

LOCALSTORAGE, SESSIONSTORAGE Y COOKIES

LocalStorage, **sessionStorage** y **cookies** son propiedades de HTML5 (web storage), que permiten almacenar datos en nuestro navegador web. De manera muy similar a como lo hacen las cookies.

Local Storage

Guarda información que permanecerá almacenada por tiempo indefinido; sin importar que el navegador se cierre.

Session Storage

Almacena los datos de una sesión y éstos se eliminan cuando el navegador se cierra.

Cookies

Las cookies se utilizan para mantener la sesión, para ofrecer personalización o para realizar seguimiento de usuarios. Cuando se realiza la autenticación en un servidor la sesión se mantiene creando una cookie que contiene un identificador de sesión y en el lado del servidor el identificador de la sesión se mantiene en memoria o externalizada del servidor en una base de datos como Redis o relacional. El identificador de la sesión guardado en una cookie no es más que al menos 128 bits aleatorios únicos para cada usuario.

Las características de Local Storage y Session Storage son:

- Permiten almacenar entre 5MB y 10MB de información; incluyendo texto y multimedia
- La información está almacenada en la computadora del cliente y NO es enviada en cada petición del servidor, a diferencia de las cookies.
- Utilizan un número mínimo de peticiones al servidor para reducir el tráfico de la red.
- Previenen pérdidas de información cuando se desconecta de la red.
- La información es guardada por dominio web (incluye todas las páginas del dominio).

Las propiedades de las cookies son:

- Se mantienen en el cliente.
- Se envían en cada petición al servidor.
- Su tamaño es muy reducido, no pueden superar los 4 KiB.
- Tienen una fecha de expiración.
- Tienen un nombre y guardan un valor.
- Las cookies solo se envían al dominio origen.
- Pueden crearse sin la posibilidad de que desde JavaScript sean accesibles por seguridad.
- Se pueden crear tanto en el lado del servidor, como en el lado del cliente con JavaScript.

Ejemplo de LocalStorage

A continuación, se presenta un ejemplo de cómo funciona este tipo de almacenamiento.

Introducimos los datos y hacemos clic en Guarda. Si cerramos y abrimos nuevamente la pestaña o ventana nuestro navegador, daremos clic en “Cargar” y nos daremos cuenta de que los datos que almacenamos inicialmente siguen ahí, gracias a la propiedad de LocalStorage.

index.html

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width,
    initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <label>Introduce nombre</label>
  <input type="text" name="nombre" id="nombre"><br>
  <label>Introduce email</label>
  <input type="email" name="email" id="email"><br>

  <button id="guardar">guardar</button>
  <button id="mostrar">mostrar</button>
  <hr>
  <div id="salida"></div>
  <script src="eje.js"></script>
</body>
</html>
```

eje.js

```
document.querySelector("#guardar").addEventListener("click",
function(){
  let nombreDOM=document.querySelector("#nombre").value;
  let emailDOM=document.querySelector("#email").value;
  localStorage.setItem("nombre", nombreDOM);
  localStorage.setItem("email", emailDOM);
  alert("Insertado correctamente");
});

document.querySelector("#mostrar").addEventListener("click",
function(){
  let nombre=localStorage.getItem("nombre");
  let email=localStorage.getItem("email");
  let salida=`Nombre: ${nombre}<br>Email: ${email}`;
  document.querySelector("#salida").innerHTML=salida;
});
```

Introduce nombre

Introduce email

guardar

mostrar

Introduce nombre

Introduce email

guardar

mostrar

Nombre: Juan Ferrer

Email: jferrer437@gmail.com

127.0.0.1:5500 dice
Insertado correctamente

Aceptar

Si cerramos el navegador y lo volvemos abrir, los datos no se pierden.

Introduce nombre

Introduce email

Nombre: Juan Ferrer

Email: jferrer437@gmail.com

Ejemplo de Session Storage

En este ejemplo los datos se almacenan en una sesión.

De esta forma, cuando la pestaña o ventana del navegador se cierra, los datos se eliminan de forma permanente y no podrán cargarse nuevamente.

Este método es usado para realizar aplicaciones más seguras y eliminar un almacenamiento innecesario de datos.

index.html

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width,
    initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <label>Introduce nombre</label>
  <input type="text" name="nombre" id="nombre"><br>
  <label>Introduce email</label>
  <input type="email" name="email" id="email"><br>

  <button id="guardar">guardar</button>
  <button id="mostrar">mostrar</button>
  <hr>
  <div id="salida"></div>
  <script src="eje.js"></script>
</body>
</html>
```

eje.js

```
document.querySelector("#guardar").addEventListener("click",
function(){
  let nombreDOM=document.querySelector("#nombre").value;
  let emailDOM=document.querySelector("#email").value;
  sessionStorage.setItem("nombre", nombreDOM);
  sessionStorage.setItem("email", emailDOM);
  alert("Insertado correctamente");
});

document.querySelector("#mostrar").addEventListener("click",
function(){
  let nombre=sessionStorage.getItem("nombre");
  let email=sessionStorage.getItem("email");
  let salida=`Nombre: ${nombre}<br>Email: ${email}`;
  document.querySelector("#salida").innerHTML=salida;
});
```

Introduce nombre

Introduce email

guardar

mostrar

Introduce nombre

Introduce email

guardar

mostrar

Nombre: Juan Ferrer

Email: jferrer437@gmail.com

127.0.0.1:5500 dice
Insertado correctamente

Aceptar

Si cerramos el navegador y lo volvemos abrir, los datos se pierden.

Introduce nombre

Introduce email

Nombre: null
Email: null