XML

- XML stands for **Extensible Markup Language**.
- It is used to store and transport data in a structured format.
- XML is both **human-readable** and **machine-readable**.
- It is widely used in web development, configuration files, and data exchange.

Features of XML

•**Self-descriptive:** XML tags define data and its structure.

```
<book>
        <title>XML for Beginners</title>
        <author>John Doe</author>
</book>
```

•**Platform-independent:** Works on any system or software.

XML works on all operating systems and programming languages.

Example: A web application can send XML data to a mobile app, and both can understand it.

•**Supports hierarchy:** Data is organized in a tree structure.

XML organizes data in a tree-like structure with a **root element** and **nested child elements**.

•**Extensible:** Users can define their own tags.

•**Interoperability:** Easily integrates with different applications.

XML helps different applications communicate with each other. For example, an **XML-based API** can be used by multiple programming languages.

Basic Structure of an XML Document

```
<?xml version="1.0" encoding="UTF-8"?>
<students>
        <student>
                <name>John Doe</name>
                 <age>20</age>
                <department>Computer Science</department>
        </student>
</students>
```

- The **prolog** (<?xml version="1.0"?>) defines XML version and encoding.
- The **root element** (<students>) contains all the data.
- **Nested elements** (<student>, <name>, <age>) define the structure.

XML vs HTML

Feature	XML	HTML
Purpose	Stores and transports data	Displays data
Tag Definition	User-defined	Predefined (like <p>, <h1>)
Syntax Rules	Strict	Flexible
Nesting Rules	Must be well-formed	Can be incorrect
Usage	Data exchange (APIs, databases)	Web pages

XML Rules

To ensure that XML is well-formed, it must follow these rules:

1. Every XML document must have a single root element

✅ Correct:

```
<library>
    <book>XML Basics</book>
</library>
```

❌ Incorrect:

```
<book>XML Basics</book>
<author>John Doe</author>
```

(Multiple root elements are not allowed.)

XML Rules

Tags must be properly nested

✅ Correct:

```
<student>
        <name>John</name>
 </student>
```

❌ Incorrect:

```
<student>
        <name>John</student>
 </name>
```

(Tags must close in the correct order.)

XML Rules

Tags are case-sensitive

✅ Correct:

<Student>John</Student>

❌ Incorrect:

<Student>John</student>

(Student and student are treated as different tags.)

XML Rules

Attributes must be enclosed in quotes

✅ Correct:

<student name="John" age="20"/>

❌ Incorrect:

<student name=John age=20/>

(Attributes must be inside quotes.)

Uses of XML

XML is widely used in various domains:

a) Web Services (APIs)
Many web services use XML for data exchange, such as:

- **SOAP (Simple Object Access Protocol)**
- **RSS Feeds** (Used in news websites)
- **Web APIs** (Many APIs return data in XML format)

b) Configuration Files
Many software applications use XML to store settings, like:

- **Microsoft Office** (.docx, .xlsx use XML internally)
- **Android Manifest File** (Android apps use XML to define permissions and UI layout)

XML Processing (Parsing XML)

There are two main ways to process XML:

a) DOM (Document Object Model) Parsing

•Loads the entire XML document into memory.

•Used in languages like JavaScript, Python, Java.

Example (JavaScript):

```
let xmlDoc = new DOMParser().parseFromString(xmlString, "text/xml");
console.log(xmlDoc.getElementsByTagName("student")[0].textContent);
```

SOAP

- SOAP stands for **S**imple **O**bject **A**ccess **P**rotocol
- SOAP is an application communication protocol
- SOAP is a format for sending and receiving messages
- SOAP is platform independent
- SOAP is based on XML


ST. ANDREWS INSTITUTE OF TECHNOLOGY & MANAGEMENT


NS3EDU.
Learn Today
Earn Tomorrow

THANK YOU