UNIT-2

HTML

1. HTML ? Features ? Advantages ? Disadvantages ? Example ?
2. HTML tags (common html tags)
3. HTML lists (ordered list, unordered list, Description list)
4. html tables
5. HTML forms
6. Attributes of form tag
7. HTML frames ? Disadvantages ? Attributes of frameset tag ?
8. HTML images
9. HTML Cascading Style sheets (HTML CSS), properties, Inline, Internal, External CSS.

XML

1. XML ? features and advantages ? Difference between HTML & XML
2. XML tags
3. XML Attributes and values
4. XML Schema or XML schema definition (XSD)
5. XML DOM (Document object model)
6. XHTML parsing XML Data., Difference between HTML and XHTML
7. XML DTD (Document type Definition), Internal DTD and External DTD
8. DOM and SAX parsers , Difference between DOM and SAX parser.

HTML:

html Stands for "Hyper text markup language"

⇒ HTML is the Standard markup language, for Creating web pages.

ie, it is not a programming language but it is a markup language.

A markup language is a set of markup tags. HTML uses markup tags to Represent web pages. ie, markup tags tells the web browser, Such as mozilla firefox, Google chrome, how to display the page. An HTML file must have an htm or html file extension.

HTML Elements and tags:

⇒ An HTML element is defined by Start tag, Some Content, end tag. ie, HTML element is everything from Start tag to end tag. HTML uses predefined tags.

Example: <tagname> Content </tagname>

Features of HTML:

1. It is easy to learn and easy to use.
2. platform independent
3. Images, videos, audio's Can be added to a web page
4. It is a markup language

Advantages:

1. HTML is used to build websites
2. It is Supported by all browsers.
3. it is integrated with other languages like CSS, JavaScript etc

Disadvantages:

A large amount of code has to be written to create simple web page and Security feature is not good.

BASIC STRUCTURE OF HTML CODE:

```
<!DOCTYPE html>        ← tells version of html
<html>                 ← root element of html
<head>
<title> my web page </title>
</head>
<body>
<h1> web technologies </h1>
<p> what is html </p>
</body>
</html>
```

o/p: webtechnologies
what is html

where,

`<head>`: used to contain page HTML metadata

`<title>` title of HTML page

`<body>` hold content of HTML

`<h1>`: HTML heading tag

`<p>`: HTML paragraph tag.

HTML VERSIONS

1991	1993	1995	2000	2014	2017
HTML	HTML+	HTML2.0	XHTML1.0	HTML5	HTML 5.2

HTML TAGS:

⇒ HTML tags are used to Create HTML documents and each HTML tags have different properties. HTML tags Contain three main parts ie, Starting tag, Content, ending tag. when a web browser reads an HTML document, browser reads it from top to bottom and left to right. HTML file must have some essential tags So that web browser Can differentiate between a Simple text and html text. you Can use as many tags you want as per your Code requirement.

⇒ All HTML tags must enclosed with <> these Angle brackets

⇒ Every tag in HTML perform different tasks. If you have used an open tag <tag>, then you must use a close tag </tag>. (except Some tags).

⇒ HTML tags are always written in lower Case letters.

BASIC HTML TAGS:

① html tag: It represents root of an html document.

Syntax: <head> . . . </head>

② title ~~head~~ tag: It defines the title or name of an HTML document

Syntax: <title> . . . </title>

③ body tag: It is used to define body Section of an HTML document.

Syntax: <body> . . . </body>

(4) heading tags:-

It is used for making heading in web pages. there are 6 types of heading tags. all of these tags are automatically in bold form.

① <h1> ... </h1> Bigger to [text size: 17]
② <h2> ... </h2>
⋮
⑥ <h6> ... </h6> Smaller

Some HTML tags are not closed, for example br and hr

 tag:- br stands for break line, it breaks the line of code (1)

<hr> tag:- hr stands for horizontal Rule. This tag is used to put a line across the web page

⇒ <!DOCTYPE> tag is used to define html document type

⇒ tag:- you can use tag to put any content in the center of the page

Other frequently used tags in html:-

<P> paragraph Tag </P>
<h2> Heading Tag </h2>
 BOLD Tag
 Italic tag
<u> Underline tag </u>

HTML LISTS :-

A list is a record of short pieces of related information or used to display the data or any information in web pages in the ordered or unordered form. HTML Lists are used to specify lists of information. All lists may contain one or more list elements. There are three different types of HTML lists.

1. Ordered List (or) Numbered List (ol)
2. Unordered List (or) Bulleted List (ul)
3. Description List (or) Definition List (dl)

(1) Ordered List:

In the ordered HTML lists, all the list items are marked with numbers by default. it is known as numbered list also The ordered list starts with <ol> tag and the list items starts with <li> tag.

Example

```
<!DOCTYPE>
<html>
<body>
<ol>
<li> Samsung </li>
<li> mi </li>
<li> vivo </li>
<li> iphone </li>
</ol>
</body>
</html>
```

output:

1. Samsung
2. ~~Bing~~ mi
3. vivo
4. iphone

There Can be different types of numbered list. they are

1. Numeric Numbers (1,2,3)
2. Capital Roman Numbers (I II III)
3. Small Roman Numbers (i ii iii)
4. Capital Alphabet (ABC)
5. Small Alphabet (abc)

To represent different ordered lists, there are 5 types of attributes in <ol> tag.

Type	Description
Type "1"	This is the default type, in this list items are numbered with numbers
Type "I"	In this list items are numbered with upper case roman numb
Type "i"	In this " " " with lower case Roman numbers
Type "A"	" " " with upper case letters
Type "a"	" " " lower case letters

Example to Display list in roman numbers lower case:

```
<!DOCTYPE html>
<html>
<body>
<ol type="i">
<li> Samsung </li>
<li> mi </li>
<li> vivo </li>
</ol>
</body>
</html>
```

output:

i. Samsung
ii. mi
iii. vivo

Example of Start Attribute :-

the Start Attribute is used with ol tag to Specify from where to start the list items.

Example:
Syntax: `<ol type = "1". Start = "5">` :- it shows numeric value starting with "5".

Example:

```
<! DOCTYPE html>
<html>
<body>
<ol type="1" Start = "5">
  <li> SamSung </li>
  <li> VIVO </li>
  <li> mi </li>
</ol>
</body>
</html>
```

output:

5. SamSung
6. VIVO
7. mi

② Unordered list :-

In HTML Unordered list, all the list items are marked with bullets. It is also known as bulleted list. The Unordered list starts with `<ul>` tag and list items start with `<li>` tag.

Example:

```
<!DOCTYPE>
<html>
<body>
<ul>
<li> SamSung </li>
<li> Sony </li>
<li> LG </li>
</ul>
</body>
</html>
```

output

- SamSung
- Sony
- LG

(3) Description list:

HTML also Supports description lists. A description list is a list of terms, with a description of each term.

=> the <dl> tag defines description list

=> " <dt> tag " the term (name)

=> " " <dd> tag describes each term

Example:

```
<!DOCTYPE html>
<html>
<body>
<dl>
<dt> c language </dt>
<dd> developed in the year 1972 </dd>
<dt> php </dt>
<dd> developed by Apache group </dd>
</dl>
```

```
</dl>
</body>
</html>
```

output:

c language - developed in the year 1972

php - developed by Apache group

where,

<dl> tag defines description list

<dt> tag defines data term

<dd> tag defines data definition (description)

---

3. HTML TABLES :-

⇒ HTML tables allow web developers to arrange data into rows and columns. There can be many columns in a row. We can create a table to display data in tabular form, using <table> element, with the help of <tr>, <td> <th> elements.

where <tr> tag defines table row

<th> tag defines table header

<td> tag defines table data.

Other HTML table tags include:

<caption> tag defines table caption

<thead> Groups the header content in a table

<tbody> " " body " " "

<tfoot> " " footer " "

Example:

```
<!DOCTYPE html>
<html>
<head>
<Style>
 table, th, td {
  border : 1px solid black;
   }
</style>
</head>
<body>
<h1> Marks of Student </h1>
<table>
 <tr>
  <th> Student </th>
  <th> marks </th>
 </tr>
  <tr>
   <td> Ramesh </td>
   <td> 100 </td>
  </tr>
```

<tr>
<td> shiva </td>
<td> 75 </td>
</tr>
</table>
</body>
</html>

output:

Marks of Student

Student	marks
Ramesh	100
shiva	75

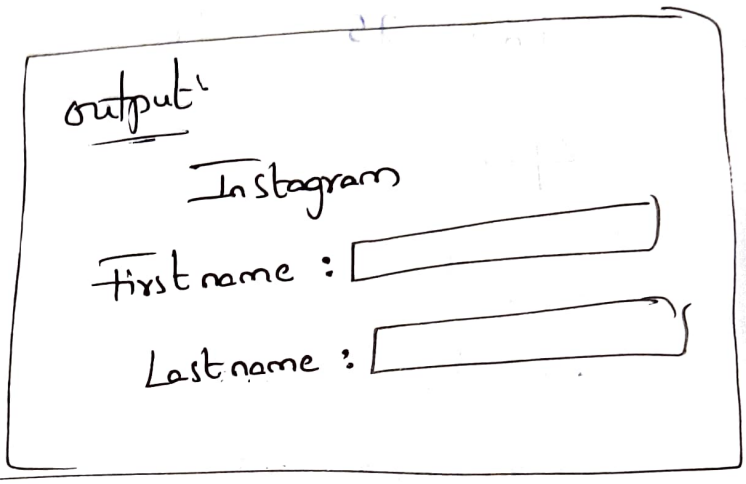
HTML FORMS :-

An HTML forms is used to collect user input. the user input is most often Sent to a Server for processing

Basic Example:

```
<!DOCTYPE html>
<html>
<body>
<h2> Instagram </h2>
<form action = "/action_page.php">
</form>
First name : <input type = "text" name = "first_name" />
<br>
```

```
Last name : <input type = "text" name= "last_name" />
</form>
</body>
</html>
```



⇒ An HTML form is a Section of document which contains Controls Such as text fields, password fields, checkboxes, radio buttons, Submit button, menus etc.

⇒ An HTML form facilitates the user to enter data that is to be Sent to the Server for processing Such as name, email address, password, phone number etc..

⇒ HTML forms are required if you want to collect data from the Site visitor. for example, If a user want to purchase Some items on internet, he/she must fill the form Such as Shipping address, Credit Card/ debit Card details So that item Can be Sent to given address.

There are four primary elements is used with in form tag:

① <input>:- it is whatever message that we want to insert in database.

② <Select>:- when ever we want to select particular data from data base we use Select tag.

③ <text area>:- Used to write data in the text boxes

④ <button>:- we can use various kinds of buttons like radio button, checkboxes etc..

Example 2: Designing of text field in web page:

```
<html>
<head>
<title> facebook </title>
</head>
<body> <h1> facebook </h1>
<form>
Enter name < input type = "text" size = "30" /> <br>
Enter password < input type = "text" size = "30" /> <br>
<input type = "button" value = "Send" />
<input type = "button" value = "Exit" />
</form>
</body>
</html>
```

output:

facebook

Enter name

Enter password

Send Exit

5. ATTRIBUTES OF FORM TAG:

there are three Attributes of form tag they are.

1. Action
2. method
3. Encryption type.

(1) Action:

⇒ It is used to determine, where to Send data.

⇒ It Specify url (Uniform resource locator) to which form data will be Submitted

⇒ we would Specify url of a program on a Server or an email

Example: <form action = "data.asp">

where asp - active Server page
jsp - java " "
CGI - Common gateway interface

(2) method:

It is an Attribute of form tag. It determines how form data will be Submitted.

the 2 options of these attribute is

Get & post method

Get: we get data from Server, no url is required. data passed through get request is visible on url browser

post: it is used to Submit form that have le amount of data. data passed through url is not visible on browse So it is Secure

3) Encryption type:

It Specify the format of data being Submitted. It Specify an Encoding protocol known as multipurpose internet mail Extension (MIME) for Security.

⇒ MIME Ensure that data doesnot become Corrupted when transmit across the internet.

Example:- <form action = "data.asp" method = "post" Encrypt = "plain/text">

HTML FRAMES

HTML frames are used to divide your browser window into multiple Sections where each Section Can load a Separate HTML document. A Collection of frames in the in the browser window is Known as a frameset. The window is divided into frames in a Similar way the tables are Organized into rows and Columns

Disadvantages of Frames:

There are few drawbacks with using frames, So it's never recommended to use frames in your webpage

(1) Some Smaller devices Cannot cope with frames because their Screen is not big enough to be divided up.

(2) Sometimes your page will be displayed differently on different Computers due to different Screen Resolution

(3) the browser's back button might not work as the user hopes

(4) there are Still few browsers that do not Support frames technology.

⇒ It Can display one or more than one html document in Same browser window. Each html document is called FRAME each frame is independent of other.

<frameset> tag is used to divide browser window.

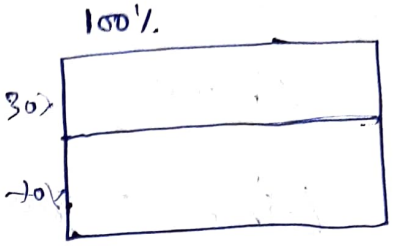
<body> tag is not required.

Attributes of frameset tag :-

- → Rows
- → Columns
- → frame border
- → Border Color
- → name

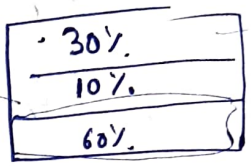
Rows:

It divides browser into row wise



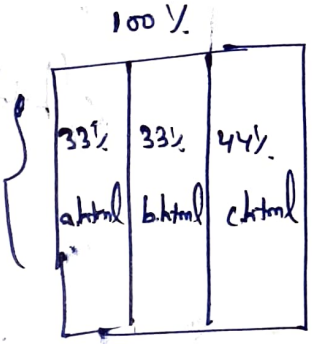
A.html.

```
<html>
{ <frameset rows = "30%, 70%">
  </frameset>  (or)
{ <frameset rows = "30%, 10%, *">
  </frameset>
</html>
```



② Columns:

It divides browser window column wise




```
<html>
 <frameset cols = "33%, 33%, 34%">
<frame src = "a.html">
<frame src = "b.html">
<frame src = "c.html">
 </frameset>
</html>
```

Example: To design an Indian flag:

a.html

```
<html>
<body bgcolor = "orange">
</body>
</html>
```

background

b.html.

```
<html>
<body>
</body>
</html>
```

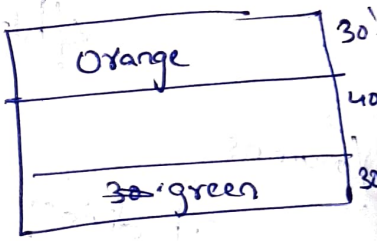
c.html

```
<html>
<body bgcolor = "green">
</body>
</html>
```

main.html:

```
<html>
<frameset rows = "30%, 40%, 30%">
<frame src = "a.html">
<frame src = "b.html">
<frame src = "c.html">
</frameset>
</html>
```

o/p:-



HTML IMAGES

⇒ Images can improve the design and the apperance of the web page.
The <img> tag is used to embed (place) an image in an HTML page.
we cannot import image directly to the web page, we need to use <img> tag to link images to web pages. This <img> tag creates a holding space for the referenced image.

the <img> tag has two required attributes.

① src:- Specifies the path to the image

② alt:- Specifies an alternate text for the image; if the image for some reason cannot be displayed

other attributes include

(1) Size: Specifies size of the image

(2) Title: It adds title to the image

(3) Alignment: where to place an image, like left, right, top or bottom

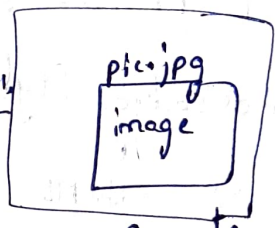
(4) Border Size: Images can be appeared with the border and we can increase size or the border or " " " decrease " " " "

Example:

```
<html>
<body>
<img src = "pic.jpg" align = "right">
</img>
</body>
</html>
```

Source file

o/p:

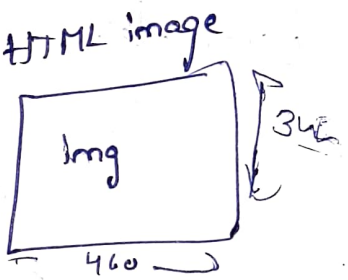


⇒ Images are not part of webpage file. They are separate files which are inserted into the page where it is used by the browser. We can also add source of image file to the html web page.

Example 2:

```
<!DOCTYPE html>
<html>
<body>
<h2> HTML Image </h2>
<img src = "image.jpg" alt = "Flowers" , width = "460"
                                height = "345" >
</img>
</body>
</html>
```

o/p:



HTML CSS :→ Stands for Cascading Style sheets.

⇒ Cascading Style sheets (CSS) is the language we use to Style an HTML document. CSS describes how HTML elements should be displayed. CSS Saves lot of work. it can control the layout of multiple web pages all at once.

⇒ with CSS you can control the color, font, size of text the spacing between elements, how elements are positioned and laid out., what background ~~colours~~ images and background colors are to be used etc~

⇒ The word Cascading means that a Style applied to a parent element will also applied to all children elements within the parent. So, if you Set the color of body text to "blue", all 'headings, paragraphs, and other elements with in the body will also get the Same color.

⇒ CSS Can be added to HTML document in 3 ways.

① Inline:-
② Internal
③ External

① Inline CSS:

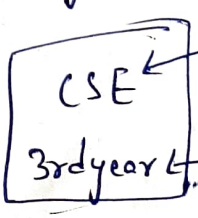
An inline CSS is used to apply a unique Style to a Single HTML element. An inline CSS uses the Style attribute of an HTML element.

The following example Sets the text color of the <h1> element to blue, and the text color of the <p> element to red.

Example:

```
<html>
<body>
<h1 Style = "color: blue;"> CSE </h1>
<p Style = "color: red;"> 3rdyear </p>
</body>
</html>
```

o/p:



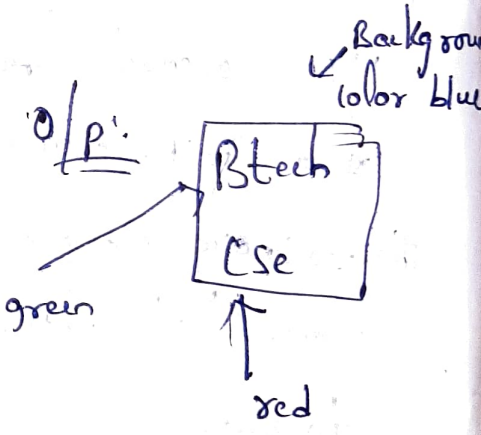
(2) Internal CSS:

An internal CSS is used to define a style for a single HTML page.

An internal CSS is defined in the <head> Section of an HTML page, within a <Style> element

Example:

```
<!DOCTYPE html>
<html>
<head>
<Style>
body { background-color: blue; }
h1 { color: green; }
p { color: red; }
</Style>
</head>
<body>
<h1> Btech </h1>
<p> CSE </p>
</body>
</html>
```

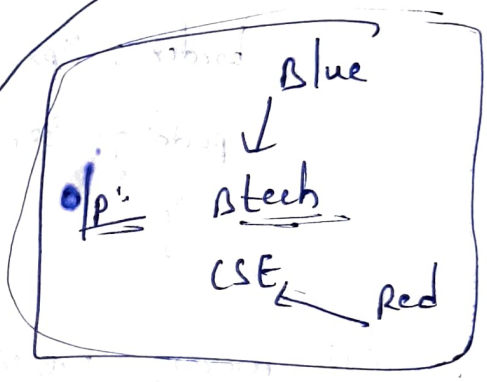


(3) External CSS:

An external Style Sheet is used to define the style for many HTML pages

Example:

```
<html>
  <head>
    <link rel = "stylesheet" href = "style.css">
  </head>
  <body>
    <h1> Btech </h1>
    <p> CSE </p>
  </body>
</html>
```



style.css

```
body {
    background-color : green;
}

h1 {
    color : blue;
}

p {
    color : red;
}
```

(2) CSS properties.

CSS color property defines the text color to be used

CSS font-family property defines the font to be used

CSS font-size " " " text size " "

CSS Border " " " border around an HTML element

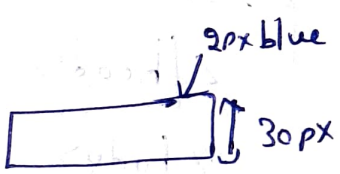
CSS padding " " " a space between text and border

Ex: p{

border : 2px solid blue; →

padding: 30px;

}



CSS margin property defines space outside the border

Xml:-

⇒ Xml Stands for Extensible Markup language. Xml donot Contain any predefined tags we Can Create and define our own tags in Xml. Xml was designed to Store and transport data.

⇒ XML was designed to be both human and machine readable. XML plays an important role in many different IT Systems. it is often used for distributing data over the internet. So it is important for all Software developers to have a good understanding of XML.

⇒ XML was designed to be Self descriptive. ie,

1. It has Sender information
2. It has receiver "
3. It has a heading
4. It has a message body

⇒ XML became a W3C Recommendation as early as in February 1998 (world wide web Consortium)

Features (or) Advantages:

1. Separate data from HTML
2. Simple data sharing
3. Simple data transport
4. Increases data availability
5. Simple platform change

(2)

⇒ Example:

```
<?xml version = "1.0" encoding = "UTF-8" ?>
    <email>
      <to> Raju </to>
      <from> Kishore </from>
      <heading> Important message </heading>
      <body> Come to College on monday </body>
    </email>
```

output.

email

To : Raju

from : Kishore

heading : important message

body : Come to College on monday

⇒ XML is designed to Carry data not to display data.

HTML VS XML :-

HTML	XML
(1) To display data	(1) to store and transport data
(2) markup language	(2) provide framework to define markup language
(3) non Case Sensitive	(3) Case Sensitive
(4) Static	(4) Dynamic

⑤ HTML Example

```
<html>
<body>
<p> Student 1 </p>
</body>
</html>
```

⑥ HTML Stands for hypertext markup language

⑦ HTML focus on ~~Appea~~ Appearance and presentation

⑧ Closing tags are not necessarily needed

⑤ XML example.

```
<college>
<class 1>
<Student> Raju </Student>
</ class 1>
</ college>
```

⑥ XML Stands for extensible markup language

⑦ XML focus only on the exchange of information.

⑧ Closing tags are used mandatory

Example 2: XML

```
<? xml version = "1.0" encoding = " UTF-8 " ?>
<college>
<class 1> CSE </class 1>
<Student> Shiva </Student>
<Roll no> 912 </ Roll no>
</college>
```

o/p:

```
college
  class 1 : CSE
  Student : shiva
  Roll no : 912
```

XML tags:

⇒ XML tags are important features of XML document. It is similar to HTML but XML is more flexable then HTML. It allows to create new tags (user defined tags). the first element of XML document is called root element. the Simple XML document Contains opening tag and closing tag. XML tags are case Sensitive. :<root> and <Root> both are different

Properties of XML tags:

① Every XML document must have a root tag which enclose the XML document. it is not necessary to name root tag as root. the name of root tag is any possible tag name.

Example:

```
<root>
<name> Sai </name>
<age> 22 </age>
</root>
```

Example-2

```
<class>
<name> Rushi </name>
<section> A </section>
</class>
```

② The Starting tag is Known as root tag. and we need to include in angle brackets <name>.

Example: <name>

③ the tag which is Started by Start tag must end

with the same tag with forward slash. in other words every XML document must be ended with end_tag. the end tag start with '<' followed by / and ends with >.

Example: <Name> Sai </Name>

Both must be same

forward slash at the ending

(3) XML tags are Case Sensitive. It means that <Root> and <root> both are different.

Example1: <Name> Sai </name>

it will display error because N and n both are different

Example2:

<Name> Sai </Name>

<name> Ravi </name>

(4) The text that appears between start-tag and end-tag is called Content. An element which has no content is known as empty.

Example: <name> </name>

it is empty as there is no content in between tags

(5)

Attributes and Values in XML:

XML elements can have attributes, just like HTML. Attributes are designed to contain data related to a specific element and attribute values must always be quoted. Either single or double quotes are used.

Example: `<person gender = "female"> </person>`

- `person` → elements
- `gender` → attribute
- `"female"` → value

(or)

`<person gender = 'female'>`

Things to be considered while using attributes:

1. attributes cannot contain multiple values (where elements can)
2. attributes cannot contain tree structures (where elements can)
3. attributes are not easily expandable for future change.

If we want to change attributes in feature that may be complicated

(4) Attributes are more difficult to be manipulated by program code.

Example: Same information can be Represented in two ways!

1st way:

<book publisher = "SIA"></book>

2nd way:

<book>
<publisher> SIA </publisher>
</book>

in first example publisher is used as an Attribute and in the Second example publisher is an element. Both examples provide the same information but it is good to avoid attribute in XML and use elements instead of attributes.

XML Schemas: (or) XML Schema definition (XSD):

⇒ An XML Schema describes the structure of an XML document

XML Schema language is also referred to as XML Schema definition (XSD)

⇒ XML schemas are written in XML and they are extensible. XML schemas support datatypes and name spaces It is like DTD but provide more control on XML structure

By using XML schema:

1. It is easier to describe document content
2. " " define restrictions of data
3. " " Validate Correctness of data.
4. " " convert data between different datatypes

⇒ Another great strength about XML schema is that they are written in XML and you don't have to learn a new language and you can use your XML editor to edit your schema files.

⇒ you can use XML parser to parse your schema files.

we can verify data with XML Schema.

Example:

Syntax:

Simple

Used only

text

to have te

contains less

and it c

⇒ you Can

and yo

⇒ with X

(a) Reuse

(b) Create

(c) Refer

⇒ wit

Example:

Syntax: <XS:schema xmlns:xs="">

Prefix

xmlns → it is name space

we define URI in it

Definition types

Simple type:

Used only in the context of text i.e, Simple type allows you to have text based element. It contains less attributes, child elements and it cannot be left empty.

Complex type

The complex type allows you to hold multiple attributes and elements. and it can also contain additional sub elements. It can be left empty.

⇒ you can manipulate your Schema with the XML DOM. and you can transform your Schema with XSLT.

⇒ with XSD (extensible Schema definition) you can :-

(a) Reuse your Schema in other Schemas
(b) Create your own datatype derived from the Standard types
(c) Reference multiple Schemas in same document.

⇒ with XML Schemas, the Sender can describe the data in a way that the receiver will understand.

⇒ In the XML world, hundreds of standardized XML formats are in daily use, many of these XML standards are defined by XML Schemas, and this XML Schema is an XML based and more powerful alternative to DTD

Example: Following Example shows how to use Schema:

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

<xs:schema xmlns:xs = "https://www.------" >
<xs:element name = "Contact" >
<xs:complexType >
<xs:Sequence >
<xs:element name = "name" type = "xs:String" />
<xs:element name = "company" type = "xs:String" />
<xs:element name = "phone" type = "xs:int" />
</xs:Sequence>
</xs:complexType>
</xs:element>
</xs:Schema>
```

Description of XML Schema:-

<xs:element name = "contact"> :- It defines the element name contact.

<xs:complexType> :- It defines that the element 'contact' is complex type.

<xs:sequence> :- It defines that the complex type is a sequence of elements.

<xs:element name = "name" type = "xs:string" /> :-
It defines that the element "name" is of string/text type.

XML DOM [Document object model]

⇒ The XML DOM defines a standard way for accessing and manipulating XML documents. It presents an XML document as a tree-structure. So, understanding the DOM is a must for anyone working with HTML or XML.

⇒ DOM stands for Document object model. DOM is a programming API (Application programming Interface) for HTML and XML document. DOM provides standard programming interface that can be used in a wide variety of environment and various Applications. This DOM can be used with any programming language.

=> XML DOM makes a tree-structure view of 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 attribut are all Known as nodes.

```
<? xml version = "1.0" ?>
<books>
  <book>
    <author> Rahul </author>
    <price> 256 </price>
    <publishdate> 05/10/21 </publishdate>
  </book>
  <publication>
    <publisher> Nagendra </publisher>
    <State> telangana </State>
  </publication>
</books>
```

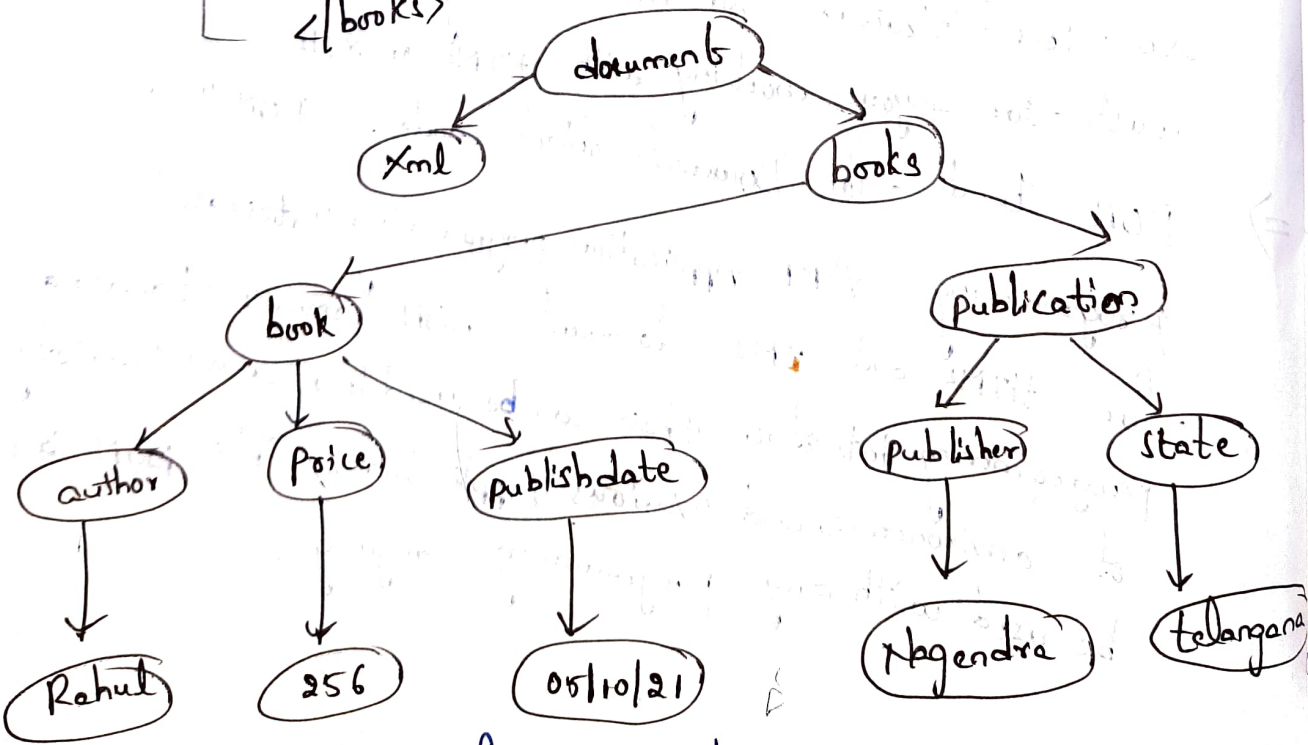


fig: XML document structure

⇒ Within the XML document structure, each circle represents a node, which is called an XML Node object. The XmlNode object is the basic object in DOM tree. Nodes have a single parent node; a parent node being a node directly above them. The only nodes that do not have a parent is the Document root, as it is the top level node and contains the document itself.

⇒ most nodes can have multiple child nodes, which are nodes directly below them. The DOM document is a collection of nodes or pieces of information organized in a hierarchy. The hierarchy allows a developer to find specific information easily by looking into the tree.

Advantages of XML DOM:-

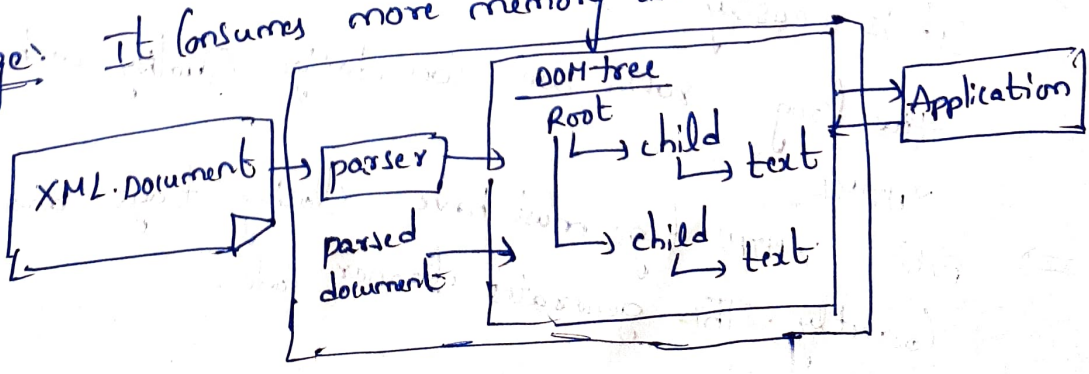
(1) XML DOM is language and platform independent

(2) XML DOM is traversable ie, information is arranged in hierarchy which makes us easy to find information

(3) XML DOM is modifiable ie, it is dynamic in nature providing the developer a scope to add, edit, move or remove nodes at any point on tree.

Disadvantage: It consumes more memory and it is slow.

XML DOM Structure :-



XHTML Parsing XML Data:-

⇒ XHTML Stands for Extensible Hypertext markup language. It is the next step to evolution of internet. XHTML was developed by world wide web Consortium (W3C). It helps web developers to make the transition from HTML to XML. Using XHTML, developers can enter the XML world with all the features of it,

⇒ The XHTML 1.0 is the first document type in the XHTML family and it is Recommended by W3C in 26 january 2000. The XHTML 1.1 is Recommended by W3C in 31 may 2001. at present we are using HTML 5. The XHTML document contains three parts, They are.

1. DOCTYPE :- It is used to declare a DTD.
2. head :- the head Section is used to declare title and other attributes.
3. body :- The body tag contains the content of web pages. It contains many tags.

⇒ XHTML documents are Validated with Standard XML tools. It is easily to mantain, Convert, edit document in the long run. we can design Quality web pages in XHTML.

⇒ All XHTML tags must have closing tags and are nested correctly. This generates cleaner code.

⇒ XHTML works in association with CSS to create web pages that can easily be updated.

HTML	XHTML
⇒ HTML or hypertext markup language is the main markup language for creating web pages.	XHTML (Extensible hypertext markup language) is a family of XML markup languages that mirror or extend versions of the widely used hyper text markup language (HTML)
⇒ not Supported by all browsers	⇒ Supported by all major browsers.
⇒ proposed by Tim Berners lee in 1987	⇒ it is world wide web Consortium (w3c) Recommendation in 2000.
⇒ It is Application of Standard Generalized markup language (SGML)	⇒ It is Application of XML
⇒ Extended from SGML	⇒ Extended from XML, HTML
⇒ changes needed inorder to work in Various devices	⇒ works in all devices without any changes.

⇒ XHTML is almost identical to HTML but it is stricter than HTML. XHTML is stricter than HTML in Syntax and Case Sensitive. XHTML documents are well-formed and parsed using Standard XML parsers.

⇒ many pages on the internet contains "bad" HTML i.e not follow the HTML rule. HTML code works fine in most browsers even if it does not follow HTML rules.

```
<html>
<head>
<title> example of bad html </title>
<body>
<h1> Bad HTML
<p> This is Bad html
</body>
```

⇒ the above code doesn't follow the HTML rule although it runs and this HTML is not supported in smaller devices. Unlike HTML, XHTML doesn't facilitates you to make badly formed code to be. where simple errors like missing out a closing tag are ignored by browser. but XHTML strictly follows code rules, code must be exactly how it is specified to be.

⇒ In XHTML → <!DOCTYPE> is mandatory.
→ Xmlns attribute in <html> is mandatory
→ <html>, <head>, <title>, <body> are mandatory.
⇒ → Elements must always be properly nested and properly closed
→ Elements and attributes must be in lower case
→ Attribute names must be in lower case and Attribute values Quoted.

XML DTD (Document Type Definition)

⇒ DTD Stands for Document type definition. It is a document that defines the structure of an XML document. It is also used to define elements and attributes of XML document.

⇒ This XML DTD is also used for performance Validation.

We can also called DTD as Document type declaration.

DTD Syntax:

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

Types of DTD

Internal DTD

elements are declared with in the XML files.

Syntax:

```
<! DOCTYPE root-element
[element-declaration] >
```

External DTD

elements are declared outside XML file

Syntax:

```
<! DOCTYPE root-element
SYSTEM "file-name" >
```

Internal DTD Example:

```
<? xml version = "1.0" encoding = "UTF-8" ?>
```

DTD

```
<! DOCTYPE Address [                          → root element
   <! Element Address (Name, Company, phone) >
   <! ELEMENT Name (#PCDATA) >
   <! ELEMENT Company (#PCDATA) >
   <! ELEMENT phone (#PCDATA) >
]>
```

xml.

```
<Address>
   <name> Sai </name>
   <company> DELL </company>
   <phone> 92488.... </phone>
</Address>
```

External DTD Example:

```
<?xml version="1.0" ?>
<!DOCTYPE Address SYSTEM "Add.dtd">
<Address>
<name> Sai </name>
<Company> DELL </Company>
<phone> 9248.. </phone>
</Address>
```

Address — root element

Add.dtd file

```
<!ELEMENT Address (Name, Company, phone)>
<!ELEMENT Name (#PCDATA)>
<!ELEMENT Company (#PCDATA)>
<!ELEMENT phone (#PCDATA)>
```

characteristics:

⇒ It defines the Compulsory and optional elements in XML document.

⇒ It Validates the Structure of XML document.

⇒ It check for the grammar of XML document.

Advantages:

⇒ we Can define our own format for the XML files by DTD

⇒ It helps in Validation of XML file

⇒ It provides us with a proper documentation.

⇒ It enables us to describe a XML document efficiently

Disadvantages:

⇒ DTD's are hard to read and maintain if they are large in size

⇒ it is not object oriented

⇒ The Documentation Support is limited.

⇒ DTD doesnot Support namespaces.

DOM and SAX parsers in java :-

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 programmers to use XML.

XML parser validates the document and check that the document is well formatted.

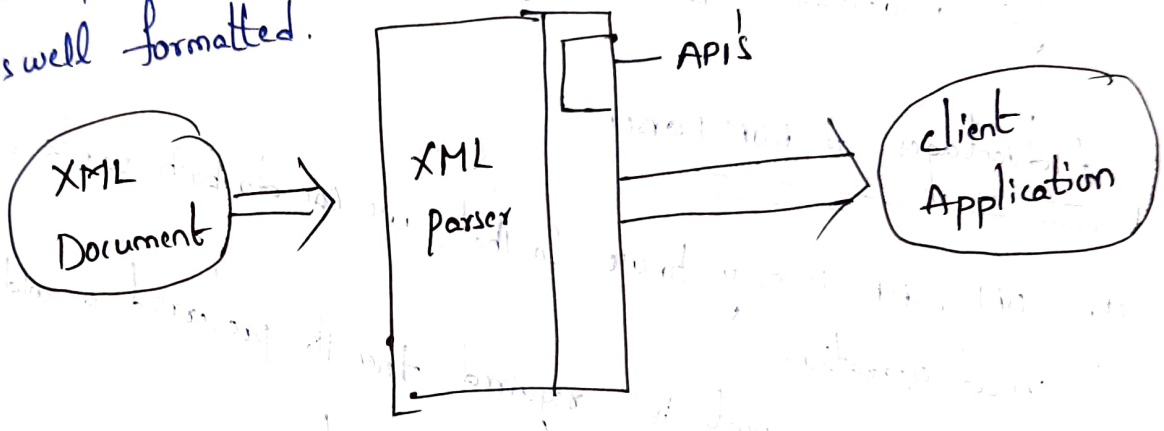


fig: working of XML parser

There are two main types of XML parsers :-

1. DOM
2. SAX

1) DOM (Document objects 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. DOM reads an entire document. It is useful when reading small to medium size XML files. The DOM API

Provides the classes to read and write an XML file. we can insert and delete nodes using DOM API.

Features of DOM parser:

(1) The internal structure can be created by DOM parser.

(2) Because of these internal structure, the client can get information about original XML doc

Advantages of DOM parser:

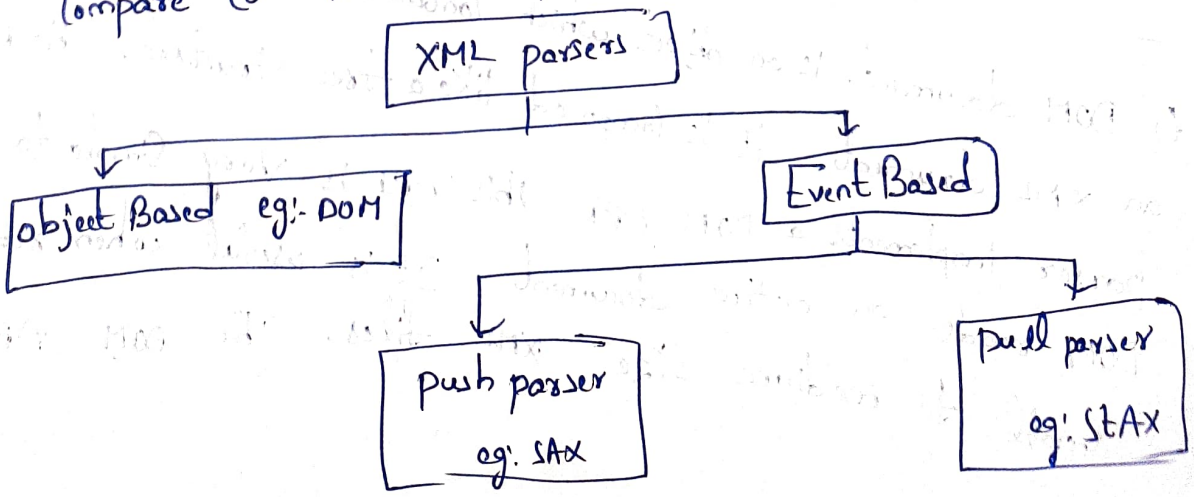
(1) DOM API is easy to use so that we can do both write and read operations

(2) when a document is required then it preferred a wide part that can be randomly accessed.

Disadvantages of DOM parser:

(1) Its efficiency of memory is not too good, it takes more memory because XML docs needed to load in there.

(2) Compare to the SAX parser, it is too slow.



SAX parser :

SAX represents Simple API for XML. and a SAX API is implemented by SAX parser. This API was called event-based API which provides interfaces on handlers. there are four handler interfaces.

1. Content handler
2. DTD handler
3. Entity Resolver
4. Error Handler interface.

⇒ It does not ~~not~~ Create any internal structure rather it takes the occurrences of Components of an input document as events, and then it tells the client what it reads through the input document.

It is Suitable for large XML files because it doesn't requires loading the whole XML file.

Features of SAX parser :

1. The internal structure Cannot be Created by SAX parser
2. these event-based SAX parsers work the same as event handler in Java.

Advantages of SAX parser :

1. Very Simple to use and good efficient of memory.
2. Its runtime is too fast and it Can be work for a bigger document or file System.

Disadvantages of SAX parser :

1. Its Ability to understand API's is too less than an event-based API
2. we Can't Know the full information because of lot of piece of data

SAX PARSER

1. It is called as Simple API for XML parsing
2. It's an event based parser
3. SAX parser is slower than DOM parser
4. Best for smaller size files
5. It is suitable for making XML files in java
6. The internal structure cannot be created by SAX parser
7. It is readonly
8. In SAX parser backward navigation is not possible
9. Suitable for efficient memory
10. A small part of the XML file is only loaded in memory

DOM PARSER

1. It is called as Document object model
2. It is tree structure.
3. DOM parser is faster than SAX Parser.
4. Best for larger size of file.
5. It is not good at making XML files in lower memory
6. The internal structure can be created by DOM parser.
7. It can insert or delete nodes
8. In DOM parser backward and forward search is possible
9. Suitable for large XML document.
10. It loads whole XML document in memory