

## **Almacenamiento HTML**

El almacenamiento web html permite que las aplicaciones web puedan almacenar datos localmente dentro del navegador del usuario.

Antes de HTML5, los datos de la aplicación debían almacenarse en cookies, incluidos en cada solicitud del servidor. El almacenamiento web es más seguro y se pueden almacenar grandes cantidades de datos localmente, sin afectar el rendimiento del sitio web.

A diferencia de las cookies, el límite de almacenamiento es mucho mayor (al menos 5 MB) y la información nunca se transfiere al servidor.

El almacenamiento web es por origen (por dominio y protocolo). Todas las páginas, desde un origen, pueden almacenar y acceder a los mismos datos.

### **Objetos de almacenamiento web HTML**

El almacenamiento web HTML proporciona dos objetos para almacenar datos en el cliente:

- **window.localStorage**: Almacena datos sin fecha de caducidad.
- **window.sessionStorage**: Almacena datos para una sesión (los datos se pierden cuando se cierra la pestaña del navegador)

Antes de usar WebStorage es necesario asegurarse que el navegador soporta la funcionalidad ya que los navegadores antiguos no tenían soporte. Para ello puede usarse las siguientes líneas de código:

```
if (typeof(Storage) !== "undefined") {  
    // Código para cuando hay soporte.....  
} else {  
    // El navegador no soporta web storage  
}
```

### **El objeto localStorage**

El objeto localStorage almacena los datos sin fecha de caducidad.

Los datos no se eliminarán cuando se cierre el navegador, y estarán disponibles el día, la semana o el año siguiente.

Los datos se guardan en formato **Atributo = Valor** usando una estructura similar a las cookies. Para guardar información de una aplicación en el webstorage del navegador se usa el método **setItem** del objeto **localStorage**.

Para recuperar la información guardada se emplea el método **getItem**

### **Ejemplo:**

#### **// Almacenamiento**

```
localStorage.setItem("Apellido", "Torres");
```

Guarda en el **localStorage** un atributo **Apellido** cuyo valor es **Torres**

#### **// Recuperación**

```
document.getElementById("result").innerHTML =  
localStorage.getItem("Apellido");
```

Recupera el valor guardado en el atributo **Apellido** y lo carga en el **innerHTML** del elemento HTML cuyo id es **result**

**o esta versión también es válida:**

```
document.getElementById("result").innerHTML =  
localStorage.Apellido;
```

**localStorage.Atributo** devuelve el valor guardado si este existe, sino devuelve false, por lo que puede usarse también para averiguar si se guardó un valor de esta manera:

```
if(localStorage.Apellido) // devuelve true si hay un valor guardado  
previamente
```

**Atención:** Los valores guardados en el navegador son siempre String, conviértalos a lo que sea necesario si no corresponden formalmente a dicho formato.

Si desea guardar información que solo esté disponible durante una sesión, puede emplear el objeto **sessionStorage**. Este objeto es similar a **localStorage**, solo que la información se perderá al cerrar la pestaña del navegador que contenga la página que evocó al objeto.

### **Actividad N°1:**

Cree una página que contenga un formulario de registro donde se ingresará Apellido, Nombre, DNI, Calle, Número, Barrio, Provincia, Email.

Si los datos ingresados y correctos, se guardarán usando **localStorage**, de manera que cuando se abra nuevamente el navegador, si hay información previa, un botón recuperar info rellene el formulario con la información registrada.

## Actividad N°2:

Crear una página HTML que permita el registro de alumnos(Apellido, Legajo) y sus calificaciones (consideraremos tres calificaciones por alumno).

La página debe permitir cargar los Datos de los Alumnos, sus notas, calcular el promedio obtenido e indicar la condición (Pueden tomar como base la app que habían construido en actividades anteriores)

La información de los alumnos registrados y sus notas se mostrará en una tabla. Se aplicará el siguiente criterio y estilos:

Promedio < 4 Condición Libre - Color Rojo

4 ≤ Promedio < 7 Condición Regular - Color Verde

7 ≤ Promedio Condición Promocionado - Color Azul

La propuesta de página HTML de Registro debería ser algo similar a lo siguiente:



The screenshot shows a web browser window with the address bar displaying "C:/Users/CARLOMAGNO/Documents/alumnos.html". The page title is "REGISTRO DE ALUMNOS". Below the title, there is a link "Ingresar Nuevo Alumno". A form with five input fields labeled "Apellido", "DNI", "Nota 1", "Nota 2", and "Nota 3" is shown, followed by a "Guardar" button. Below the form, there is a link "Listado de Alumnos Registrados". A table with 8 columns (Apellido, DNI, Nota 1, Nota 2, Nota3, Promedio, Condicion, and a delete button) displays three rows of student data: TORRES (Promocionado), GOMEZ (Libre), and HERRERA (Regular).

Apellido	DNI	Nota 1	Nota 2	Nota 3	
					Guardar

  

Apellido	DNI	Nota 1	Nota 2	Nota3	Promedio	Condicion	
TORRES	1213	7	6	8	7	Promocionado	Borrar
GOMEZ	6589	3	4	2	3	Libre	Borrar
HERRERA	6325	5	6	5	5.33	Regular	Borrar

**Tips:** Cuando se oprime el botón Guardar, los datos ingresados en los inputs ubicados en el área Ingresar Nuevo Alumno, deben agregarse a la tabla de manera dinámica. Los datos se guardarán en **localstorage**, de manera que al reabrir el navegador, se recupere el listado de los alumnos cargados anteriormente.

## Actividad N°3:

Modifique la actividad anterior para que el listado se muestre ordenado alfabéticamente por Apellido

## Actividad N° 4:

Agregue a la actividad 2 una función de búsqueda que permita encontrar un alumno en particular y mostrar el registro en particular del mismo.

### **Actividad N° 5:**

Agregue una función a la actividad 2 que permita ordenar la tabla haciendo clic en el encabezado de una columna de la misma, por ejemplo si se hace clic en la columna DNI, se ordena por este criterio.

### **Actividad N°6:**

Agregue un botón Editar al lado de cada registro, de manera que se puedan modificar los datos registrados de un alumno en particular. (**localStorage** no admite edición en forma directa, debe reemplazar el registro existente por el nuevo editado)