Unit 2

Web Technology

Content

- **Web Page Designing**
- HTML
- Attributes and various tags.
- Images
- Tables
- Linking Documents and creating Frames.
- Form Controls
- CSS, Types and Syntax
- XML, components
- DTD
- XSD
- XML Parser and types.

What is HTML?

- HTML stands for Hypertext Markup Language, and it is the most widely used language to write Web Pages.
- Hypertext refers to the way in which Web pages (HTML documents) are linked together. Thus, the link available on a webpage is called Hypertext.
- As its name suggests, HTML is a Markup Language which means we use HTML to simply "mark-up" a text document with tags that tell a Web browser how to structure it to display.

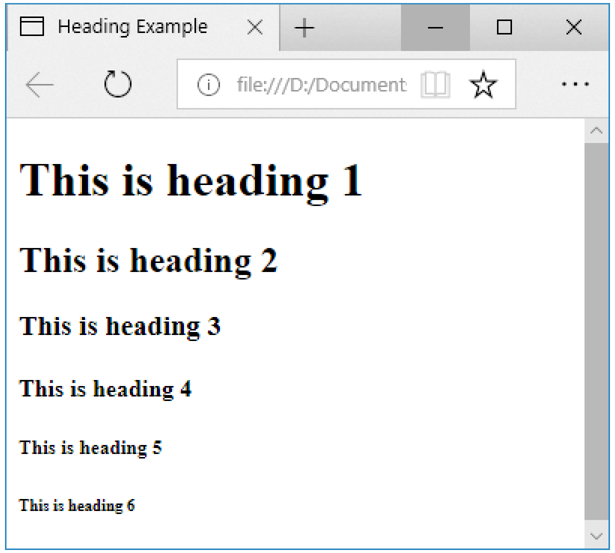
HTML <body> Tag Attributes

Attribute	Value	Description
alink	color	Specifies the color of an active link in a document
background	URL	Specifies a background image for a document
bgcolor	color	Specifies the background color of a document
link	color	Specifies the color of unvisited links in a document
text	color	Specifies the color of the text in a document
vlink	color	Specifies the color of visited links in a document

HTML Headings Tags

```
<html>
  <head>
    <title>Heading Example</title>
  </head>

  <body>
    <h1>This is heading 1</h1>
    <h2>This is heading 2</h2>
    <h3>This is heading 3</h3>
    <h4>This is heading 4</h4>
    <h5>This is heading 5</h5>
    <h6>This is heading 6</h6>
  </body>
</html>
```



This is heading 1

This is heading 2

This is heading 3

This is heading 4

This is heading 5

This is heading 6

HTML Elements

- An HTML element usually consists of a start tag and end tag, with the content inserted in between:

<tagname> Content goes here... </tagname>

- The HTML element is everything from the start tag to the end tag:

<p>My first paragraph.</p>

HTML Attributes

- All HTML elements can have attributes
- Attributes provide additional information about an element
- Attributes are always specified in the start tag
- Attributes usually come in name/value pairs like: name="value"

HTML Comment Tags

- Comment tags are used to insert comments in the HTML source code.
- Single Line Comment

```
<!-- Write your comments here -->
```

- Multiple Line Comment

```
<!--
 This is a multiline comment and it can
  span through as many as lines you like.
-->
```

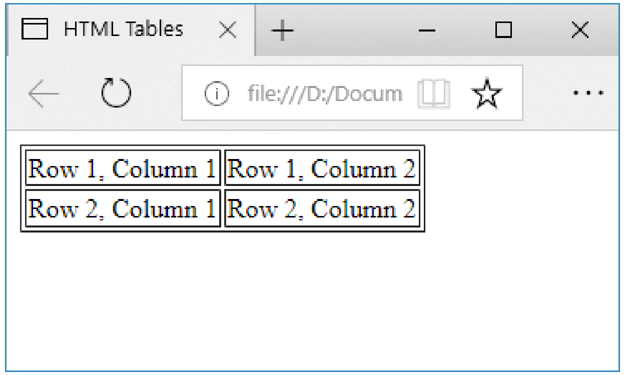
Images

- Set Image Alignment

```
<img src = "/html/images/test.png" alt = "Test Image" border = "3" align =
"right">
```

HTML - Tables

```
<html>
  <head>
    <title>HTML Tables</title>
  </head>
  <body>
    <table border = "1">
      <tr>
        <td>Row 1, Column 1</td>
        <td>Row 1, Column 2</td>
      </tr>
      <tr>
        <td>Row 2, Column 1</td>
        <td>Row 2, Column 2</td>
      </tr>
    </table>
  </body>
</html>
```



Row 1, Column 1	Row 1, Column 2
Row 2, Column 1	Row 2, Column 2

Linking Documents

- A link is specified using HTML tag <a>.
- This tag is called **anchor tag** and anything between the opening <a> tag and the closing </a> tag becomes part of the link and a user can click that part to reach to the linked document.
- Following is the simple syntax to use <a> tag.

<a href="*url*">*link text*</a>

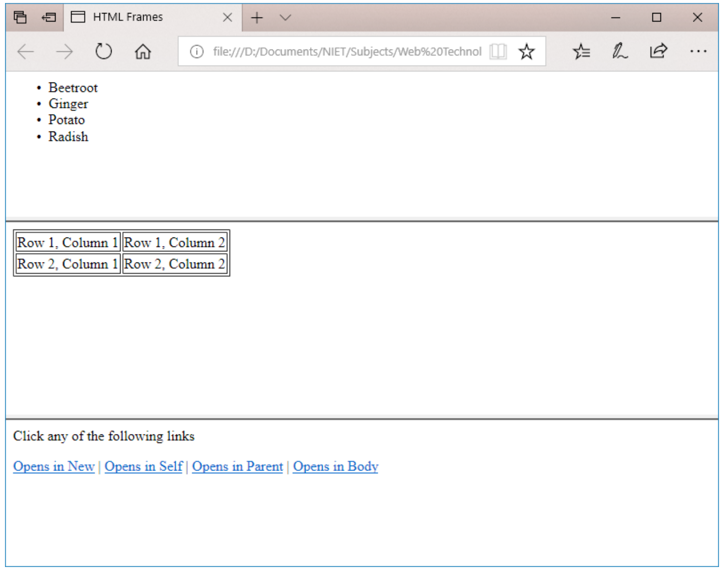
<a href = " *url* " target = "_self"> *link text* </a>

- Example:

<a href="https://www.w3schools.com/html/">Visit our HTML tutorial</a>

Creating Frames

```
<html>
  <head>
    <title>HTML Frames</title>
  </head>
  <frameset rows = "30%,40%,30%">
    <frame name = "top" src = "D:/Documents/NIET/Subjects/Web
Technologies (RCS-E12)/HTML/17 - UList.html" />
    <frame name = "main" src = "D:/Documents/NIET/Subjects/Web
Technologies (RCS-E12)/HTML/08 - Table.html" />
    <frame name = "bottom" src =
"D:/Documents/NIET/Subjects/Web Technologies (RCS-
E12)/HTML/23 - Hyperlink.html" />
    <noframes>
      <body>Your browser does not support frames.</body>
    </noframes>
  </frameset>
</html>
```



- Beetroot
- Ginger
- Potato
- Radish

Row 1, Column 1	Row 1, Column 2
Row 2, Column 1	Row 2, Column 2

Click any of the following links

Opens in New | Opens in Self | Opens in Parent | Opens in Body

HTML Form Controls

- There are different types of form controls that we can use to collect data using HTML form
  - Text Input Controls
  - Checkboxes Controls
  - Radio Box Controls
  - Select Box Controls
  - File Select boxes
  - Hidden Controls
  - Clickable Buttons
  - Submit and Reset Button

HTML Form Controls

- There are different types of form controls that we can use to collect data using HTML form
  - Text Input Controls
  - Checkboxes Controls
  - Radio Box Controls
  - Select Box Controls
  - File Select boxes
  - Hidden Controls
  - Clickable Buttons
  - Submit and Reset Button

What is CSS?

- CSS stands for Cascading Style Sheets. CSS is a standard style sheet language used for describing the presentation (i.e. the layout and formatting) of the web pages.
- Prior to CSS, nearly all of the presentational attributes of HTML documents were contained within the HTML markup (specifically inside the HTML tags); all the font colors, background styles, element alignments, borders and sizes had to be explicitly described within the HTML.

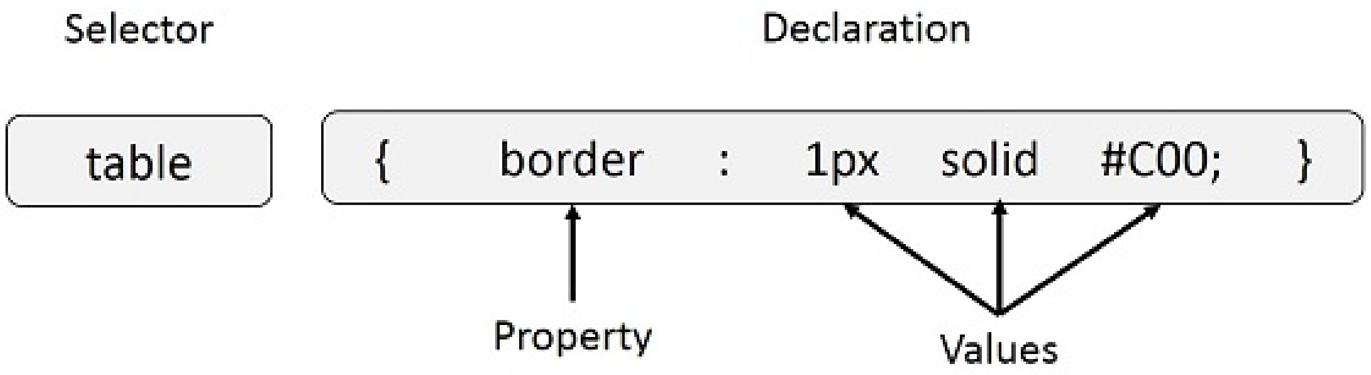
Types of CSS

- CSS can be added to HTML elements in 3 ways:
  - Inline style sheet - Using the style attribute in the HTML start tag.
  - Internal/Embedded style sheet - Using the <style> element in the head section of a document.
  - External style sheet - Using the <link> element, pointing to an external CSS file.

CSS Syntax

- Selectors are used to declare which of the markup elements a style applies to. This could be any tag like <h1> or <table> etc.
- The declarations that appear in the block that follows the selector may be applied to all elements of a specific type, or only those elements that match a certain attribute.

CSS Syntax



Syntax:	selector
	{
	property: value
	}

Example:	Table
	{
	border :1px solid #C00;
	}

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.

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

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) <u>XML separates data from HTML</u>

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.

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 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.

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".

```
<?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>
</note>
```

XML documents must contain a **root element.** This element is "the parent" of all other elements.
The elements in an XML document form a document tree. The tree starts at the root and branches to the lowest level of the tree.
All elements can have sub elements (child elements).

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

The terms parent, child, and sibling are used to describe the relationships between elements. Parent elements have children. Children on the same level are called siblings (brothers or sisters).
All elements can have text content and attributes (just like in HTML).

XML Attributes

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.

Let us take an example of a book publisher. Here, book is the element and publisher is the attribute.

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

Metadata should be stored as attribute and data should be stored as element.
<book>
<book category="computer"**>**
<author> A & B **</author>**
</book>

Data can be stored in attributes or in child elements. But there are some limitations in using attributes, over child elements.

XML Comments

XML comments are just like HTML comments. We know that the comments are used to make codes more understandable other developers.
XML Comments add notes or lines for understanding the purpose of an XML code. Although XML is known as self-describing data but sometimes XML comments are necessary.

An XML comment should be written as:

<!-- Write your comment-->

Rules for adding XML comments:

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

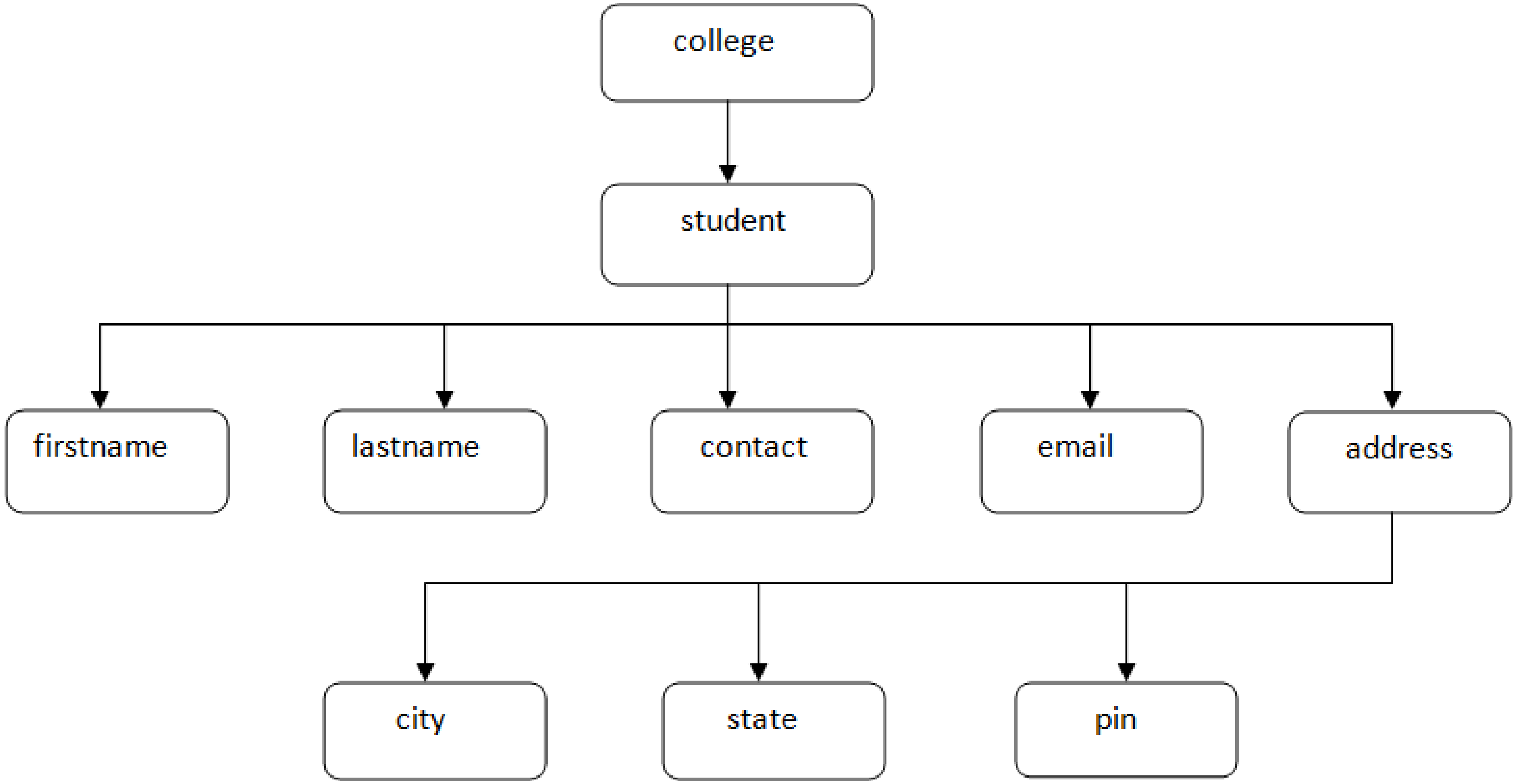
XML Tree Structure

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.

A tree structure contains root element (as parent), child element and so on. It is very easy to traverse all succeeding branches and sub-branches and leaf nodes starting from the root.

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

Let's see the tree-structure representation of that example.



XML Tree Rules

These rules are used to figure out the relationship of the elements. It shows if an element is a child or a parent of the other element.

Descendants: If element A is contained by element B, then A is known as descendant of B. In the above example "College" is the root element and all the other elements are the descendants of "College".

Ancestors: The containing element which contains other elements is called "Ancestor" of other element. In the above example Root element (College) is ancestor of all other elements.

XML Validation

A well formed XML document can be validated against DTD or Schema.
A well-formed XML document is an XML document with correct syntax. It is very necessary to know about valid XML document before knowing XML validation.

Valid XML document

It must be well formed (satisfy all the basic syntax condition)
It should be behave according to predefined DTD or XML schema

Rules for well formed XML

- It must begin with the XML declaration.
- It must have one unique root element.
- All start tags of XML documents must match end tags.
- XML tags are case sensitive.
- All elements must be closed.
- All elements must be properly nested.
- All attributes values must be quoted.
- XML entities must be used for special characters.

XML DTD

What is DTD

DTD stands for **Document Type Definition**. It defines the legal building blocks of an XML document. It is used to define document structure with a list of legal elements and attributes.

Purpose of DTD

Its main purpose is to define the structure of an XML document. It contains a list of legal elements and define the structure with the help of them.

Checking Validation

Before proceeding with XML DTD, you must check the validation. An XML document is called "well-formed" if it contains the correct syntax.

A well-formed and valid XML document is one which have been validated against DTD.

Visit **http://www.xmlvalidation.com** to validate the XML file.

Valid and well-formed XML document with DTD

Let's take an example of well-formed and valid XML document. It follows all the rules of DTD.

employee.xml

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

In the above example, the DOCTYPE declaration refers to an external DTD file. The content of the file is shown in below paragraph.

employee.dtd

```
<!ELEMENT employee (firstname,lastname,email)>
<!ELEMENT firstname (#PCDATA)>
<!ELEMENT lastname (#PCDATA)>
<!ELEMENT email (#PCDATA)>
```

Description of DTD

<!DOCTYPE employee : It defines that the root element of the document is employee.
<!ELEMENT employee: It defines that the employee element contains 3 elements "firstname, lastname and email".
<!ELEMENT firstname: It defines that the firstname element is #PCDATA typed. (parse-able data type).
<!ELEMENT lastname: It defines that the lastname element is #PCDATA typed. (parse-able data type).
<!ELEMENT email: It defines that the email element is #PCDATA typed. (parse-able data type).

XML DTD with entity declaration

A doctype declaration can also define special strings that can be used in the XML file.

An entity has three parts:

- An ampersand (&)
- An entity name
- A semicolon (;)

Syntax to declare entity:

<!ENTITY entity-name "entity-value">

Let's see a code to define the ENTITY in doctype declaration.

author.xml

```
<?xml version="1.0" standalone="yes" ?>
<!DOCTYPE author [
  <!ELEMENT author (#PCDATA)>
  <!ENTITY sj "Sonoo Jaiswal">
]>
<author>&sj;</author>
```

In the above example, sj is an entity that is used inside the author element. In such case, it will print the value of sj entity that is "Sonoo Jaiswal".

XML Schema

XML schema is a language which is used for expressing constraint about XML documents. There are so many schema languages which are used now a days for example Relax- NG and XSD (XML schema definition).

An XML schema is used to define the structure of an XML document. It is like DTD but provides more control on XML structure.

Checking Validation

An XML document is called "well-formed" if it contains the correct syntax. A well-formed and valid XML document is one which have been validated against Schema. Visit **http://www.xmlvalidation.com** to validate the XML file against schema or DTD.

XML Schema Example

employee.xsd

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.javatpoint.com"
xmlns="http://www.javatpoint.com"
elementFormDefault="qualified">

<xs:element name="employee">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="firstname" type="xs:string"/>
      <xs:element name="lastname" type="xs:string"/>
      <xs:element name="email" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

</xs:schema>
```

employee.xml

```
<?xml version="1.0"?>
<employee
xmlns="http://www.javatpoint.com"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.javatpoint.com employee.
xsd">

  <firstname>vimal</firstname>
  <lastname>jaiswal</lastname>
  <email>vimal@javatpoint.com</email>
</employee>
```

Description of XML Schema

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

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

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

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

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

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

XML Schema Data types

There are two types of data types in XML schema.

1. simpleType
2. complexType

simpleType

The simpleType allows you to have text-based elements. It contains less attributes, child elements, and cannot be left empty.

complexType

The complexType allows you to hold multiple attributes and elements. It can contain additional sub elements and can be left empty.

DTD vs XSD

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.
3)	DTD **doesn't support datatypes**.	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.

XML DOM

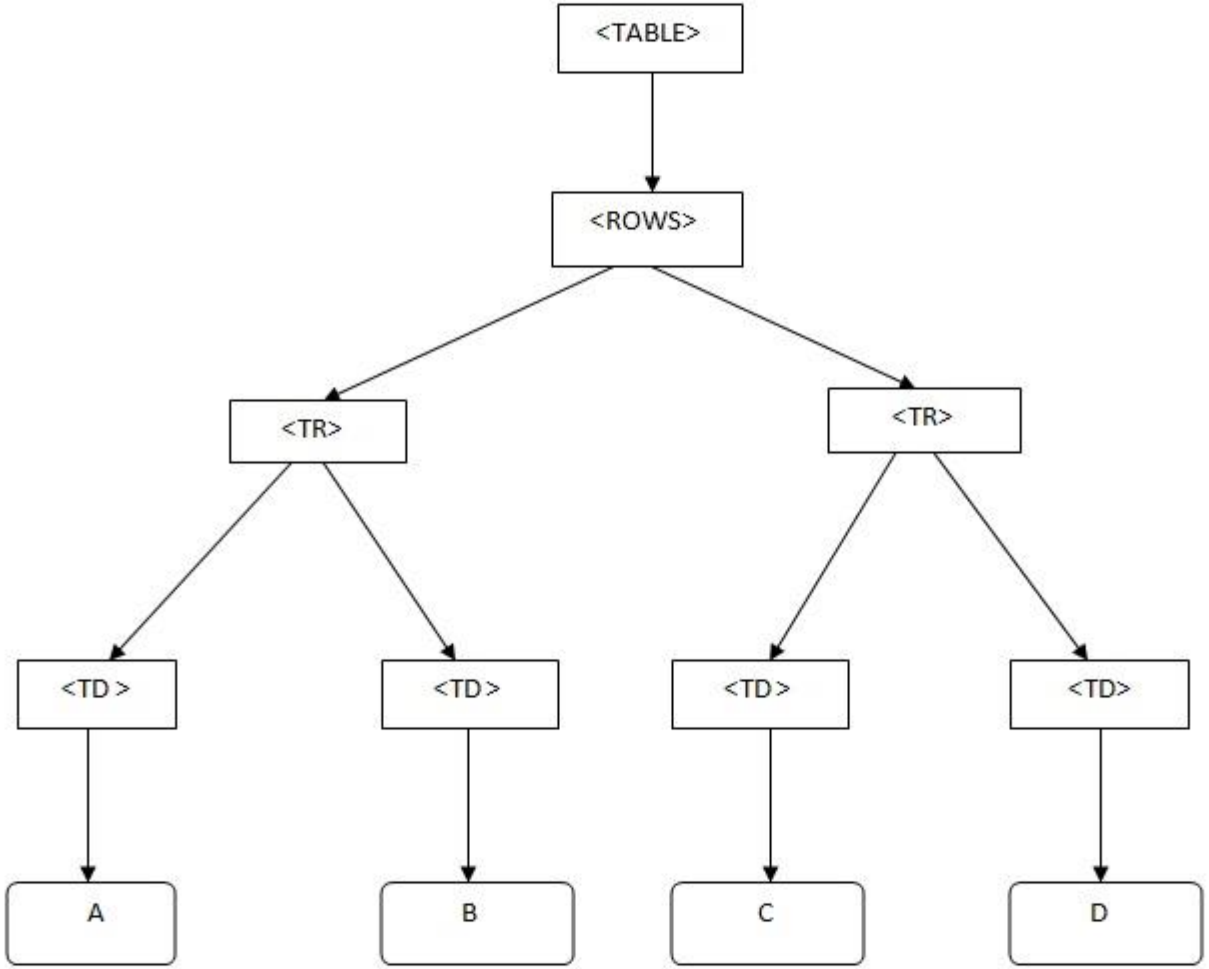
DOM is an acronym stands for Document Object Model. It defines a standard way to access and manipulate documents. The Document Object Model (DOM) is a programming API for HTML and XML documents. It defines the logical structure of documents and the way a document is accessed and manipulated.

As a W3C specification, one important objective for the Document Object Model is to provide a standard programming interface that can be used in a wide variety of environments and applications. The Document Object Model can be used with any programming language.

XML DOM defines a standard way to access and manipulate XML documents.

For example, consider this table, taken from an HTML document:

```
<TABLE>
<ROWS>
<TR>
<TD>A</TD>
<TD>B</TD>
</TR>
<TR>
<TD>C</TD>
<TD>D</TD>
</TR>
</ROWS>
</TABLE>
```


<TABLE>
<ROWS>
<TR>
<TR>
<TD >
<TD >
<TD >
<TD>
A
B
C
D

XML Components

- XML Prolog/PI
- Components of Documents
    - Tags
    - Elements
    - Attributes
    - Entity
    - Comments
    - CDATA (Character Data)

XML Components

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

<card>
        <name>John Doe</name>
        <title>CEO, Widget Inc.</title>
        <email>john.doe@widget.inc</email>
        <phone>(202) 456-1414</phone>
</card>
```

What is Document Type Definition (DTD) ?

- The XML Document Type Declaration, commonly known as DTD, is a way to describe XML language precisely.
- DTDs check vocabulary and validity of the structure of XML documents against grammatical rules of appropriate XML language.
- An XML DTD can be either specified inside the document, or it can be kept in a separate document and then liked separately.

Syntax for DTD

```
<!DOCTYPE element DTD identifier
[
  declaration1
  declaration2
  ........
]>
```

example

```
<?xml version = "1.0" encoding = "UTF-8" standalone = "yes" ?>
  <!DOCTYPE address [
    <!ELEMENT address (name,college,phone)>
    <!ELEMENT name (#PCDATA)>
    <!ELEMENT college (#PCDATA)>
    <!ELEMENT phone (#PCDATA)>
  ]>
<address>
  <name>Tarun Kumar</name>
  <college>NIET, Greater Noida</college>
  <phone> +91-8899775511</phone>
</address>
```

About Standards of XSD

- "DTD" was the first formalized standard, but is rarely used anymore.
- "XDR" was an early attempt by Microsoft to provide a more comprehensive standard than DTD. This standard has pretty much been abandoned now in favor of XSD.
- "XSD" is currently the de facto standard for describing XML documents. There are 2 versions in use 1.0 and 1.1, which are on the whole the same. An XSD schema is itself an XML document, there is even an a XSD schema to describe the XSD standard.
- There are also a number of other standards.

What is XML Schema Definition (XSD)?

- XML Schema is commonly known as XML Schema Definition (XSD). It is used to describe and validate the structure and the content of XML data. XML schema defines the elements, attributes and data types. Schema element supports Namespaces. It is similar to a database schema that describes the data in a database.
- XML Schema is an XML-based alternative to DTD.
- An XML document with correct syntax is called "Well Formed".
- An XML document validated against an XML Schema is both "Well Formed" and "Valid".

XML Parser

- Common XML Parsers include
  - Microsoft XML Parser (MSXML)
  - Apache Xerces
  - The Oracle XML parser.
- We can write an application that uses any of these parsers.
- Some XML parsers are also available as prepackaged software that install automatically.

Types of XML Processor

- validating types
- non-validating types

Tree-based Parser - **D**ocument **O**bject **M**odel (DOM)

- Tree-based parsers referred to as Document Object Model (DOM) parsers.
- The DOM is a W3C recommendation that provides an application programming interface (API) to an XML document.
- Any application can use this API to manipulate an XML document, read information, add new nodes, and edit the existing content.

Event-based Parser - Simple API for XML (SAX)

- SAX is an acronym for Simple API for XML. SAX Parser parses the XML file line by line and triggers events when it encounters opening tag, closing tag or character data in XML file. This is why SAX parser is called an event-based parser.
- SAX is not a W3C recommendation, but it does enjoy support from both large and small software companies.

What is DHTML?

DHTML stands for **Dynamic Hypertext Markup language** i.e., **Dynamic HTML**.
Dynamic HTML is not a markup or programming language but it is a term that combines the features of various web development technologies for creating the web pages dynamic and interactive.
The DHTML application was introduced by Microsoft with the release of the 4th version of IE (Internet Explorer) in 1997

Components of Dynamic HTML

DHTML consists of the following four components or languages:

- HTML 4.0
- CSS
- JavaScript
- DOM.

HTML 4.0

HTML is a client-side markup language, which is a core component of the DHTML. It defines the structure of a web page with various defined basic elements or tags.

CSS

CSS stands for Cascading Style Sheet, which allows the web users or developers for controlling the style and layout of the HTML elements on the web pages.

JavaScript

JavaScript is a scripting language which is done on a client-side. The various browser supports JavaScript technology. DHTML uses the JavaScript technology for accessing, controlling, and manipulating the HTML elements. The statements in JavaScript are the commands which tell the browser for performing an action.

DOM

DOM is the document object model. It is a w3c standard, which is a standard interface of programming for HTML. It is mainly used for defining the objects and properties of all elements in HTML.

Uses of DHTML

Following are the uses of DHTML (Dynamic HTML):

- It is used for designing the animated and interactive web pages that are developed in real-time.
- DHTML helps users by animating the text and images in their documents.
- It allows the authors for adding the effects on their pages.
- It also allows the page authors for including the drop-down menus or rollover buttons.
- This term is also used to create various browser-based action games.
- It is also used to add the ticker on various websites, which needs to refresh their content automatically.

Features of DHTML

Following are the various characteristics or features of DHTML (Dynamic HTML):

- Its simplest and main feature is that we can create the web page dynamically.
- **Dynamic Style** is a feature, that allows the users to alter the font, size, color, and content of a web page.
- It provides the facility for using the events, methods, and properties. And, also provides the feature of code reusability.
- It also provides the feature in browsers for data binding.
- Using DHTML, users can easily create dynamic fonts for their web sites or web pages.
- With the help of DHTML, users can easily change the tags and their properties.
- The web page functionality is enhanced because the DHTML uses low-bandwidth effect.

Difference between HTML and DHTML

HTML (Hypertext Markup language)	**DHTML (Dynamic Hypertext Markup language)**
1. HTML is simply a markup language.	1. DHTML is not a language, but it is a set of technologies of web development.
2. It is used for developing and creating web pages.	2. It is used for creating and designing the animated and interactive web sites or pages.
3. This markup language creates static web pages.	3. This concept creates dynamic web pages.
4. It does not contain any server-side scripting code.	4. It may contain the code of server-side scripting.
5. The files of HTML are stored with the .html or .htm extension in a system.	5. The files of DHTML are stored with the .dhtm extension in a system.
6. A simple page which is created by a user without using the scripts or styles called as an HTML page.	6. A page which is created by a user using the HTML, CSS, DOM, and JavaScript technologies called a DHTML page.
7. This markup language does not need database connectivity.	7. This concept needs database connectivity because it interacts with users.