

Unit IV

Page No.

Date

XML

XML stands for Extensible markup language. XML is a markup language that defines a set of rules for encoding documents in a format that is both human-readable and machine-readable.

- * It can run on any platform, operating system or environment.
- * XML is designed to provide developer with a mechanism to describe their content better.
- * It ~~has~~ has been developed to make exchange of data on the web much easier and more efficient, by allowing developers to write their own document type definition (DTD).

Features of XML :-

- ① Platform Independent :- It is platform

- ① Platform Independent :- It is created using standard text file & These files are compatible with all platforms.
- ② Designed to carry data :- XML was designed to focus on what data is where HTML was designed to display data.
- ③ Separate data from HTML :- HTML is for displaying where XML is for storing or describing data.
- ④ Free & Extensible :- XML tags are Not predefined. you can create your own tag as per your requirements.
- ⑤ Modularity :- You can attach multiple DTD's on one document or single DTD for multiple documents.
- ⑥ Container tag :- XML allow user defined tag. But it must be container tag.

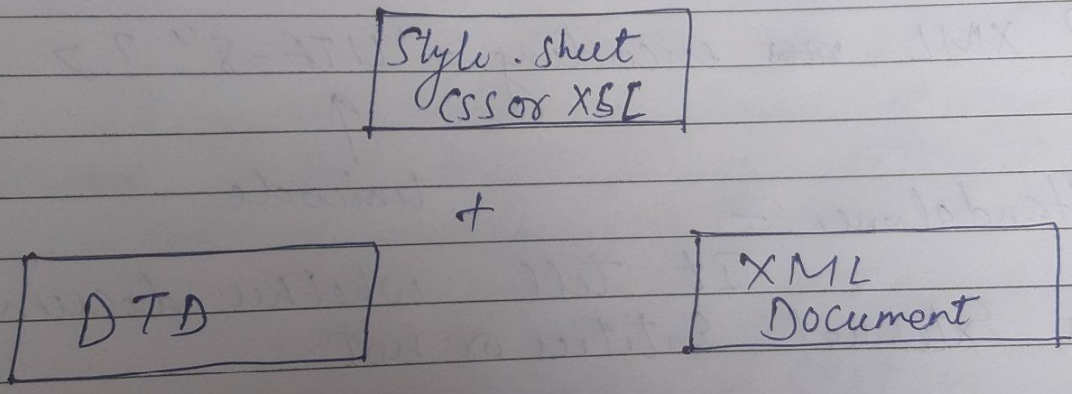
7. Well formed Documents :-

An XML document must follow the syntax & Rules setup for XML, by W3C (World wide web Consortium) in XML 1.0.

8. Valid XML documents :-

An XML document is valid if there is a DTD associated with it.

* Structure of XML document System *



* Structure of XML Document *

- (i) Prolog
- (ii) Data Instance

Prolog

① XML declaration :-
`<?XML?>`

It may contain following attributes.

(i) Version :-

refer to XML version. XML 1.0 is the only version

e.g. `<?XML version="1.0"?>`

(ii) Encoding :- specifies the character set.

e.g. `<?XML xml encoding="UTF-8"?>`
↑
Unicode

(iii) Standalone :-

It tell whether document requires External Entities or not.

9) Standalone value = No means it does not need to refer to any External Entity.

By default value is "yes".

e.g. `<?XML version="1.0" encoding="UTF-8" standalone="No"?>`

② PI (Processing Instruction) :-

It is the instruction that pass information to the application

e.g.

```
<?XML -stylesheet
```

↑
tell that identifies a stylesheet for XML document

It is the instruction that pass information to the application
↓ type

```
type="text/css" href="..."
```

↑
name of CSS file

③ DTD (Document type definition declaration) :-

It is also called grammar of XML document because it tell the application what each element means & how to use it.

e.g.

```
<!DOCTYPE College SYSTEM "College.dtd"
```

↑
tag that start dtd

↑
root element

↑
Instruct the processor to fetch an external document "College.dtd"

④ Comment :-

It begin with

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

↑
end with

⑤ White space :- refers to space, tab, blank lines. It enhances readability of your document.

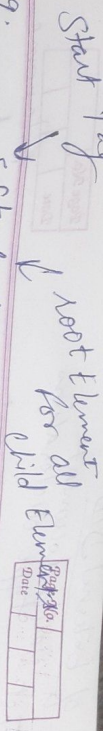
* Data Instance *

① Root Element :- It is parent for all element

② Element Types :-
Start tag :- It start definition of Element
End Tag :- It end definition of Element
Empty tag :- It does not have any end tag.

③ Child tag :- The element nested inside other elements are child elements.

④ Attributes :- Is property setting of the element. & value of attribute must be enclosed in quotation marks.

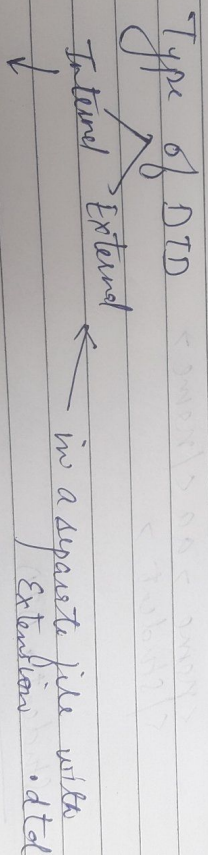


e.g.

```

<?xml version="1.0" >
<Student >
  <name > Ram </name >
  <marks > 10 </marks >
  <address >
    <hno > 11 </hno >
    <pin > 132103 </pin >
  </address >
</Student >
end tag
  
```

* DTD (Document Type Definition)
 defines rule for XML documents. It is optional



Inside XML file

```

<?xml version="1.0" >
<!DOCTYPE Student [
  <!ELEMENT Student (rollno, name) >
  <!ELEMENT rollno (#PCDATA) >
  <!ELEMENT name (#PCDATA) >
</Student >
<rollno > 1 </rollno >
<name > aa </name >
</Student >
  
```

Ex: of External DTD
That start a document declaration

Page No.	
Date	

Ex: ↓ root Element
↓ filename

```
<!DOCTYPE College SYSTEM "College.dtd">
```

That instruct the processor to fetch an external document "College.dtd"

Ex:

```
<? XML version="1.0" ?>
<!DOCTYPE Student SYSTEM "Student.dtd">
<Student >
  <rollno > 1 </rollno >
  <name > aa </name >
</Student >
```

Student a DTD

```
<!ELEMENT Student (rollno, name) >
<!ELEMENT rollno (#PCDATA) >
<!ELEMENT Name (#PCDATA) >
```