

Session 10

Javascript Cookies

Cookies are small text files stored on the user's client-side computer that allow web applications to "remember" information across page reloads or future visits. While modern web development often uses `localStorage` or `sessionStorage` for client-side persistence, cookies remain essential for session tracking, authentication tokens, and server-side state management because they are automatically sent with every HTTP request to the matching domain. Session 10 covers the complete lifecycle of cookies using the native `document.cookie` API, teaching you how to create, read, parse, modify, and delete them programmatically.

Objectives

× Do not copy. Read for understanding and the viva

- Understand how cookies work and their role in web state management
- Create, read, update, and delete cookies using `document.cookie`
- Parse the raw cookie string to extract specific name=value pairs
- Implement practical exercises for user preference storage and session tracking

What is a JavaScript Cookie?

- A cookie is a small piece of data stored as a name=value pair in the browser.
- Stored as plain text files on the client machine.
- Automatically included in Cookie HTTP headers for subsequent requests to the same domain/path.

Cookie Structure & Attributes

```
1 document.cookie = "username=JohnDoe; expires=Fri, 31 Dec 2026 23:59:59 GMT";
```

Attribute	Purpose
<code>expires</code> / <code>max-age</code>	Defines lifespan. If omitted, cookie is a session cookie (deleted when browser closes)
<code>path</code>	Scope of the cookie. <code>path=</code> makes it available site-wide
<code>domain</code>	Host scope. Default is current domain
<code>secure</code>	Only sent over HTTPS
<code>samesite</code>	CSRF protection (<code>Strict</code> , <code>Lax</code> , <code>None</code>)

The `document.cookie` API

Operation	Syntax	Behavior
Create/Update	<code>document.cookie = "name=value; path=/"</code>	Appends or overwrites existing cookie with same name
Read All	<code>let all = document.cookie</code>	Returns a single string: <code>"id=1; theme=dark; user=John"</code>
Delete	<code>document.cookie = "name=; expires=Thu, 01 Jan 1970 00:00:00 UTC; path=/"</code>	Forces expiration to a past date

Important: `document.cookie` does not return an object or array. You must manually parse the string to extract specific values.

Parsing the Cookie String

```
1 function getCookie(name) {
2   const nameEQ = name + "=";
3   const cookies = document.cookie.split(';');
4   for(let i = 0; i < cookies.length; i++) {
5     let c = cookies[i].trim();
6     if (c.indexOf(nameEQ) === 0) return c.substring(nameEQ.length);
7   }
8   return null; // Not found
9 }
```

JS

Best Practices & Limitations

Limitation	Workaround / Best Practice
Size limit (~4KB per cookie)	Store only essential identifiers; use localStorage for bulk data
Sent with every request	Minimize cookie count & size for performance
Accessible via JS (document.cookie)	Never store passwords/sensitive data; use HttpOnly (server-set only) for security-critical cookies
Parsing overhead	Use helper functions or modern cookie libraries in production

Quick Guide

✗ Do not copy. Read for understanding and the viva

JS

```
1 // ✅ Set Cookie
2 function setCookie(name, value, days) {
3     const d = new Date(); d.setTime(d.getTime() + (days * 24 * 60 * 60 * 1000))
4     document.cookie = `${name}=${value};expires=${d.toUTCString()};path=/`;
5 }
6
7 // 🔍 Get Cookie
8 function getCookie(name) {
9     const match = document.cookie.match(new RegExp('(^| )' + name + '=(^[^;]+)')
10     return match ? decodeURIComponent(match[2]) : null;
11 }
12
13 // 🗑 Delete Cookie
14 function deleteCookie(name) {
15     document.cookie = `${name}=;expires=Thu, 01 Jan 1970 00:00:00 UTC;path=/`;
16 }
```

Question 1

Problem Statement

■ Write in lab record

Create an HTML web page, as shown below. The cookie will be set on pressing setCookie button and the stored cookie value will be displayed on pressing getCookie button.

Note: Check lab manual for images

Demo

✖ Do not copy. Read for understanding and the viva

Set & Get Cookie

[Set Cookie](#)[Get Cookie](#)

Code

■ Write in lab record

time.html

```
1 <!doctype html>
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8" />
5     <title>Ex1: Set & Get Cookie</title>
6     <style>
7       body {
8         font-family: sans-serif;
9         padding: 20px;
10        max-width: 400px;
11      }
12      input,
13      button {
14        padding: 8px;
15        margin: 5px;
16        width: 100%;
17      }
18    </style>
19  </head>
20  <body>
21    <h3>Set & Get Cookie</h3>
22    <input id="cookieVal" type="text" placeholder="Enter cookie value" />
23    <button id="setBtn">Set Cookie</button>
24    <button id="getBtn">Get Cookie</button>
25    <p id="display" style="margin-top: 10px; font-weight: bold"></p>
26
27    <script>
28      function setCookie(name, value, days) {
29        const d = new Date();
30        d.setTime(d.getTime() + days * 24 * 60 * 60 * 1000);
31        document.cookie = `${name}=${value};expires=${d.toUTCString}
32      }
33      function getCookie(name) {
34        const nameEQ = name + "=";
35        const ca = document.cookie.split(";");
36        for (let i = 0; i < ca.length; i++) {
37          let c = ca[i].trim();
38          if (c.indexOf(nameEQ) === 0)
39            return c.substring(nameEQ.length);
40        }
41        return null;
42      }
43
```

```
44         document.getElementById("setBtn").onclick = () =>
45             setCookie(
46                 "labCookie1",
47                 document.getElementById("cookieVal").value,
48                 30,
49             );
50         document.getElementById("getBtn").onclick = () =>
51             (document.getElementById("display").textContent =
52                 getCookie("labCookie1") || "❌ No cookie found");
53     </script>
54 </body>
55 </html>
```

HTML View

Question 2

Problem Statement

■ Write in lab record

Considering exercise 1, modify the web page to display the cookie's name-value pair separately as shown in figure below.

Note: Check lab manual for images/figures

Demo

✗ Do not copy. Read for understanding and the viva

All Cookies (Name-Value Pairs)

Parse & Display

No cookies found

Code

■ Write in lab record

time.html

```
1 <!doctype html>
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8" />
5     <title>Ex2: Parse Cookies</title>
6     <style>
7       body {
8         font-family: sans-serif;
9         padding: 20px;
10      }
11      ul {
12        list-style: none;
13        padding: 0;
14      }
15      li {
16        background: #f4f4f4;
17        margin: 5px;
18        padding: 10px;
19        border-left: 4px solid #333;
20      }
21    </style>
22  </head>
23  <body>
24    <h3>All Cookies (Name-Value Pairs)</h3>
25    <button id="parseBtn" style="padding: 8px 16px; margin-bottom: 15px">
26      Parse & Display Cookies
27    </button>
28    <ul id="cookieList">
29      <li>No cookies yet</li>
30    </ul>
31
32    <script>
33      document.getElementById("parseBtn").onclick = () => {
34        const raw = document.cookie;
35        const list = document.getElementById("cookieList");
36        list.innerHTML = "";
37        if (!raw) {
38          list.innerHTML = "<li>No cookies found</li>";
39          return;
40        }
41
42        raw.split(";").forEach((pair) => {
43          const [name, value] = pair.split("=").map((s) => s.trim(
```

```
44         if (name)
45             list.innerHTML += `<li><strong>${name}</strong> ${v
46         });
47     };
48     </script>
49 </body>
50 </html>
```

HTML View

Question 3

Problem Statement

■ Write in lab record

A list of color is provided in the form of drop down list. The background color of the web page changes as per the selection of the color from the list by the user. At the same time the chosen color is passed to the cookie. The cookie stores the last choice of a user in a browser which will be used on reloading the web page to display the same color as background color.

Note: Check lab manual for images/figures

Demo

✕ Do not copy. Read for understanding and the viva

Select Background Color (Persisted via Cookie)

White ▼

Current: white

Code

■ Write in lab record

time.html

```
1 <!doctype html>
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8" />
5     <title>Ex3: Color Cookie</title>
6     <style>
7       body {
8         font-family: sans-serif;
9         padding: 20px;
10        transition: background 0.3s;
11      }
12      select {
13        padding: 8px;
14        font-size: 16px;
15      }
16    </style>
17  </head>
18  <body>
19    <h3>Select Background Color (Persisted via Cookie)</h3>
20    <select id="colorSelect" onchange="applyColor(this.value)">
21      <option value="white">White</option>
22      <option value="#ffeb3b">Yellow</option>
23      <option value="#4caf50">Green</option>
24      <option value="#2196f3">Blue</option>
25      <option value="#9c27b0">Purple</option>
26    </select>
27
28    <script>
29      function setCookie(name, value, days) {
30        const d = new Date();
31        d.setTime(d.getTime() + days * 86400000);
32        document.cookie = `${name}=${value};expires=${d.toUTCString}
33      }
34      function applyColor(color) {
35        document.body.style.backgroundColor = color;
36        setCookie("bgColor", color, 365);
37      }
38      // Restore on load
39      window.onload = () => {
40        const match = document.cookie.match(/bgColor=([^;]+)/);
41        if (match) applyColor(match[1]);
42      };
43    </script>
```

```
44     </body>
45 </html>
```

HTML View

Question 4

Problem Statement

■ Write in lab record

Create an HTML web page, as shown below. The cookie1 and cookie2 will be set on pressing Set Cookie1 or Set Cookie2 button and the stored cookie value will be displayed on pressing Get Cookie1 or Get Cookie2 button respectively. On the other hand selectively cookie can be deleted by pressing Delete Cookie1 or Delete Cookie2 button. Display all cookies button will show all the stored cookies.

Note: Check lab manual for images/figures

Demo

✕ Do not copy. Read for understanding and the viva

Cookie Manager

Cookie1 Value	Set	Get	Del
Cookie2 Value	Set	Get	Del
Display All Cookies			
Ready ...			

Code

■ Write in lab record

time.html

```
1 <!doctype html>
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8" />
5     <title>Ex4: Multi-Cookie Manager</title>
6     <style>
7       body {
8         font-family: sans-serif;
9         padding: 20px;
10        max-width: 500px;
11      }
12      input,
13      button {
14        padding: 8px;
15        margin: 4px;
16      }
17      .row {
18        display: flex;
19        gap: 5px;
20        margin-bottom: 10px;
21      }
22      pre {
23        background: #f4f4f4;
24        padding: 10px;
25        border-radius: 5px;
26        white-space: pre-wrap;
27      }
28    </style>
29  </head>
30  <body>
31    <h3>Cookie Manager (Cookie1 & Cookie2)</h3>
32    <div class="row">
33      <input id="val1" placeholder="Value for Cookie1" />
34      <button onclick="set('c1', 'val1')">Set C1</button>
35      <button onclick="get('c1')">Get C1</button>
36      <button onclick="del('c1')">Del C1</button>
37    </div>
38    <div class="row">
39      <input id="val2" placeholder="Value for Cookie2" />
40      <button onclick="set('c2', 'val2')">Set C2</button>
41      <button onclick="get('c2')">Get C2</button>
42      <button onclick="del('c2')">Del C2</button>
43    </div>
```

```
44     <button onclick="showAll()" style="margin-top: 10px; width: 100%">
45         Display All Cookies
46     </button>
47     <pre id="out">Ready... </pre>
48
49     <script>
50         function set(name, valId) {
51             if (!name || !valId) {
52                 document.getElementById("out").textContent =
53                     `Provide valid cookie${name[1]} value`;
54                 return;
55             }
56             const v = document.getElementById(valId).value;
57             document.cookie = `cookie${name[1]}=${v};expires=Fri, 31 Dec
58             document.getElementById("out").textContent =
59                 `Set cookie${name[1]} = ${v}`;
60         }
61         function get(name) {
62             const match = document.cookie.match(
63                 new RegExp(`(^| )cookie${name[1]}=([^;]+)`),
64             );
65             document.getElementById("out").textContent = match
66                 ? `cookie${name[1]} = ${match[2]}`
67                 : `❌ cookie${name[1]} not found`;
68         }
69         function del(name) {
70             document.cookie = `cookie${name[1]}=;expires=Thu, 01 Jan 197
71             document.getElementById("out").textContent =
72                 `Deleted cookie${name[1]}`;
73         }
74         function showAll() {
75             document.getElementById("out").textContent =
76                 document.cookie || "No cookies stored";
77         }
78     </script>
79 </body>
80 </html>
```

HTML View

[Edit this page](#)

< Previous
Session 9

Next >
Introduction