

# Introduccion

Con el almacenamiento web, las aplicaciones web pueden almacenar datos localmente dentro del navegador del usuario.

Antes de HTML5, los datos de la aplicación tenían que ser almacenados en cookies, incluidos en cada petición al servidor. El almacenamiento web es mas seguro y puede almacenar grandes cantidades de datos sin afectar al rendimiento de la web.

A diferencia de las cookies, el limite de almacenamiento es mucho mas grande (de al menos 5MB) y la informacion no es enviada nunca al servidor.

Antes de usar esta funcionalidad debemos comprobar que nuestro navegador es compatible con ella con el siguiente codigo:

```
if (typeof(Storage) !== "undefined") {  
  // El navegador es compatible con la tecnologia.  
} else {  
  // El navegador no es compatible con la tecnologia.  
}
```

Como nota añadir que los valores almacenados con esta tecnologia se guardan como una cadena (String) que deberá ser convertida a otro formato en caso de ser necesario. Por ejemplo podriamos guardar un objeto usando **JSON.stringify(mi\_objeto)** y así poder guardar objetos en dichas variables.

El caso mas habitual de uso de sessionStorage es para desarrollar paginas web totalmente en FrontEnd, ya que sustituye las variables de sesión de BackEnd.

El almacenamiento en el navegador en el navegador tiene 2 tipos de almacenamiento que pasaré a explicar:

## LocalStorage

Los datos almacenados no expiran, es decir, se conservan incluso despues de cerrar el navegador, pudiendo ser consultada el siguiente dia, mes o año.

Los datos estan almacenados en pares de clave/valor.

Para almacenar un dato en el navegador es tan simple como acceder a **localStorage** como un objeto o llamar a alguno de los metodos que posee dicho objeto.

Se puede usar las siguientes nomenclaturas:

- Para asignar:

```
localStorage.setItem("lastname", "Smith");
```

```
localStorage.lastname = "Smith";
```

- Para acceder:

```
localStorage.getItem("lastname");
```

```
localStorage.lastname;
```

- Para iterar:

Se le pasa un numero, en caso de que exista algo con ese indice lo devuelve, en otro caso devuelve null.

```
localStorage.key(0);
```

- Para eliminar:

```
localStorage.removeItem("lastname");
```

Este borra todos los datos almacenados con esta tecnologia.

```
localStorage.clear();
```

Añado un ejemplo HTML en la carpeta **Ejemplos/LocalStorage**.

## SessionStorage

Los datos almacenados expiran cuando se cierra el navegador.

Los datos estan almacenados en pares de clave/valor.

Para almacenar un dato en el navegador es tan simple como acceder a sessionStorage como un objeto o llamar a alguno de los metodos que posee dicho objeto.

Se puede usar las siguientes nomenclaturas:

- Para asignar:

```
sessionStorage.setItem("lastname", "Smith");
```

```
sessionStorage.lastname = "Smith";
```

- Para acceder:

```
sessionStorage.getItem("lastname");
```

```
sessionStorage.lastname;
```

- Para eliminar:

```
sessionStorage.removeItem("lastname");
```

Este borra todos los datos almacenados con esta tecnologia.

```
sessionStorage.clear();
```

Añado un ejemplo HTML en la carpeta ***Ejemplos/SessionStorage***.