

XML :- XML is a extensible Markup Language.

- XML is designed to store and transport data. It stores the data in text format.
- XML was designed in late 90's.
- It was created to provide an easy to use and store self describing data.
- XML is not a replacement for HTML.
- XML is design to be self-descriptive.
- XML is design to carry data, not to display data.
- XML tags are not predefined. ~~tag~~  
You must define your own tags.
- XML is platform independent and language independent.
- XML is a W3C Recommendation.

Why XML ?

XML is 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 program and platforms.

- You can communicate b/w two platform which are generally very difficult.

## \* Feature and Advantage of XML :-

- 1.) XML allow sharing
- 2.) XML simplifies data transport.
- 3.) XML simplifies platform change
- 4.) XML increase data availability
- 5.) XML separate data from HTML.
- 6.) XML can be used to create new Internet Languages :-

A lot of new internet language are created with XML.

Ex :- XHTML, WSDL, RSS , OWL etc.

## \* Syntax of XML document :-

### XML Declaration :-

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

Example :-

```

<?xml version="1.0"?>
<contact-info>
  <name> Renu </name>
  <company> JavalPoint </company>
  <phone> (011) 1234567 </phone>
</contact-info>

```

## XML Syntax Rule's

- (1) XML documents Must have a Root Element.  
In a given example, `<contact-info>` is a root element.
- (2) The XML Prolog

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

This line known as XML prolog.

- (3) All XML Elements must have a closing Tag  
Ex :- `<name> Renu </name>`
- (4) XML Tags are case sensitive

The tag `<letter>` is differ from `<Letter>`.  
→ Opening and closing tag must be written in same case.

- (5) XML Attribute Values must always be quoted

XML elements can have attributes just like in HTML:

Ex :-

```
<note date="12/05/23">
```

Note XML-elements also called XML-nodes or XML-tags.

# Tree Structure in data Organization

DELTA Pg No.

or

## XML document Structure :-

XML document has a self descriptive structure.

It form a tree structure that starts at "the root" and branches to "the leaves", which is referred as an XML tree.

The tree structure makes easy to describe an XML document.

All elements can have sub-elements:

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

### Example :-

```
<?xml version="1.0"?>
<college>
  <Student> +
    <FirstName> Renu </FirstName>
    <LastName> Kumar </LastName>
    <contact> 0999088678 </contact>
    <email> tanna@gmail.com </email>
```

<address>

<city> Ghaziabad </city>

<state> Uttar Pradesh </state>

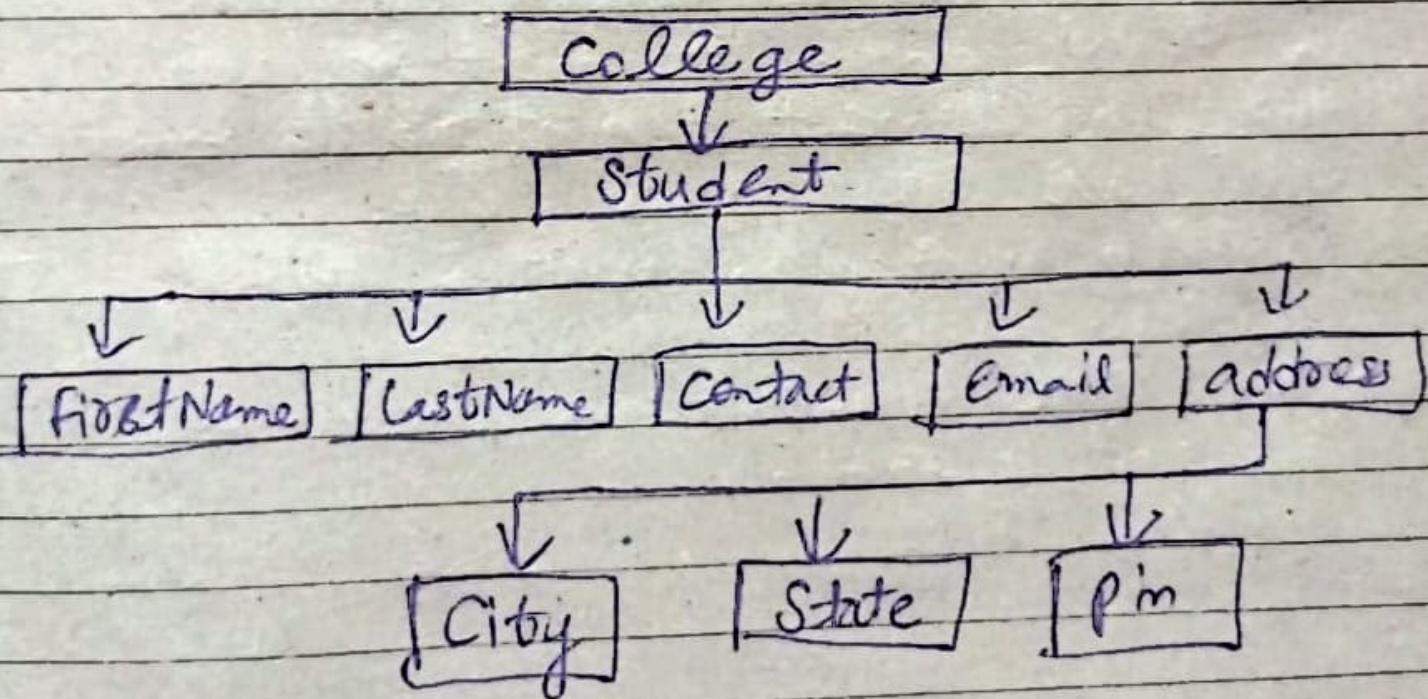
<pin> 121007 </pin>

</address>

</student>

</college>

### Tree Structure of XML Document



## \* Well Formed XML :-

A XML document is called well-formed if satisfied by the W3C.

These rules are :-

- 1.) A well-formed XML document must have a corresponding end tag for all of its start tags.
- 2.) Nesting of element within each other in an XML document must be proper.
- 3.) In each element two attributes must not have the same value.
- 4.) Markup characters must be properly specified.
- 5.) An XML document can contain only one root element.

## DTD and its Structure :-

The XML Document Type Declaration (DTD),

commonly known as DTD, is a way to describe XML language precisely.

- It is used to define document structure with a list of legal elements and attributes.
- DTD checks vocabulary and validity of the structure of XML document against grammatical rules of appropriate XML language.
- An XML DTD can be either specified inside the document.

## Syntax :-

<!DOCTYPE element DTD identifier

[

declaration1

declaration2

-----

]>

In above Syntax,

- The DTD starts with `<!DOCTYPE` delimiter.
- DTD identifier is an identifier for the document type definition.
- `element` tells the parser to parse the document from the specified root element.
- The square brackets `[]` enclose an optional list of entity declaration called Internal Subset.

Internal DTD :-

A DTD is referred to as an internal DTD if elements are declared within the XML files.

Syntax

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

Here,

- root element is the name of root element.
- element - declaration is where you

declare the element.

Example 6:-

(Start Declaration)

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

standalone="yes"?>

<!DOCTYPE address > (DTD)

{ <!ELEMENT address'(name,company,phone)>  
<!ELEMENT name (#PCDATA)>  
<!ELEMENT company (#PCDATA)>  
<!ELEMENT phone (#PCDATA)>  
}> → (End declaration)

<address>

<name> Tanmay </name>

<company> JavaPoint </company>

<phone> (011) 123-4567 </phone>

</address>

\* External DTD :- In external DTD elements are declared outside the XML file.

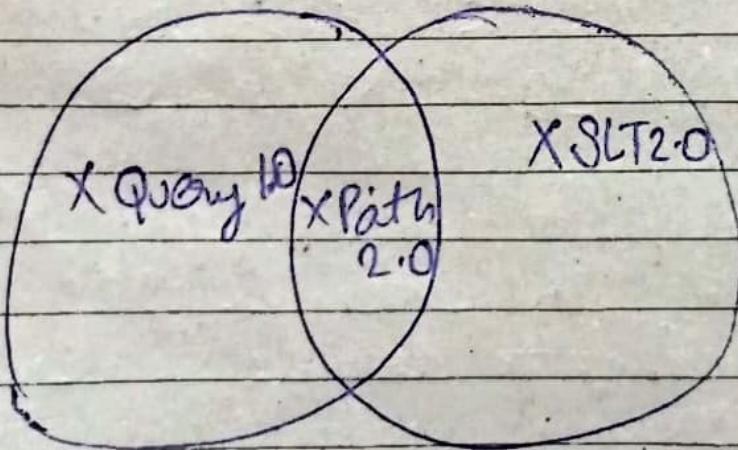
Syntax

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

## \* XPath :-

XPath is a component of XSLT standard provided by W3C. It is used to traverse the element and attribute of an XML document.

- XPath is a W3C recommendation.
- XPath provides different types of expressions to retrieve relevant information from the XML document.
- Its syntax for defining parts of XML document.



## Important Feature 6

- XPath defines structure (Parts of XML document)
- XPath provides path expression for selecting nodes, list of nodes in XML document.
- XPath is a core component of XSLT.
- XPath provides a rich library of standard functions to manipulate string, numeric value & data & time.
- XPath is W3C recommendation.