Session-1

XML:-

- XML stands for Extensible Markup Language. A markup language uses a set of additional items called markups to create a document of hierarchical structure.
- It is called extensible because it allows the author of the document to define the markup elements by their own.
- It is a markup language to develop documents containing structured information. By saying that, it is meant that xml contains content (i.e. text, images etc.) and along with that it also contains some information or hints about what role that content plays.
- XML is recommended by W3C.

EXAMPLE:

message.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<message>
  <to>Krishna</to>
  <from>venkatesh</from>
  <heading>Reminder</heading>
  <body>Don't forget me this weekend!</body>
</message>
```

Advantages of xml:

- Since xml supports UNICODE, all most all the human readable written languages can be communicated using xml.
- It can be used to render data structure, i.e. records and lists and trees.
- XML is self-documenting, i.e. it contains data and description about the data.
- XML needs another software application called Parser. An XML document is very strict while maintaining a standard. So the parser software for XML is very simple and easy.

- XML is used both on and offline for storing and processing data.
- XML follows international standards.
- XML supports incremental update.
- XML allows validation of the document using XSD or Schematron. These are types of the schema for validating xml documents.
- The tree-like structure of xml is suitable for almost all the types of documents.
- Being platform independent, it has lot of benefits. Like it is not very prone to technological changes.
- Even though changes are made in DTD or schema, it is easier to keep forward or backward compatibility available.
- SGML is its predecessor, which is working from 1986.So a vast experience works behind the implementation of xml.

Differences between XML and HTML:

No.	HTML	XML
1)	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.
2)	HTML is a **markup language** itself.	XML provides a **framework to define markup languages**.
3)	HTML is **not case sensitive**.	XML is **case sensitive**.
4)	HTML is a presentation language.	XML is neither a presentation language nor a programming language.
5)	HTML **has its own predefined tags**.	You **can define tags according to your need**.
6)	In HTML, it is **not necessary to use a closing tag**.	XML **makes it mandatory to use a closing tag**.

7)	HTML is **static** because it is used to display data.	XML is **dynamic** because it is used to transport data.
8)	HTML **does not preserve whitespaces**.	XML **preserve whitespaces**.

XML Features

- Excellent for handling data with a complex structure or atypical data.
- Data described using markup language.
- Text data description.
- Human- and computer-friendly format.
- Handles data in a tree structure having one-and only one-root element.
- Excellent for long-term data **storage** and data reusability.

1. XML focuses on data rather than how it looks

One of the reason, XML is popular because it focuses on data rather than data presentation. The other markup language such as HTML is used for data presentation. This separates the data and its presentation part and gives us the freedom to present the data, the way we want, once we receive it using XML.

Two or more systems can receive the same data from a same XML and present it in a different way using other markup language such as HTML.

2. Easy and efficient data sharing

Since XML is **software and hardware independent**, it is easier to share data between different systems with different hardware and software configuration. Any system with any programming language can read and process a XML document.

3. Compatibility with other markup language HTML

It is so much easier to read the data from XML and display it on an GUI(graphical user interface) using HTML markup language.

When the data changes over time, we need not to make any changes in the HTML.

4. Supports platform transition

The main reason why changing to new systems and platform is challenging, because it involves the headache of data conversion between incompatible formats which often results in data loss. XML simplifies this process as the data is transported on new upgraded systems without any data loss.

5. Allows XML validation

A XML document can be validated using DTD or XML schema. This ensures that the XML document is syntactically correct and avoids any issues that may arise due to the incorrect XML.

6. Adapts technology advancements

The reason why XML is popular and being used from a very long time is because, it can adapt to the new technologies because of its platform-independent nature.

7. XML supports Unicode

XML supports Unicode that allows it to communicate almost any information in any written human language.

XML ATTRIBUTES: -

An element can have multiple unique attributes. Attribute gives more information about XML elements. To be more precise, they define properties of elements. An XML attribute is always a name-value pair.

Syntax

An XML attribute has the following syntax −

```
<element-nameattribute1attribute2>
....content..
< /element-name>
```

where *attribute1* and *attribute2* has the following form −

name = "value"

value has to be in double (" ") or single (' ') quotes. Here, *attribute1* and *attribute2* are unique attribute labels.

Attributes are used to add a unique label to an element, place the label in a category, add a Boolean flag, or otherwise associate it with some string of data. Following example demonstrates the use of attributes −

```
<?xml version ="1.0" encoding ="UTF-8"?>
<! DOCTYPE garden [
<!ELEMENT garden (plants)*>
<!ELEMENT plants (#PCDATA)>
<!ATTLIST plants category CDATA #REQUIRED>
]>

<garden>
<plantscategory="flowers"/>
<plantscategory="shrubs">
</plants>
</garden>
```

Element Attribute Rules

Following are the rules that need to be followed for attributes −

- An attribute name must not appear more than once in the same start-tag or empty-element tag.
- An attribute must be declared in the Document Type Definition (DTD) using an Attribute-List Declaration.
- Attribute values must not contain direct or indirect entity references to external entities.
- The replacement text of any entity referred to directly or indirectly in an attribute value must not contain a less than sign (**<**)