XML

SGML

- The ***Standard Generalized Markup Language*** (SGML), is a standard for defining generalized markup languages for documents.
- SGML is based somewhat on earlier generalized markup languages developed at IBM, including General Markup Language (GML).
- Such a specification is itself a document type definition (DTD). SGML is not in itself a document language, but a description of how to specify one.
- SGML is based on the idea that documents have structural and other semantic elements that can be described without reference to how such elements should be displayed.
- HTML is an example of an SGML-based language.
- The Extensible Markup Language (XML) which is a data description language (and a document can be viewed as a collection of data) uses SGML principles.

XML

- XML stands for **Ex**tensible **M**arkup **L**anguage and is a text-based markup language derived from Standard Generalized Markup Language (SGML).
- XML tags identify the data and are used to store and organize the data, rather than specifying how to display it like HTML tags, which are used to display the data.
- XML was designed to be self-descriptive.
- There are three important characteristics of XML that make it useful in a variety of systems and solutions −
  - **XML is extensible** − XML allows you to create your own self-descriptive tags, or language, that suits your application.
  - **XML carries the data, does not present it** − XML allows you to store the data irrespective of how it will be presented.
  - **XML is a public standard** − XML was developed by an organization called the World Wide Web Consortium (W3C) and is available as an open standard.

XML

- Many computer systems contain data in incompatible formats. Exchanging data between incompatible systems (or upgraded systems) is a time-consuming task for web developers. Large amounts of data must be converted, and incompatible data is often lost.
- XML stores data in plain text format. This provides a software- and hardware-independent way of storing, transporting, and sharing data.
- XML separates data from HTML i.e. changes in the underlying data will not require any changes to the HTML.

Comments in XML

- The syntax for writing comments in XML is similar to that of HTML:

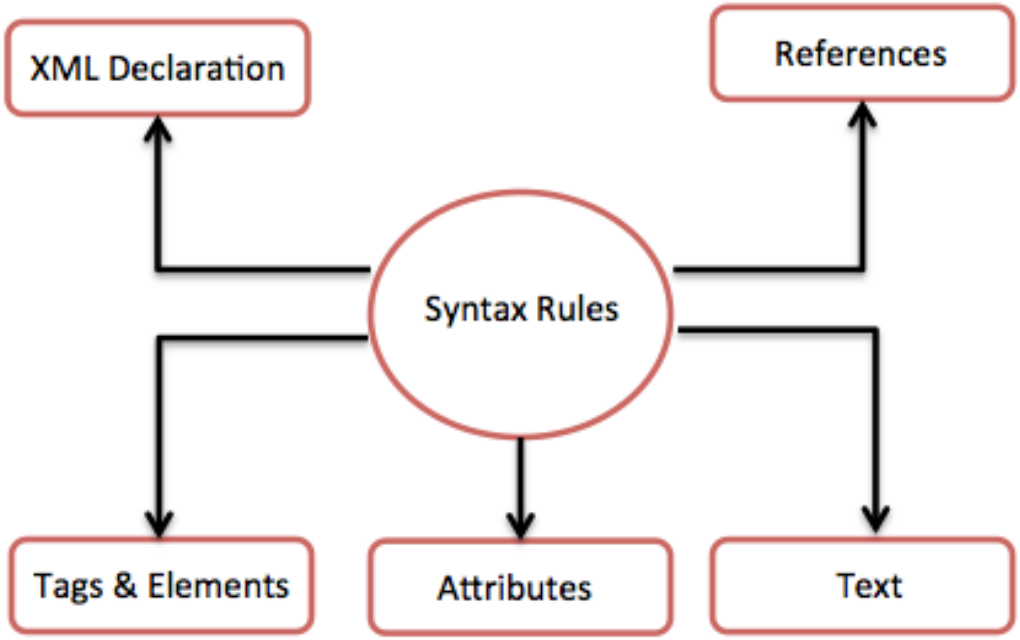
  <!-- This is a comment -->

Features and Advantages of XML

- Separates data from HTML.
- Simplifies data sharing.
- Simplifies data transport.
- Increase data availability.
- Simplifies platform change.

XML Syntax

- The following diagram depicts the syntax rules to write different types of markup and text in an XML document.



XML Declaration

- The XML document can optionally have an XML declaration. It is written as follows −

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

Where *version* is the XML version and *encoding* specifies the character encoding used in the document.

Syntax Rules for XML Declaration

- The XML declaration is case sensitive and must begin with "**<?xml>**" where "**xml**" is written in lower-case.
- If document contains XML declaration, then it strictly needs to be the first statement of the XML document.
- An HTTP protocol can override the value of *encoding* that you put in the XML declaration.

Tags and Elements

- An XML file is structured by several XML-elements, also called XML-nodes or XML-tags. The names of XML-elements are enclosed in triangular brackets < >.
- **Syntax Rules for Tags and Elements**
  - **Element Syntax** − Each XML-element needs to be closed either with start or with end elements as shown below −

    <element>....</element>

    or

    <element/>
  - **Root Element** − An XML document can have only one root element.

    <root> <x>...</x> <y>...</y> </root>
  - **Case Sensitivity** − The names of XML-elements are case-sensitive. That means the name of the start and the end elements need to be exactly in the same case.

XML Attributes

- An **attribute** specifies a single property for the element, using a name/value pair. An XML-element can have one or more attributes.

  For example –

  ```
  <a href = "http://www.google.co.in">Google</a>
  ```

- **Syntax Rules for XML Attributes**
  - Attribute names in XML (unlike HTML) are case sensitive. That is, *HREF*and *href* are considered two different XML attributes.
  - Same attribute cannot have two values in a syntax.
  - Attribute names are defined without quotation marks, whereas attribute values must always appear in quotation marks.

XML References

- References usually allow you to add or include additional text or markup in an XML document. References always begin with the symbol **"&"** which is a reserved character and end with the symbol **";".** XML has two types of references −
  - **Entity References** − An entity reference contains a name between the start and the end delimiters. For example **&** where *amp* is *name*. The *name* refers to a predefined string of text and/or markup.
  - **Character References** − These contain references, such as **A**, contains a hash mark (“#”) followed by a number. The number always refers to the Unicode code of a character. In this case, 65 refers to alphabet "A".

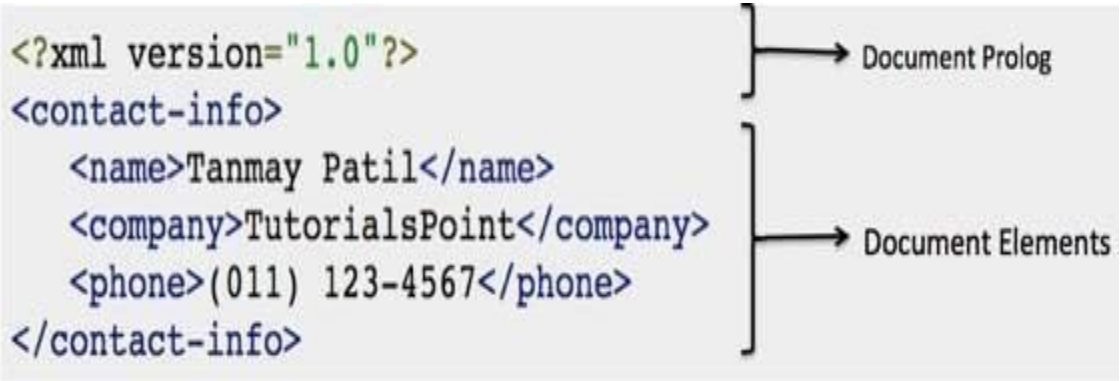
XML Text

- The names of XML-elements and XML-attributes are case-sensitive, which means the name of start and end elements need to be written in the same case. To avoid character encoding problems, all XML files should be saved as Unicode UTF-8 or UTF-16 files.
- Whitespace characters like blanks, tabs and line-breaks between XML-elements and between the XML-attributes will be ignored.
- Some characters are reserved by the XML syntax itself. Hence, they cannot be used directly. To use them, some replacement-entities are used, which are listed below −

Not Allowed Character	Replacement Entity	Character Description
<	<	less than
>	>	greater than
&	&amp;	ampersand
'	'	apostrophe
"	"	quotation mark

XML - Documents

- An XML *document* is a basic unit of XML information composed of elements and other markup in an orderly package. An XML *document* can contains wide variety of data. For example, database of numbers, numbers representing molecular structure or a mathematical equation.
- The following image depicts the parts of XML document.



- **Document Prolog Section**

Document Prolog comes at the top of the document, before the root element. This section contains −

- XML declaration
- Document type declaration

- **Document Elements Section**

Document Elements are the building blocks of XML. These divide the document into a hierarchy of sections, each serving a specific purpose. You can separate a document into multiple sections so that they can be rendered differently, or used by a search engine. The elements can be containers, with a combination of text and other elements.

XML DTD

- The purpose of a DTD is to define the structure of an XML document. It defines the structure with a list of legal elements.
- **Syntax:**

<!DOCTYPE element

[

declaration1

declaration2

]>

- **DTD are of two types**: **Internal** DTD and **External** DTD.

Internal DTD: If the DTD is declared inside the XML file, it must be wrapped inside the <!DOCTYPE> definition

External DTD: If the DTD is declared in an external file, the <!DOCTYPE> definition must contain a reference to the DTD file.

XML document with an internal DTD

- <?xml version="1.0"?>
  <!DOCTYPE note [
  <!ELEMENT note (to,from,heading,body)>
  <!ELEMENT to (#PCDATA)>
  <!ELEMENT from (#PCDATA)>
  <!ELEMENT heading (#PCDATA)>
  <!ELEMENT body (#PCDATA)>
  ]>
  <note>
  <to>Tove</to>
  <from>Jani</from>
  <heading>Reminder</heading>
  <body>Don't forget me this weekend</body>
  </note>

XML document with an internal DTD

The DTD above is interpreted like this:

- !DOCTYPE note defines that the root element of the document is note
- !ELEMENT note defines that the note element must contain the elements: "to, from, heading, body"
- !ELEMENT to defines the to element to be of type "#PCDATA"
- !ELEMENT from defines the from element to be of type "#PCDATA"
- !ELEMENT heading defines the heading element to be of type "#PCDATA"
- !ELEMENT body defines the body element to be of type "#PCDATA"

#PCDATA means parse-able text data.

XML document with a reference to an external DTD

```
<?xml version="1.0"?>
<!DOCTYPE note SYSTEM "note.dtd">
<note>
  <to>Tove</to>
  <from>Jani</from>
  <heading>Reminder</heading>
  <body>Don't forget me this weekend!</body>
</note>
```

And here is the file "note.dtd", which contains the DTD:

```
<!ELEMENT note (to,from,heading,body)>
<!ELEMENT to (#PCDATA)>
<!ELEMENT from (#PCDATA)>
<!ELEMENT heading (#PCDATA)>
<!ELEMENT body (#PCDATA)>
```

No.	HTML	XML
1)	HTML is an abbreviation for HyperText Markup Language.	XML stands for eXtensible Markup Language.
2)	HTML is used to display data and focuses on how data looks.	XML is a software and hardware independent tool used to transport and store data. It focuses on what data is.
3)	HTML is a markup language itself.	XML provides a framework to define markup languages.
4)	HTML is not case sensitive.	XML is case sensitive.
5)	HTML is a presentation language.	XML is neither a presentation language nor a programming language.
6)	HTML has its own predefined tags.	You can define tags according to your need.
7)	In HTML, it is not necessary to use a closing tag.	XML makes it mandatory to use a closing tag.
8)	HTML is static because it is used to display data.	XML is dynamic because it is used to transport data.
9)	HTML does not preserve whitespaces.	XML preserve whitespaces.