Introduction to XML

XML stands for eXtensible Markup Language.

XML was designed to store and transport data.

XML was designed to be both human- and machine-readable.

XML is a software- and hardware-independent tool for storing and transporting data.

XML, or Extensible Markup Language, is a text-based data format that is widely used to represent structured information in a human-readable and machine-readable way.

XML is not a programming language but a set of rules for encoding documents in a format that can be easily processed by computers. Here are some key characteristics and uses of XML:

XML uses tags to define elements and their structure within a document. Tags are enclosed in angle brackets (< and >) and come in two primary forms: start tags and end tags. Start tags define the beginning of an element, and end tags indicate the end of an element. Here's an overview of XML tags:

To demonstrate how to parse and use XML in PHP, here's an example of an XML document and a PHP script that parses the XML to generate an output:

**XML Data (data.xml):**

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

<books>

<book>

<title>XML Guide</title>

<author>John Doe</author>

<genre>Technical</genre>

<published>2021</published>

</book>

<book>

<title>Harry Potter and the Sorcerer's Stone</title>

<author>J.K. Rowling</author>

<genre>Fiction</genre>

<published>1997</published>

</book>

</books>

**PHP Script (parse\_xml.php):**

<?php

// Load and parse the XML file

$xml = simplexml\_load\_file('data.xml');

// Check if the XML was loaded successfully

if ($xml === false) {

die('Failed to load XML data.');

}

// Output the information

echo '<h1>Book Information</h1>';

foreach ($xml->book as $book) {

echo '<div class="book">';

echo '<h2>Title: ' . $book->title . '</h2>';

echo '<p>Author: ' . $book->author . '</p>';

echo '<p>Genre: ' . $book->genre . '</p>';

echo '<p>Published: ' . $book->published . '</p>';

echo '</div>';

}

This PHP script loads and parses the XML file **data.xml** using SimpleXML. It then iterates through the **<book>** elements and displays the information about each book.

**Output:**

Book Information

Title: XML Guide

Author: John Doe

Genre: Technical

Published: 2021

Title: Harry Potter and the Sorcerer's Stone

Author: J.K. Rowling

Genre: Fiction

Published: 1997

What is JSON?

JSON stands for **J**ava**S**cript **O**bject **N**otation

JSON is a **text format** for storing and transporting data

JSON is "self-describing" and easy to understand

JSON is used to send data between computers

JSON is language independent \*

Why Use JSON?

The JSON format is syntactically similar to the code for creating JavaScript objects. Because of this, a JavaScript program can easily convert JSON data into JavaScript objects.

Since the format is text only, JSON data can easily be sent between computers, and used by any programming language.

JavaScript has a built in function for converting JSON strings into JavaScript objects:

JSON.parse()

JavaScript also has a built in function for converting an object into a JSON string:

JSON.stringify()

<!DOCTYPE html>

<html>

<body>

<h2>Access a JavaScript object</h2>

<p id="demo"></p>

<script>

const myObj = {name:"John", age:30, city:"New York"};

document.getElementById("demo").innerHTML = myObj.name;

</script>

</body>

</html>

JSON vs XML

Both JSON and XML can be used to receive data from a web server.

The following JSON and XML examples both define an employees object, with an array of 3 employees:

### **JSON Example**

{"employees":[  
  { "firstName":"John", "lastName":"Doe" },  
  { "firstName":"Anna", "lastName":"Smith" },  
  { "firstName":"Peter", "lastName":"Jones" }  
]}

### **XML Example**

<employees>  
  <employee>  
    <firstName>John</firstName> <lastName>Doe</lastName>  
  </employee>  
  <employee>  
    <firstName>Anna</firstName> <lastName>Smith</lastName>  
  </employee>  
  <employee>  
    <firstName>Peter</firstName> <lastName>Jones</lastName>  
  </employee>  
</employees>

## **JSON is Like XML Because**

* Both JSON and XML are "self describing" (human readable)
* Both JSON and XML are hierarchical (values within values)
* Both JSON and XML can be parsed and used by lots of programming languages
* Both JSON and XML can be fetched with an XMLHttpRequest

## **JSON is Unlike XML Because**

* JSON doesn't use end tag
* JSON is shorter
* JSON is quicker to read and write
* JSON can use arrays

The biggest difference is:

 XML has to be parsed with an XML parser. JSON can be parsed by a standard JavaScript function.