Difference between Json and xml

| **JSON** | **XML** |
| --- | --- |
| It is [JavaScript Object Notation](https://www.geeksforgeeks.org/javascript-json/) | It is [Extensible markup language](https://www.geeksforgeeks.org/xml-basics/) |
| It is based on JavaScript language. | It is derived from [SGML](https://www.geeksforgeeks.org/what-is-sgml/). |
| It is a way of representing objects. | It is a markup language and uses tag structure to represent data items. |
| It does not provides any support for namespaces. | It supports [namespaces](https://www.geeksforgeeks.org/javascript-namespace/). |
| It supports array. | It doesn’t supports [array](https://www.geeksforgeeks.org/array-data-structure/). |
| Its files are very easy to read as compared to XML. | Its documents are comparatively difficult to read and interpret. |
| It doesn’t use end tag. | It has start and end tags. |
| It is less secured. | It is more secured than JSON. |
| It doesn’t supports comments. | It supports comments. |
| It supports only [UTF-8 encoding](https://www.geeksforgeeks.org/understanding-character-encoding/). | It supports various encoding. |

Difference between authorization and authentication

| **Authentication** | **Authorization** |
| --- | --- |
| In the [authentication](https://www.geeksforgeeks.org/authentication-in-computer-network/) process, the identity of users are checked for providing the access to the system. | While in [authorization](https://www.geeksforgeeks.org/what-is-aaa-authentication-authorization-and-accounting/) process, a the person’s or user’s authorities are checked for accessing the resources. |
| In the authentication process, users or persons are verified. | While in this process, users or persons are validated. |
| It is done before the authorization process. | While this process is done after the authentication process. |
| It needs usually the user’s login details. | While it needs the user’s privilege or security levels. |
| Authentication determines whether the person is user or not. | While it determines **What permission does the user have?** |
| Generally, transmit information through an ID Token. | Generally, transmit information through an Access Token. |
| The OpenID Connect (OIDC) protocol is an authentication protocol that is generally in charge of user authentication process. | The OAuth 2.0 protocol governs the overall system of user authorization process. |
| Popular Authentication Techniques-   * Password-Based Authentication * Passwordless Authentication * 2FA/MFA (Two-Factor Authentication / Multi-Factor Authentication) * [Single sign-on (SSO)](https://www.geeksforgeeks.org/introduction-of-single-sign-on-sso/) * Social authentication | Popular  Authorization Techniques-   * Role-Based Access Controls (RBAC) * [JSON web token (JWT) Authorization](https://www.geeksforgeeks.org/json-web-token-jwt/) * SAML Authorization * OpenID Authorization * OAuth 2.0 Authorization |
| The authentication credentials can be changed in part as and when required by the user. | The authorization permissions cannot be changed by user as these are granted by the owner of the system and only he/she has the access to change it. |
| The user authentication is visible at user end. | The user authorization is not visible at the user end. |
| The user authentication is identified with username, password, face recognition, retina scan, fingerprints, etc. | The user authorization is carried out through the access rights to resources by using roles that have been pre-defined. |
| **Example**: Employees in a company are required to authenticate through the network before accessing their company email. | **Example:** After an employee successfully authenticates, the system determines what information the employees are allowed to access. |

Create a HTML page to upload with a file

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>File Upload Form</title>

</head>

<body>

<h1>Upload a File</h1>

<form action="/upload" method="post" enctype="multipart/form-data">

<label for="file">Choose a file:</label>

<input type="file" id="file" name="file">

<br><br>

<input type="submit" value="Upload">

</form>

</body>

</html>

Create HTML page with audio file

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Audio File Example</title>

</head>

<body>

<h1>Listen to this Audio</h1>

<audio controls>

<source src="path\_to\_your\_audio\_file.mp3" type="audio/mpeg">

Your browser does not support the audio element.

</audio>

</body>

</html>

Create a HTML page with video file

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Video File Example</title>

</head>

<body>

<h1>Watch this Video</h1>

<video width="640" height="480" controls>

<source src="path\_to\_your\_video\_file.mp4" type="video/mp4">

Your browser does not support the video tag.

</video>

</body>

</html>

Create a login screen

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Login Screen</title>

<style>

body {

font-family: Arial, sans-serif;

background-color: #f0f0f0;

display: flex;

justify-content: center;

align-items: center;

height: 100vh;

margin: 0;

}

.login-container {

background-color: #fff;

padding: 20px;

border-radius: 8px;

box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);

width: 300px;

}

.login-container h2 {

margin-top: 0;

text-align: center;

}

.login-container label {

display: block;

margin-bottom: 8px;

font-weight: bold;

}

.login-container input[type="text"],

.login-container input[type="password"] {

width: 100%;

padding: 10px;

margin-bottom: 10px;

border: 1px solid #ccc;

border-radius: 4px;

box-sizing: border-box;

}

.login-container input[type="submit"] {

width: 100%;

padding: 10px;

background-color: #4CAF50;

border: none;

border-radius: 4px;

color: #fff;

font-size: 16px;

cursor: pointer;

}

.login-container input[type="submit"]:hover {

background-color: #45a049;

}

</style>

</head>

<body>

<div class="login-container">

<h2>Login</h2>

<form action="/login" method="post">

<label for="username">Username</label>

<input type="text" id="username" name="username" required>

<label for="password">Password</label>

<input type="password" id="password" name="password" required>

<input type="submit" value="Login">

</form>

</div>

</body>

</html>

Create 3xml and json files for department,year,student

{

"departments": [

{

"id": 1,

"name": "Computer Science"

},

{

"id": 2,

"name": "Mathematics"

},

{

"id": 3,

"name": "Physics"

}

]

}

Create a HTML page with google map

<!DOCTYPE html>

<html>

<head>

<title>Google Maps Example</title>

<style>

/\* Set the size of the div element that contains the map \*/

#map {

height: 400px; /\* The height is 400 pixels \*/

width: 100%; /\* The width is the width of the web page \*/

}

</style>

</head>

<body>

<h3>My Google Map</h3>

<!-- The div element for the map -->

<div id="map"></div>

<script>

// Initialize and add the map

function initMap() {

// The location to center the map (example: New York City)

const location = { lat: 40.7128, lng: -74.0060 };

// The map, centered at the location

const map = new google.maps.Map(document.getElementById("map"), {

zoom: 10,

center: location,

});

// The marker, positioned at the location

const marker = new google.maps.Marker({

position: location,

map: map,

});

}

</script>

<!-- Load the Google Maps API script with your API key -->

<script async defer

src="https://maps.googleapis.com/maps/api/js?key=YOUR\_API\_KEY&callback=initMap">

</script>

</body>

</html>