**Tutorial 2 Use of JSON in web development**

**2.1 JSON?**

JSON (JavaScript Object Notation) is a lightweight, text-based format for storing and transporting data. It is easy for humans to read and write, and easy for machines to parse and generate.

**2.2 JSON Structure**

JSON is built on two structures:

* **Object** – A collection of key/value pairs, enclosed in {}

Example:

|  |
| --- |
| {  "name": "Sneha",  "age": 22,  "city": "Pune"  } |

* **Array** – An ordered list of values, enclosed in []

Example:

|  |
| --- |
| [  "Apple",  "Banana",  "Mango"  ] |

**2.3 Data Types Supported in JSON**

| **JSON Type** | **Description** | **Example** |
| --- | --- | --- |
| String | Sequence of characters | "hello" |
| Number | Integer or floating-point | 42, 3.14 |
| Object | Unordered collection of key/value | { "key": "value" } |
| Array | Ordered list of values | [1, 2, 3] |
| Boolean | true or false | true |
| Null | Empty or non-existent value | null |

**2.4 Uses:**

* Exchange data between **client** (browser) and **server**.
* Store configuration, settings, or local app data.
* APIs (RESTful services) use JSON to format responses and requests.
* Used in AJAX calls with JavaScript.

**2.5 JSON with JavaScript**

**Convert JS Object → JSON String:**

|  |
| --- |
| const student = { name: "Amit", age: 20 };  const jsonData = JSON.stringify(student); |

**Convert JSON String → JS Object:**

|  |
| --- |
| const data = '{"name":"Amit","age":20}';  const obj = JSON.parse(data); |

**2.6 JSON vs XML**

| **Feature** | **JSON** | **XML** |
| --- | --- | --- |
| Syntax | Lightweight, concise | Verbose |
| Readability | Easier for humans | More complex |
| Data Format | Objects and arrays | Tags and attributes |
| Speed | Faster parsing | Slower due to complexity |
| Usage | Web APIs, JS apps | Legacy systems, SOAP, etc. |

**2.7 Real-World Applications**

* Front-end JavaScript apps fetching JSON data via fetch() or XMLHttpRequest
* Server responses in Node.js, Python, PHP, etc.
* Local storage of structured data using localStorage or sessionStorage

**2.8 Code**

|  |
| --- |
| **Index.html**  <!DOCTYPE html>  <html lang="en">  <head>  <meta charset="UTF-8">  <title>Send JSON Data</title>  </head>  <body>  <h2>Enter User Details</h2>  <form id="userForm">  Name: <input type="text" id="name" required><br><br>  Age: <input type="number" id="age" required><br><br>  <button type="submit">Send</button>  </form>  <script>  document.getElementById('userForm').addEventListener('submit', function (e) {  e.preventDefault();  const name = document.getElementById('name').value;  const age = document.getElementById('age').value;  const userData = { name, age };  localStorage.setItem('user', JSON.stringify(userData)); // Store as JSON  window.location.href = 'display.html'; // Navigate to display page  });  </script>  </body>  </html> |
| **Display.html**  <!DOCTYPE html>  <html lang="en">  <head>  <meta charset="UTF-8">  <title>Display JSON Data</title>  </head>  <body>  <h2>User Details</h2>  <div id="userData"></div>  <script>  const user = JSON.parse(localStorage.getItem('user')); // Read from localStorage  const container = document.getElementById('userData');  if (user) {  container.innerHTML = `<p><strong>Name:</strong> ${user.name}</p>  <p><strong>Age:</strong> ${user.age}</p>`;  } else {  container.textContent = 'No user data found.';  }  </script>  </body>  </html> |
| **OutPut** |

**2.9 Description**

|  |
| --- |
| *document.getElementById('userForm').addEventListener('submit', function (e) {*  Adds an event listener to handle form submission using JavaScript. |
| *e.preventDefault();*  Prevents the default page reload on form submit. |
| *const name = document.getElementById('name').value;*  const age = document.getElementById('age').value;  Retrieves values entered by the user into input fields. |
| *const userData = { name, age };*  Combines name and age into a JavaScript object. |
| *localStorage.setItem('user', JSON.stringify(userData));*  Converts the object to a JSON string and stores it in localStorage with key 'user'. |
| *window.location.href = 'display.html';*  Redirects the browser to display.html to show the stored data. |

|  |
| --- |
| *const user = JSON.parse(localStorage.getItem('user'));*  Reads the 'user' item from localStorage.  Parses it back into a JavaScript object using JSON.parse(). |
| *<div id="userData"></div>*  Placeholder to show the user's data on the page. |
| *if (user) {*  *container.innerHTML = `<p><strong>Name:</strong> ${user.name}</p>*  *<p><strong>Age:</strong> ${user.age}</p>`;*  *}*  If user data exists, insert it into the #userData div as formatted HTML. |
| *else {*  *container.textContent = 'No user data found.';*  *}*  If nothing is stored, display a fallback message. |