </email>
<email>
 <to>Kartik</to>
 <from>Kumar</from>
 <heading>Health Tips</heading>
 <body>Smoking is injurious to health!</body>
</email>
</emails>
```

No.	Technology	Meaning	Description
1)	XHTML	Extensible html	It is a clearer and stricter version of XML. It belongs to the family of XML markup languages. It was developed to make html more extensible and increase inter-operability with other data.
2)	XML DOM	XML document object model	It is a standard document model that is used to access and manipulate XML. It defines the XML file in tree structure.

Education4Fun.com

3)	XSL it contain three parts: i) XSLT (xsl transform) ii) XSL iii)XPath	Extensible style sheet language	i) It transforms XML into other formats, like html. ii) It is used for formatting XML to screen, paper etc. iii) It is a language to navigate XML documents.
4)	XQuery	XML query language	It is a XML based language which is used to query XML based data.
5)	DTD	Document type definition	It is an standard which is used to define the legal elements in an XML document.
6)	XSD	XML schema definition	It is an XML based alternative to dtd. It is used to describe the structure of an XML document..
7)	XLink	XML linking	xlink stands for XML linking language. This is a

Education4Fun.com

		language	language for creating hyperlinks (external and internal links) in XML documents.
8)	XPointer	XML pointer language	It is a system for addressing components of XML based internet media. It allows the xlink hyperlinks to point to more specific parts in the XML document.
9)	SOAP	Simple object access protocol	It is an acronym stands simple object access protocol. It is XML based protocol to let applications exchange information over http. in simple words you can say that it is protocol used for accessing web services.
10)	WSDL	web services description languages	It is an XML based language to describe web services. It also describes the functionality offered by a web service.
11)	RDF	Resource description framework	RDF is an XML based language to describe web resources. It is a standard model for data interchange on the web. It is used to describe the title, author, content and copyright information of a web page.

Education4Fun.com

12)	SVG	Scalable vector graphics	It is an XML based vector image format for two- dimensional images. It defines graphics in XML format. It also supports animation.
13)	RSS	Really simple syndicatio n	RSS is a XML-based format to handle web content syndication. It is used for fast browsing for news and updates. It is generally used for news like sites.

XML Attributes ②

<h1 ↔ = " " >

XML elements can have attributes. By the use of attributes we can add the information about the element.
XML attributes enhance the properties of the elements.

```
<book publisher="Tata McGraw Hill"></book>
```

Here, book is the element and publisher is the attribute.

Metadata should be stored as attribute and data should be stored as element.

```
<book>
<book category="computer">
<author> A & B </author>
</book>
```

MD D ← D

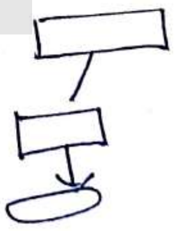
Difference between attribute and sub-element

1st way:

```
<book publisher="Tata McGraw
Hill"> </book>
```

2nd way:

```
<book>
<publisher> Tata McGraw Hill </publisher>
</book>
```



Education4Fun.com

<!- - / - . - - >

[illegible] "O"

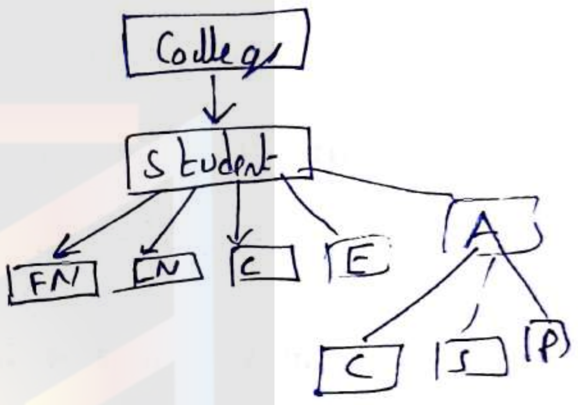
syntax An XML comment should be written as: <!-- Write your comment-->

1. Don't use a comment before an XML declaration.
2. You can use a comment anywhere in XML document except within attribute value.
3. Don't nest a comment inside the other comment.

XML Tree Structure

An XML document has a self descriptive structure. It forms a tree structure which is referred as an XML tree. The tree structure makes easy to describe an XML document.

```
<?xml version="1.0"?>
<college>
 <student>
   <firstname>Tamanna</firstname>
   <lastname>Bhatia</lastname>
   <contact>09990449935</contact>
   <email>tammanabhatia@abc.com</email>
   <address>
     <city>Ghaziabad</city>
     <state>Uttar Pradesh</state>
     <pin>201007</pin>
   </address>
 </student>
</college>
```



Education4Fun.com

→ XML DTD ③ |HTTP| ▭↙↗

→ XML Document type Definition / Declaration

→ Used to describe XML language precisely.

→ Used to define structure of a XML file

→ Contains list of legal element.

→ Used to perform validation.

Syntax:

<! DOCTYPE element DTD identifier
[declaration 1
declaration 2]>

CSS

Types of DTD

$$\begin{tikzcd} & \text{Types of DTD} \arrow[dl] \arrow[dr] & \\ \text{Internal} & & \text{External} \end{tikzcd}$$

.dtd

Internal	External
elementss are declared within the XML File	elements are declared outside XML file
<! DOCTYPE root-element [element-declaration]>	<! DOCTYPE root-element SYSTEM "file-name" >

Internal DTD example

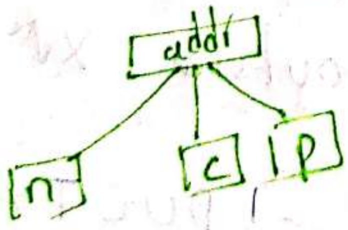
```
<? xml version = "1.0"
   encoding = " UTF-8" >
<!DOCTYPE Address[
  <!Element Address(
     name, company, phone) >
  <!Element name(#PC
               Data)
     :        :
 ]>
<Address >
   <name > ___ </name >
   <company > ___ </company
   <phone > — </phone.
</address>
```

S,XML

DTD

root

root



External DTD example

Add.dtd

```
<? xml version = "1.0"
   encoding = " UTF-8">
<!DOCTYPE Address[
 <!Element Address(
    name, company,
    phone) >
 <!Element name
    (#PC Data)
      :      :
  ]>
```

XML File

```
<!DOCTYPE
    Address
  SYSTEM
    " Add.dtd >
<Address >
   <name > ____ </name >
   <company > — </c~~
   <phone> — </pho~>
</Address >
```

Adv:
→ compact & easy to use
→ can be defined outside/inside XML document
→ Widely used

Disadv:-
→ They lack some felexibility
→ Not written using XML syntax.
→ No data typing (can't use limiting to string or integer).

④ D

→ XML NameSpace :- → Used to avoid element name conflict in XML documents.
→ It is a set of unique names
→ Identified by URI < URL / URN
(Unifrom resource name space) : table
→ Attribute name must start with "xmlns".

Syntax: <element xmlns : name="URI">

Education4Fun.com

Cofulict:- Generally conflict occurs when we try to mix XML documents from different application

exe:

Html table info

```
<table>
   <tr> Apple </tr>
   <tr> Banana </tr>
</table.
```

XML table info

```
<table>
   <name> cofee
         table </name>
   <width > 80
         </width >
   <length > 120
         </length >
</table>
```

here both data has table data hence when you combine their will be a conflict. To resolve it

```
<?xml version = "1.0" encoding = "UTF-8" ?>
<h : table  xmlns : h = " link " >
      <h : tr >
            <h : td > Apple </h : td >
            <h : td > Banana </h : td >
      </h : tr >
</h : table >
```

h:table

Education4Fun.com

```
< x : table   xmlns : x = " link " >
        < x : name > coffe table </name >
        < x . width >   80     </width >
      < x . length >  120    </length >
< x : table >
```

∴ Now there will be no conflict due to name space.

(DTD) → (.~~xml~~) (.xsd) DTD →→ ⟷

→ **XML Schemas:** Commonly known as XML Schema Definition (XSD). It is used to describe & validate the structure & content of XML Data (like DTD (Document type Definition)

→ It is a method of expressing constraints about XML documents.

→ It is like DTD but provides more control on XML structure.

→ Rule / writing is simple as we use xml model syntax only.

Education4Fun.com

→ XML Schema elements

→ Previously in DTD we are not able to define the data type (we use to give (PC Data) here in XML we can define data type.

< xs : element name = " x " type = " y " /> "xs : integer"

(< ↔ />)

ex:
For simple element :

< lastname > Ramu </lastname >

⇓

< xs : element name = "lastname"
type = " xs. string " / >

For complex element with attribute

< lastname lang = " E N " > Smith </last
name >

⇓

< xs : element name = " lastname "
type = " xs. string >

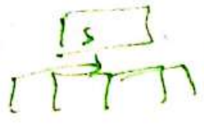
< xs : attribute name = " lang " type = " xs. string/>
</xs : element >

Education4Fun.com

ex:-

.xml

```
<student>
    <no> 72 </rno>
    <name> ramu </name>
    <avg> 90.2 </avg>
    <dob> 1-1-1990 </dob>
    <time> 07:00:00 </time>
CE <mobile no = "99999 99999"/>
    <distinction> Yes </distinction>
</student>
```



Data types
- xs: string.
- xs: decimal.
- xs: integer.
- xs: boolean.
- xs: date.
- xs: time.
- xs: double

.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xs: schema xmlns:xs="http://www.
w3.org/2001/XMLSchema">
    <xs: element name="student">
        <xs: complexType>
            <xs: sequence>
                <xs: element name="no"
                        type="xs:integer"/>
                <xs: element name="name"
                        type="xs : string"/>
```

Education4Fun.com

```
<xs:element name="avg" type="xs:double"/>
<xs:element name="dob" type="xs:date"/>
<xs:element name="time" type="xs:time"/>
<xs:element name="mobile">
    <xs:complexType>
        <xs:attribute name="no"
                  type="xs:int"/>

    </xs:complexType>
</xs:element>
<xs:element name="distinction" type="xs:
                                   boolean"/>

</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>
```

✓ DTD v/s XSD ⑥

✓ XML with CSS ⑦

X Path ⑧ ✓

XSLT ⑨

XML Processor/parser ⑩

Education4Fun.com

DTD vs XSD ⑥

No.	DTD	XSD
1)	DTD stands for **Document Type Definition.**	XSD stands for XML Schema Definition.
2)	DTDs are derived from **SGML** syntax.	XSDs are written in XML. XML UD
3)	DTD **doesn't support datatypes.** (#PC Data)	XSD **supports datatypes** for elements and attributes.
4)	DTD **doesn't support namespace.**	XSD **supports namespace.**
5)	DTD **doesn't define order** for child elements.	XSD **defines order** for child elements.
6)	DTD is **not extensible.**	XSD is **extensible.**
7)	DTD is **not simple to learn.**	XSD is **simple to learn** because you don't need to learn new language.
8)	DTD provides **less control** on XML structure.	XSD provides **more control** on XML structure.

Education4Fun.com

CDATA

CDATA: (Unparsed Character data): CDATA contains the text which is not parsed further in an XML document. Tags inside the CDATA text are not treated as markup and entities will not be expanded.

```
<?xml version="1.0"?>
<!DOCTYPE employee SYSTEM "employee.dtd">
<employee>
<![CDATA[
 <firstname>vimal</firstname>
 <lastname>jaiswal</lastname>
 <email>vimal@javatpoint.com</email>
]]>
</employee>
```

PCDATA

PCDATA: (Parsed Character Data): XML parsers are used to parse all the text in an XML document. PCDATA stands for Parsed Character data. PCDATA is the text that will be parsed by a parser. Tags inside the PCDATA will be treated as markup and entities will be expanded.

```
<?xml version="1.0"?>
<!DOCTYPE employee SYSTEM "employee.dtd">
<employee>
 <firstname>vimal</firstname>
 <lastname>jaiswal</lastname>
 <email>vimal@javatpoint.com</email>
</employee>
```

XML→ U D

XML CSS 7

Purpose of CSS in XML

CSS (Cascading Style Sheets) can be used to add style and display information to an XML document. It can format the whole XML

Education4Fun.com

document.

How to link XML file with CSS

To link XML files with CSS, you should use the following syntax:

```
<?xml-stylesheet type="text/css" href="cssemployee.css"?>
```
③ ① Ex

XML CSS Example

cssemployee.css

```
employee
{
background-color:
pink;
}
firstname,lastname
,email
{
font-
size:25px;
display:block
; color: blue;
margin-left:
50px;
}
```

employee.xml

```
<?xml
version="1.0"?>
```

Education4Fun.com

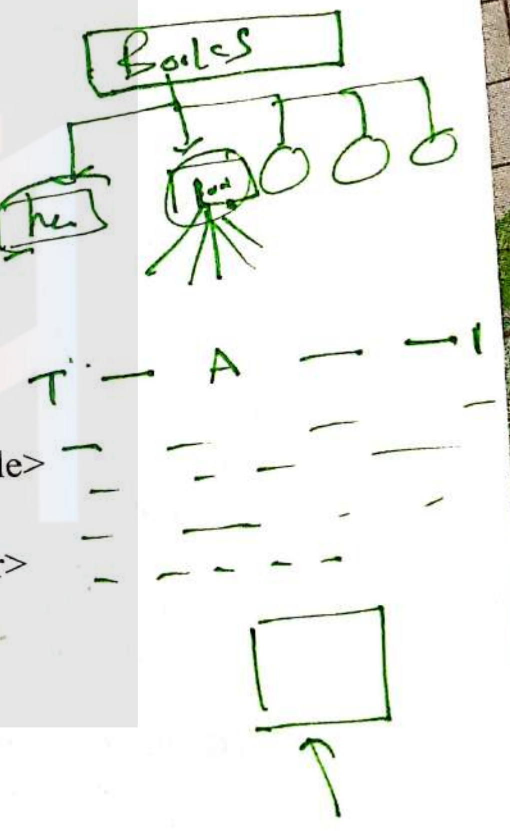
```
<?xml-stylesheet type="text/css" href="cssemployee.css"?>
<!DOCTYPE employee SYSTEM "employee.dtd">
<employee>
 <firstname>vimal</firstname>
 <lastname>jaiswal</lastname>
 <email>vimal@javatpoint.com</email>
</employee>
```

Example 1.

In this example, the XML file is created that contains the information about five books and displaying the XML file using CSS.

Books.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/css" href="Rule.css"?>
<books>
      <heading>Welcome To Library </heading>
      <book>
            <title>Title -: Web Programming</title>
            <author>Author -: Chrisbates</author>
            <publisher>Publisher -: Wiley</publisher>
            <edition>Edition -: 3</edition>
            <price> Price -: 300</price>
      </book>
      <book>
            <title>Title -: Internet world-wide-web</title>
            <author>Author -: Ditel</author>
            <publisher>Publisher -: Pearson</publisher>
            <edition>Edition -: 3</edition>
            <price>Price -: 400</price>
      </book>
      <book>
            <title>Title -: Computer Networks</title>
            <author>Author -: Foruouzan</author>
            <publisher>Publisher -: Mc Graw Hill</publisher>
            <edition>Edition -: 5</edition>
            <price>Price -: 700</price>
```



Education4Fun.com

```
        </book>
        <book>
            <title>Title -: DBMS Concepts</title>
            <author>Author -: Navath</author>
            <publisher>Publisher -: Oxford</publisher>
            <edition>Edition -: 5</edition>
            <price>Price -: 600</price>
        </book>
        <book>
            <title>Title -: Linux Programming</title>
            <author>Author -: Subhitab Das</author>
            <publisher>Publisher -: Oxford</publisher>
            <edition>Edition -: 8</edition>
            <price>Price -: 300</price>
        </book>
    </books>
```

Rule.css

```
books {
        color: white;
        background-color
        : gray; width:
        100%;
}
heading {
        color:
        green; font-
        size : 40px;
        background-color : powderblue;
}
heading, title, author, publisher,
        edition, price { display :
        block;
```

Education4Fun.com

```
   }
   title {
font-size : 25px; font-
weight : bold;
   }
```

in this example, the XML file is created that contains the information about various sections in Geeks for Geeks and the topics they contains and after that displaying the XML file using CSS .

Section.xml

```
        <?xml version="1.0"
        encoding="UTF-8"?>
   <?xml-stylesheet type="text/css" href="Geeks.css"?>
   <Geeks_for_Geeks>
        <title>Hello Everyone! Welcome to Library</title>
        <geeks_section>
              <name>Algo</name>
              <topic1>Greedy Algo</topic1>
              <topic2>Randomised Algo</topic2>
              <topic3>Searching Algo</topic3>
              <topic4>Sorting Algo</topic4>
        </geeks_section>
        <geeks_section>
              <name>Data Structures</name>
              <topic1>Array</topic1>
              <topic2>Stack</topic2>
              <topic3>Queue</topic3>
              <topic4>Linked List</topic4>
        </geeks_section>
        <geeks_section>
              <name>Web Technology</name>
              <topic1>HTML</topic1>
              <topic2>CSS</topic2>
              <topic3>Java Script</topic3>
              <topic4>Php</topic4>
        </geeks_section>
```

Education4Fun.com

```
    <geeks_section>
        <name>Languages</name>
        <topic1>C/C++</topic1>
        <topic2>Java</topic2>
        <topic3>Python</topic3>
        <topic4>Ruby</topic4>
    </geeks_section>
    <geeks_section>
        <name>DBMS</name>
        <topic1>Basics</topic1>
        <topic2>ER Diagram</topic2>
        <topic3>Normalization</topic3>
        <topic4>Transaction Concepts</topic4>
    </geeks_section>
</Geeks_for_Geeks>
```

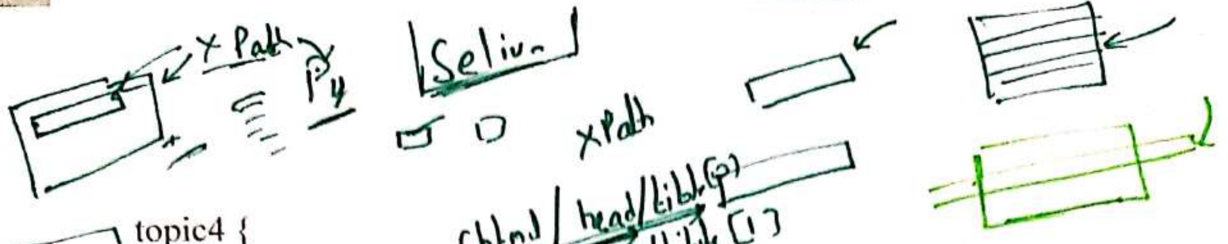
Geeks.css

```
Geeks_for_G
eeks            {
                font-
                size:80%;
                margin:0.5
                em;
                font-family:
                Verdana;
                display:block;
geeks_secti     }
on {
                display:block;
                border: 1px solid
                silver;
                margin:0.5em;
                padding:0.5em;
title {         background-color:whitesmoke;
                }
    display:block;
```

Education4Fun.com

```
          font-
          weight:bolder;
          text-
          align:center;
          font-
          size:30px;
          background-color:
          green; color:
          white;

          }
 name, topic1, topic2, topic3, topic4
 {
 display:
 block; text-
 align: center;
 }
 name {
        color: green;
        text-decoration:
        underline ; font-
        weight: bolder;
        font-size:20px;
        }
topic1 {
       color: green
       }
topic2 {
       color: brown
       }
topic3 {
       color: blue
       }
```

Education4Fun.com



F 11
F 12

topic4 {
color: orange }

WHAT IS XPATH

XPath is an official recommendation of the World Wide Web Consortium (W3C). It defines a language to find information in an XML file. It is used to traverse elements and attributes of an XML document. XPath provides various types of expressions which can be used to enquire relevant information from the XML document.

W3C

- **Structure Definitions** – XPath defines the parts of an XML document like element, attribute, text, namespace, processing-instruction, comment, and document nodes
- **Path Expressions** – XPath provides powerful path expressions select nodes or list of nodes in XML documents.
- **Standard Functions** – XPath provides a rich library of standard functions for manipulation of string values, numeric values, date and time comparison, node and QName manipulation, sequence manipulation, Boolean values etc.
- **Major part of XSLT** – XPath is one of the major elements in XSLT standard and is must have knowledge in order to work with XSLT documents.
- **W3C recommendation** – XPath is an official recommendation of World Wide Web Consortium (W3C).

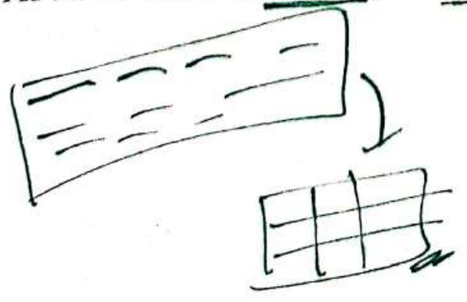
XP
XL
XML
JASON
XP
XML
XSL
HTML

One should keep the following points in mind, while working with XPath –

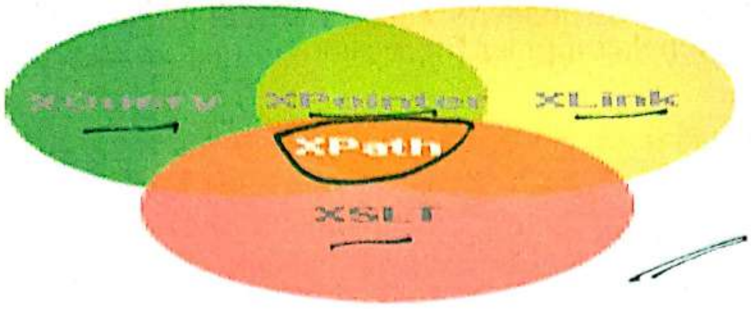
- XPath is core component of XSLT standard.
- XSLT cannot work without XPath.
- XPath is basis of XQuery and XPointer.

XSL

XML
Data



Education4Fun.com



- Path stands for XML Path Language
- XPath uses "path like" syntax to identify and navigate nodes in an XML document
- XPath contains over 200 built-in functions
- XPath is a major element in the XSLT standard
- XPath is a W3C

recommendation Important) same in intro part theory

features of XPath

X (◦ **XPath defines structure:** XPath is used to define the parts of an XML document i.e. element, attributes, text, namespace, processing-instruction, comment, and document nodes.
- **XPath provides path expression:** XPath provides powerful path expressions, select nodes, or list of nodes in XML documents.
- **XPath is a core component of XSLT:** XPath is a major element in XSLT standard and must be followed to work with XSLT documents.
- **XPath is a standard function:** XPath provides a rich library of standard functions to manipulate string values, numeric values, date and time comparison, node and QName manipulation, sequence manipulation, Boolean values etc.
- **Path is W3C recommendation.**

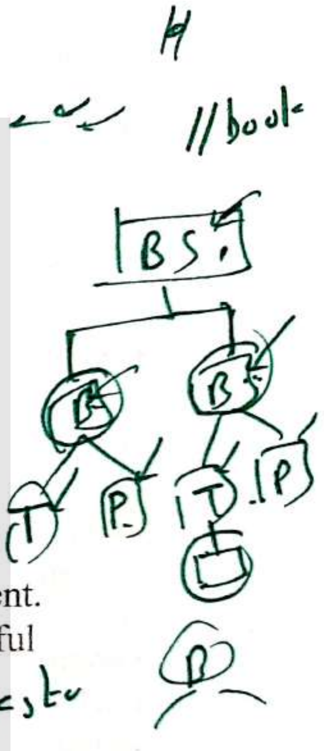
XPath Path Expressions

XPath uses path expressions to select nodes or node-sets in an XML document.) X

Education4Fun.com

We will use the following XML document in the examples below.)x

```
<?xml version="1.0" encoding="UTF-8"?>
<bookstore>
<book>
 <title lang="en">Harry Potter</title>
 <price>29.99</price>
</book>
<book>
 <title lang="en">Learning XML</title>
 <price>39.95</price>
</book>
</bookstore >
```



- Selecting

Nodes

XPath uses path expressions to select nodes in an XML document. The node is selected by following a path or steps. The most useful path expressions are listed below:

Education4Fun.com

Expression	Description
nodename	Selects all nodes with the name "*nodename*"
/	Selects from the root node
//	Selects nodes in the document from the current node that match the selection no matter where they are
.	Selects the current node
..	Selects the parent of the current node
@	Selects attributes

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

Path Expression	Result
bookstore	Selects all nodes with the name "bookstore"
/bookstore	Selects the root element bookstore
bookstore/book	Selects all book elements that are children of bookstore
//book	Selects all book elements no matter where they are in the document
bookstore//book	Selects all book elements that are descendant of the bookstore element, no matter where they are under the bookstore element
//@lang	Selects all attributes that are named lang

Education4Fun.com

Predicates

Predicates are used to find a specific node or a node that contains a specific value. Predicates are always embedded in square brackets.

In the table below we have listed some path expressions with predicates and the result of the expressions:

Path Expression	**Result**
/bookstore/book[1]	Selects the first book element that is the child of the bookstore element. **Note:** In IE 5,6,7,8,9 first node is[0], but according to W3C, it is [1]. To solve this problem in IE, set the SelectionLanguage to XPath: *In JavaScript:* *xml*.setProperty("SelectionLanguage","XPath");
/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"
/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

Education4Fun.com

(9) → XSL|T :- (XSL Transformation).

XSL :- Extensible stylesheet language.

HTML ⟶ CSS

→ So basically when we are working with html to do styling we use css

has predefined elements <h1> <p> <img> → browsers can understand css (as it's abt html)

<hello> {

→ But when it comes to XML if you want to style (how can a browser understand/analyze user defined tags) as XML has not predefined tags.

<hello>

→ So to style XML files, (WWW C) World wide Web Consortium (W3C) developed XSL which can act as a base stylesheet.

hello { }

(Few elements → template, value-of, apply-templates, for-each, sort, if, choose >

Education4Fun.com

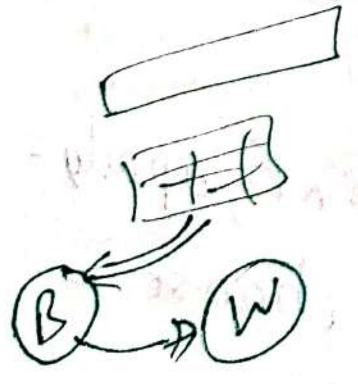
XSLT (Extensible Stylesheet Language Transformation) (XSL Transformat)

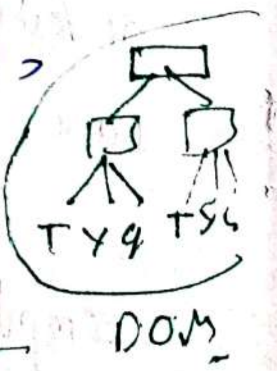
→ It is used to transform an XML document into another

(XML to another XML)
(XML to HTML) (XML to XHTML)
etc...

normal xml file:-

```
<?xml version = "1.0" encoding = "UTF-8" ?>
.xml
  <? xml-stylesheet href = "move.xsd"
      type = "text/xsd" ?>
                ~~css~~
                ~~xsl~~
  <collection> ✓
    <move> ✓
    < title > happy Gilmore </title>
    < year > 1996 </year>
    < genre > comedy </genre>
  </movie>
  <move>
    <title> Rango </title>
    <year> 2011 </year>
```

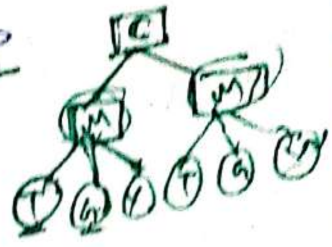




<genre> Animated </genre>

</movie>

</collection>



without xsd

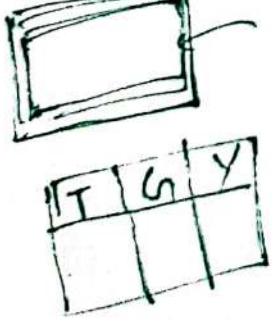
o/p :- happy Gilmore 1996 comedy Rang o

2011 Animated

more.xsd

```
<?xml version="1.0" encoding="UFT-8"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/
1999/XSL/Transform">
<xsl:template match="/collection">
<html>
  <body>
    <table border="1">
      <tr>
      <th> title </th>
      <th> Genre </th>
      <th> Year </th>
</tr>
```

100%



Education4Fun.com

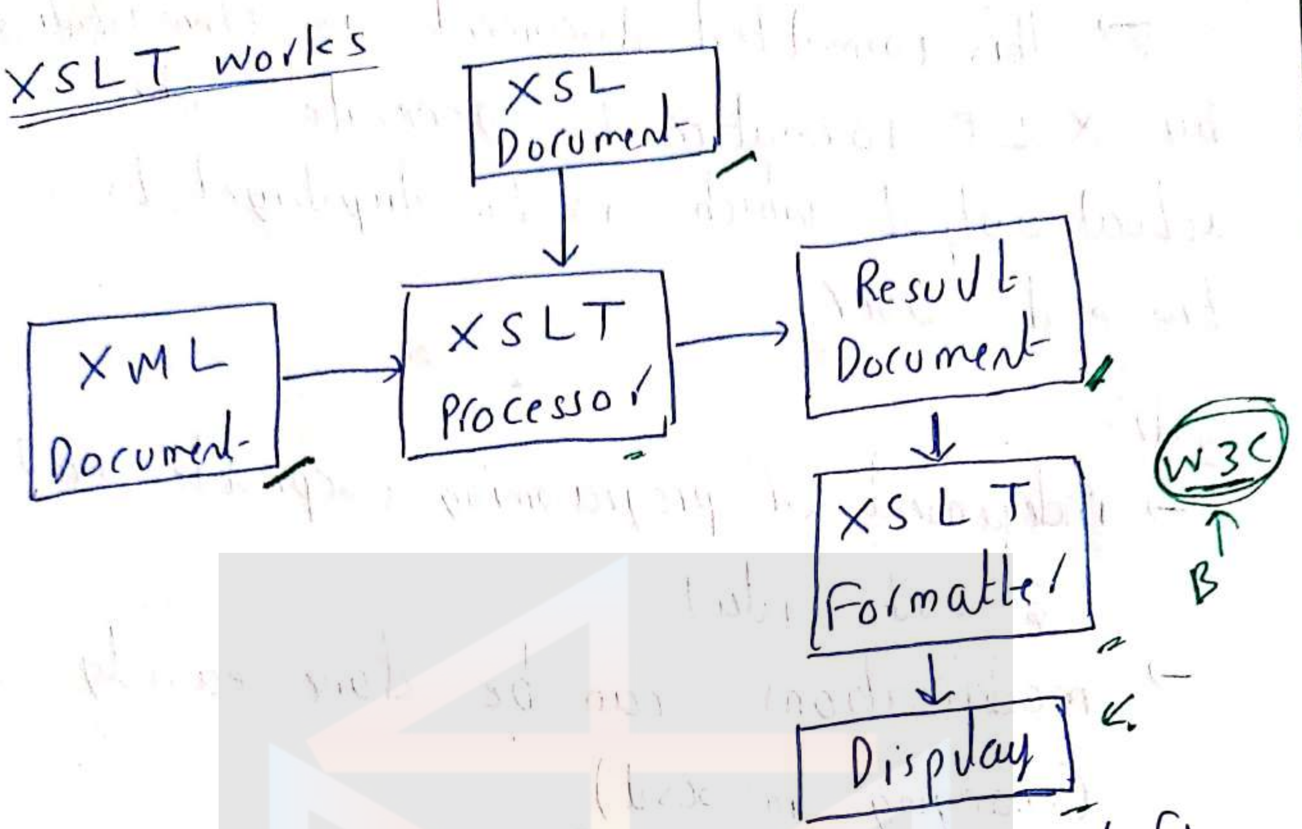
```
<xsl:for-each select="movie">
    <tr>
        <td><xsl:value-of
                    select="title"/></td>

| T | G | Y |
|---|---|---|
| x | G~ | 1~ |

        <td><xsl:value-of
                    select="genre"/></td>

        <td><xsl:value-of
                    select="year"/></td>

    </tr>
  </xsl:for-each>
 </table>
 </body>
</html>
</xsl:template>
</xsl:stylesheet>
```

o/p

title	Genre	year
happy Gilmore	~~1996~~ comedy	Com 1996
Rango	Animated	2011

Education4Fun.com

XSLT works



→ An XSLT stylesheet is used to define the transformation rules to be applied on the target XML. document.

→ XSLT stylesheet is defined in XML format

→ XSLT Processor takes the XSLT

→ XSLT stylesheet and applies the transformation rules on the target XML document & then it generates a formal document in the form of XML, HTML or text format.

Education4Fun.com

→ This formatted document is then utilized by XSLT Formatter to generate the actual output which is to displayed to the end user.

Adv:

→ independent of programming (separate xml & xsl files)

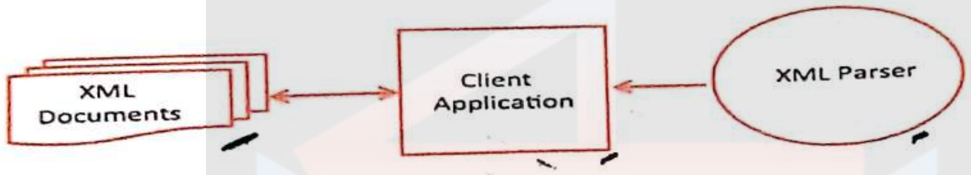
→ modifications can be done easily. (changing in xsl).

→ XML parser (/Processor → not sure)

Education4Fun.com

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. Let's understand the working of XML parser by the figure given below:



Types of XML Parsers

These are the two main types of XML Parsers:

Types of XML Parsers
These are the two main types of XML Parsers:

1. DOM
2. SAX

DOM (Document Object Model)
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.
Features of DOM Parser
A DOM Parser creates an internal structure in memory which is a DOM object and the client applications get information of the original XML document by invoking methods on this document object.
DOM Parser has a tree based structure. Advantages
1) It supports both read and write operations and the API is very simple to use.

2) It is preferred when random access to widely separated parts of a document is required. Disadvantages

1) It is memory inefficient. (consumes more memory because the whole XML document needs to loaded into memory).

2) It is comparatively slower than other parsers. SAX (Simple API for XML)

A SAX Parser implements SAX API. This API is an event based API and less intuitive.

Features of SAX Parser

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, it works like an event handler in Java. Advantages

1) It is simple and memory efficient.

2) It is very fast and works for huge documents. Disadvantages

1) It is event-based so its API is less intuitive.

2) Clients never know the full information because the data is broken into pieces.

Education4Fun.com