XML(EXTENSIBLE MARKUP LANGUAGE)

BY : PRANJAL JATAV (IT-64)
SONALI BAGHEL(IT-86)

WHAT IS XML ?

XML, or eXtensible Markup Language, is a versatile markup language designed for storing and transporting data.

Developed by the W3C (World Wide Web Consortium).

First introduced in the late 1990s, XML has become a fundamental technology for data exchange and representation.

FEATURES OF XML

- Extensible and Human Readable.
- Overall Simplicity data availability, platform modifications, data transit, and sharing are all made simpler by XML. Without losing data, XML makes it simpler to upgrade or extend to new operating systems, apps, or browsers.
- Separates Data from HTML.
- Allows XML Validation.
- XML Supports Unicode.
- Used to Create New Languages.

BENEFITS OF XML

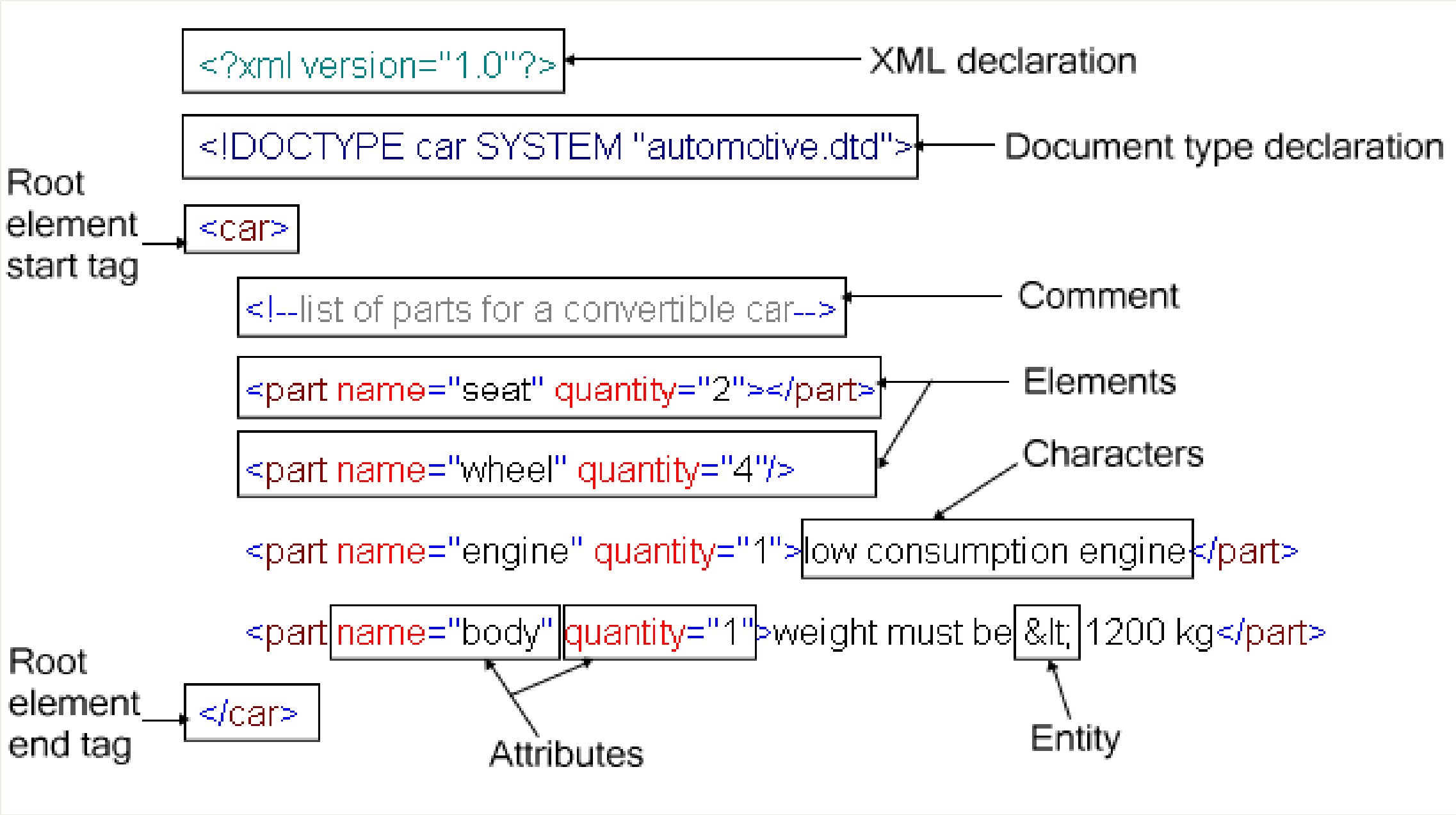
- XML is platform independent and programming language independent.
- XML supports unicode. Unicode is an international encoding standard.
- The data stored and transported using XML can be changed at any point of time without affecting the data presentation.
- XML allows validation using DTD and Schema. This validation ensures that the XML document is free from any syntax error.
- XML simplifies data sharing between various systems because of its platform independent nature. XML data doesn't require any conversion when transferred between different systems.
- (XML doesn't support array.)

XML Vs HTML

XML (Extensible Markup Language)	**HTML** (Hypertext Markup Language)
It stores and transports data.	It displays data.
It uses user-defined tags.	It uses predefined tags.
It contains structural data.	It doesn't contain any structural data.
It can distinguish uppercase and lowercase letters (case sensitive).	It can't distinguish uppercase and lowercase letters (case insensitive).
It maintains spacing, tabs, newlines, and any other whitespace formatting.	It doesn't maintain whitespace.
It needs to have an end-tag.	It doesn't need an end-tag.
It needs structure or nesting.	It doesn't need structure.

BUILDING BLOCKS OF XML

- *Elements:* Building blocks of data, identified by tags (e.g., <book>, </book>)
- *Attributes:* Provide additional details about elements (e.g., <book title="Pride and Prejudice">)
- *Text Content:* Data held within element tags
- *Comments:* Explain code without affecting data (like helpful notes).


<?xml version="1.0"?>
XML declaration
<!DOCTYPE car SYSTEM "automotive.dtd">
Document type declaration
Root element start tag
<car>
<!--list of parts for a convertible car-->
Comment
<part name="seat" quantity="2"></part>
Elements
<part name="wheel" quantity="4"/>
Characters
<part name="engine" quantity="1">low consumption engine</part>
<part name="body" quantity="1">weight must be < 1200 kg</part>
Root element end tag
</car>
Attributes
Entity

```xml
<?xml version="1.0" encoding="UTF-8"?>
<catalog>
  <book id="bk101">
    <title>XML for Beginners</title>
    <author>John Doe</author>
    <genre>Computer</genre>
    <price>29.99</price>
    <publish_date>2023-11-22</publish_date>
    <description>A comprehensive guide to learning XML basics.</description>
  </book>
  <book id="bk102">
    <title>Advanced XML Applications</title>
    <author>Jane Smith</author>
    <genre>Computer</genre>
    <price>39.99</price>
    <publish_date>2024-01-15</publish_date>
    <description>Explores advanced XML techniques for data management and web development.</descrip
  </book>
</catalog>
```

Key elements:

- <?xml version="1.0" encoding="UTF-8"?>: The XML declaration, stating the version and character encoding.
- <catalog>: The root element, containing all book information.
- <book>: Represents a single book entry, with attributes like id, title, author, and more.
- <title>, <author>, <genre>, etc.: Child elements providing specific book details.

Key points:

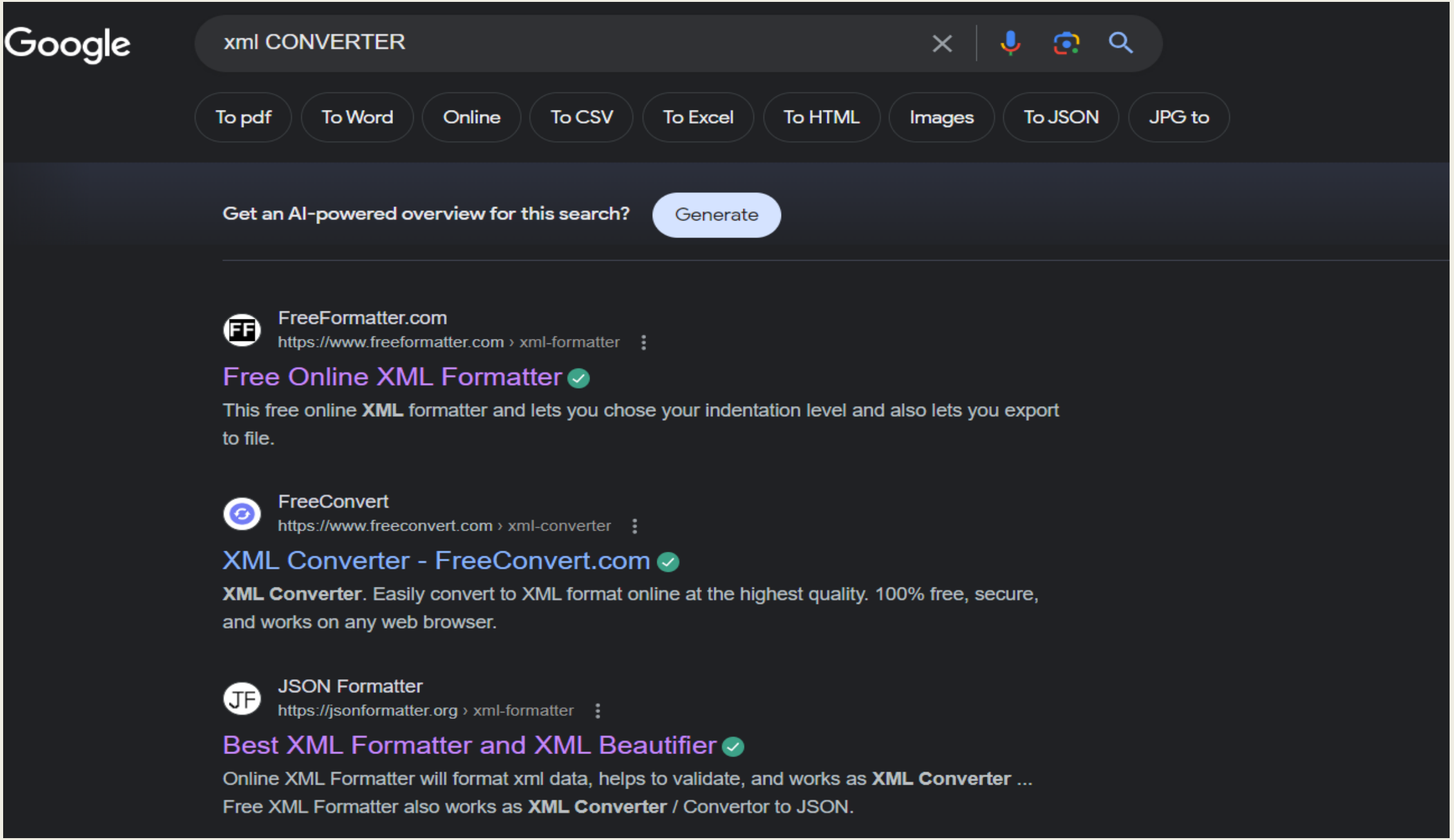
- XML uses tags to define elements and structure data.
- Tags are enclosed in angle brackets (< >).
- Elements can have attributes (e.g., id="bk101").
- Elements can nest within other elements, creating hierarchical structures.
- XML is case-sensitive (e.g., <book> is different from <Book>).

To work with this code:

1. Save it as a .xml file (e.g., "book_catalog.xml").
2. Open it in a text editor or an XML editor for viewing and editing.
3. Use programming languages or tools to process and manipulate the XML data.

Creating Simple XML Documents:

- Text editors: Notepad++ or Sublime Text are great free options for writing XML code.
- Online tools: Websites like XMLGrid offer a user-friendly interface to create and edit XML documents: https://xmlgrid.net/demo/


Google
xml CONVERTER
To pdf
To Word
Online
To CSV
To Excel
To HTML
Images
To JSON
JPG to
Get an AI-powered overview for this search?
Generate
FreeFormatter.com
https://www.freeformatter.com › xml-formatter
Free Online XML Formatter
This free online XML formatter and lets you chose your indentation level and also lets you export to file.
FreeConvert
https://www.freeconvert.com › xml-converter
XML Converter - FreeConvert.com
XML Converter. Easily convert to XML format online at the highest quality. 100% free, secure, and works on any web browser.
JSON Formatter
https://jsonformatter.org › xml-formatter
Best XML Formatter and XML Beautifier
Online XML Formatter will format xml data, helps to validate, and works as XML Converter ...
Free XML Formatter also works as XML Converter / Convertor to JSON.

BASIC SYNTAX

XML uses tags to define elements, which can contain text and other elements.

Example:

<SGSITS>

<IT>

<STUDENT>

<NAME> PRANJAL JATAV </NAME>

</STUDENT>

</IT>

</SGSITS>

XML Formatter

Formats a XML string/file with your desired indentation level. The formatting rules are not configurable but it uses a per-element indentation pattern giving the best readability.

Option 1: Copy-paste your XML here

```
<SGSITS>
<IT>
<STUDENT>
 <NAME> PRANJAL JATAV </NAME>
</STUDENT>
```

Option 2: Or upload your XML file

Choose File | No file chosen

File encoding: UTF-8

Text mode: Trim (left and right trimming)

Indentation level: 3 spaces per indent level

Format XML | Format XML to new window

-Formatted XML-

Copy Save

```
<?xml version="1.0" encoding="UTF-8"?>
<SGSITS>
   <IT>
      <STUDENT>
         <NAME>PRANJAL JATAV</NAME>
      </STUDENT>
   </IT>
</SGSITS>
```

XML Parsers:

- These are software tools that read and process XML documents. They break down the structure of the document, identifying elements, attributes, and text content
- Some popular XML parsers include libxml2, SAX, DOM, and StAX.
- the choice depends on your specific needs and programming language.
- Parsers don't typically modify the XML document; they simply prepare it for further processing.

XSLT with XML:

- XSLT (eXtensible Stylesheet Language Transformations) is a language used to transform XML documents into other formats like HTML, text, or even another XML document.
- With XSLT, you can:

  Extract, filter, and rearrange data from the XML document.

  Apply formatting and styling to the output based on the content.

  Perform calculations and conditional logic based on the XML data.

  Generate completely new documents with different structures.

USE IN REAL WORLD

XML is used in a surprising number of things in our daily lives, even if we don't always see it directly. Here are some real-world examples of how XML is used:

- Sharing news and updates
- Shopping online
- Document management
- Data exchange between businesses
- Behind the scenes in websites
- Mobile apps

These are just a few examples of how XML is used in real life. It's a versatile tool that can be used to manage and share data in a variety of ways.

CONCLUSION:

So, embrace the XML revolution! It's not just a markup language, it's a data superpower. Integrate it, explore it, and watch your data dance with newfound elegance and clarity.

Remember: XML is more than just tech, it's a mindset. It's about taking control of your data, giving it structure, and making it work for you. So, unleash your inner data maestro, and let XML lead the way to a more organized and efficient digital world!

THANK YOU