¿Qué es HTML almacenamiento local?

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

Antes de HTML5, los datos de aplicaciones tenían que ser almacenados en las cookies, incluidos en cada petición del servidor. El almacenamiento local es más segura, y grandes cantidades de datos pueden ser almacenados localmente, sin afectar el rendimiento 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 local es por origen (por dominio y protocolo). Todas las páginas, de un origen, se pueden almacenar y acceder a los mismos datos.

Soporte para el navegador

Los números de la tabla especifican la primera versión del navegador que es totalmente compatible con el almacenamiento local.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| API |  |  |  |  |  |
| Web Storage | 4.0 | 8.0 | 3.5 | 4.0 | 11.5 |

HTML almacenamiento local de Objetos

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

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

Antes de utilizar el almacenamiento local, comprobar la compatibilidad con exploradores para localStorage y sessionStorage:

if (typeof(Storage) !== "undefined") {  
    // *Code for localStorage/sessionStorage.*  
} else {  
    // Sorry! No Web Storage support..  
}

El objeto localStorage

El objeto localStorage almacena los datos sin fecha de caducidad. No se eliminarán los datos cuando se cierra el navegador, y estarán disponibles al día siguiente, semana o año.

Ejemplo

// Store  
localStorage.setItem("lastname", "Smith");  
// Retrieve  
document.getElementById("result").innerHTML = localStorage.getItem("lastname");

[Inténtalo tú mismo "](http://www.w3schools.com/html/tryit.asp?filename=tryhtml5_webstorage_local)

Ejemplo explicó:

* Crear un nombre localStorage par / valor con el nombre = "apellido" y value = "Smith"
* Recuperar el valor de "apellido" y la inserta en el elemento con id = "número"

El ejemplo anterior también podría escribirse así:

// Store  
localStorage.lastname = "Smith";  
// Retrieve  
document.getElementById("result").innerHTML = localStorage.lastname;

La sintaxis para retirar el elemento localStorage "Apellido" es la siguiente:

localStorage.removeItem("lastname");

**Nota:** Los pares nombre / valor siempre se almacenan como cadenas. Recuerde que convertirlos a otro formato cuando sea necesario!

El ejemplo siguiente cuenta el número de veces que un usuario ha hecho clic en un botón. En este código la cadena de valor se convierte a un número para ser capaz de aumentar el contador:

Ejemplo

if (localStorage.clickcount) {  
    localStorage.clickcount = Number(localStorage.clickcount) + 1;  
} else {  
    localStorage.clickcount = 1;  
}  
document.getElementById("result").innerHTML = "You have clicked the button " +  
localStorage.clickcount + " time(s).";

[Inténtalo tú mismo "](http://www.w3schools.com/html/tryit.asp?filename=tryhtml5_webstorage_local_clickcount)

El objeto sessionStorage

El objeto sessionStorage es igual al objeto localStorage, **excepto** que almacena los datos para una sola sesión. Los datos se eliminan cuando el usuario cierra la ficha específica del navegador.

El siguiente ejemplo se cuenta el número de veces que un usuario ha hecho clic en un botón, en la sesión actual:

Ejemplo

if (sessionStorage.clickcount) {  
    sessionStorage.clickcount = Number(sessionStorage.clickcount) + 1;  
} else {  
    sessionStorage.clickcount = 1;  
}  
document.getElementById("result").innerHTML = "You have clicked the button " +  
sessionStorage.clickcount + " time(s) in this session.";

## Cómo usar Local Storage de Javascript?

**Publicado hace 5 años**|  Etiquetas: [cookies](http://html5facil.com/tag/cookies/), [html5](http://html5facil.com/tag/html5-2/), [Javascript](http://html5facil.com/tag/javascript/), [Local Storage](http://html5facil.com/tag/local-storage/)

El solo hecho de hablar de **HTML5** provoca pensar en infinidad de tecnologías que podemos aplicar ahora, una de esas tecnologías es ***Local Storage***.

Local Storage nos permite almacenar hasta 5MB del lado del cliente por dominio, esto nos permite ahora hacer aplicaciones mas robustas y con mas posibilidades. ¿Recuerdan las ***cookies***?, estas no ofrecen algo parecido, pero con el limite de **100KB**, las ***cookies***siempre fueron algo muy poco fiable.

En el**Local Storage** podemos guardar cualquier tipo de información, desde texto hasta multimedia, siempre y cuando no exceda los 5MB. Si se preguntan, ¿es una nueva tecnología y solo **5MB**?, esto es por la seguridad del usuario, sería algo inseguro el contar con más espacio.

#### **¿Cómo usar el Local Storage?**

Como todo en **HTML5**, el uso no es muy complicado, de hecho es algo muy lógico de hacer, no requiere mucho conocimiento o documentación.

Nota: Estos Ejemplos usan **JQuery** para el manejo del contenido en las etiquetas.

#### **¿Cómo guardar en el** Local Storage**?**

|  |  |
| --- | --- |
| 1  2 | /\*Igualando la variable con el valor de la caja de texto y guardándola en localStorage\*/  localStorage["Nombre"] = $("#nombre").val(); |

Como pueden ver, la sintaxis es simple y en una linea de código, con esto ya podemos guardar lo que sea en esa variable, como es **Javascript**, obviamente es compatible con todos los ***Frameworks***.

#### **¿Cómo leer el** Local Storage**?**

|  |  |
| --- | --- |
| 1  2 | /\*Este ejemplo muestra "Hola tu\_nombre como estas!"\*/  $("#nombre2").text("Hola " + localStorage["Nombre"] + " como estas!"); |

En este ejemplo, podemos ver que con solo poner la sintaxis y entre corchetes el nombre de la variable del ***Local Storage***, podemos obtener su valor.

Con esto y los 5MB disponibles, podemos crear aplicaciones más robustas y con más opciones, es una herramienta que podemos explotar para mejorar ciertas cosas que una aplicación con cookies no funcionaria muy bien.

Nota: A veces los 5MB no son el límite porque algunos **navegadore**s dan más almacenamiento.

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

* **LocalStorage**: Guarda información que permanecerá almacenada por tiempo indefinido; sin importar que el navegador se cierre.
* **sessionStorage**: Almacena los datos de una sesión y éstos se eliminan cuando el navegador se cierra.

Las características de Local Storage y sessionStorage 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 domino web (incluye todas las páginas del dominio)

¡Vamos a los ejemplos!

**LocalStorage**

El siguiente ejercicio es un ejemplo de cómo funciona este tipo de almacenamiento. Primero debemos iniciar el [**DEMO**](https://platzi.com/blog/wp-content/uploads/2015/06/localStorage.html), para ingresar los datos que deseamos y guardarlos. Después, cerraremos y abriremos nuevamente la pestaña o ventana nuestro navegador, daremos clic en “cargar elementos” y nos daremos cuenta de que los datos que almacenamos inicialmente siguen ahí, gracias a la propiedad de LocalStorage.

[javascript]  
<!DOCTYPE html>  
<html>  
<head>             
<title>Ejemplo LocalStorage</title>  
<script src="https://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js">  
</script>  
<script>  
/\*Funcion de Capturar, Almacenar datos y Limpiar campos\*/  
$(document).ready(function(){    
   $(‘#boton-guardar’).click(function(){        
/\*Captura de datos escrito en los inputs\*/        
var nom = document.getElementById("nombretxt").value;  
var apel = document.getElementById("apellidotxt").value;  
/\*Guardando los datos en el LocalStorage\*/  
localStorage.setItem("Nombre", nom);  
localStorage.setItem("Apellido", apel);  
/\*Limpiando los campos o inputs\*/  
document.getElementById("nombretxt").value = "";  
document.getElementById("apellidotxt").value = "";  
   });    
});

/\*Funcion Cargar y Mostrar datos\*/  
$(document).ready(function(){    
   $(‘#boton-cargar’).click(function(){                        
/\*Obtener datos almacenados\*/  
var nombre = localStorage.getItem("Nombre");  
var apellido = localStorage.getItem("Apellido");  
/\*Mostrar datos almacenados\*/      
document.getElementById("nombre").innerHTML = nombre;  
document.getElementById("apellido").innerHTML = apellido;  
   });    
});  
</script>  
</head>  
<center>  
<h1>Ejemplo – localStorage</h1>

<input type="text" placeholder="Nombre" id="nombretxt">

<input type="text" placeholder="Apellido" id="apellidotxt">

<button id="boton-guardar">Guardar</button>

<hr />

Nombre almacenado:  
<label type="text" id="nombre"></label>  
                          
Apellido almacenado:  
<label "text" id="apellido"></label>  
<button id="boton-cargar">  
Cargar elementos  
</button>  
</center>

<hr />

</body>  
</html>  
[/javascript]

**sessionStorage**

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.

[javascript]  
<!DOCTYPE html>  
<html>  
<head>            
<title>Ejemplo sessionStorage</title>    
<script src="https://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js">  
</script>  
<script>  
/\*Funcion de Capturar, Almacenar datos y Limpiar campos\*/  
$(document).ready(function(){    
   $(‘#boton-guardar’).click(function(){        
/\*Captura de datos escrito en los inputs\*/        
var nom = document.getElementById("nombretxt").value;  
var apel = document.getElementById("apellidotxt").value;  
/\*Guardando los datos en el LocalStorage\*/  
sessionStorage.setItem("Nombre", nom);  
sessionStorage.setItem("Apellido", apel);  
/\*Limpiando los campos o inputs\*/  
document.getElementById("nombretxt").value = "";  
document.getElementById("apellidotxt").value = "";  
   });    
});  
/\*Funcion Cargar y Mostrar datos\*/  
$(document).ready(function(){    
   $(‘#boton-cargar’).click(function(){                        
/\*Obtener datos almacenados\*/  
var nombre = sessionStorage.getItem("Nombre");  
var apellido = sessionStorage.getItem("Apellido");  
/\*Mostrar datos almacenados\*/      
document.getElementById("nombre").innerHTML = nombre;  
       document.getElementById("apellido").innerHTML = apellido;  
   });    
});  
</script>  
</head>  
<center>  
<h1>Ejemplo – sessionStorage</h1>

<input type="text" placeholder="Nombre" id="nombretxt">

<input type="text" placeholder="Apellido" id="apellidotxt">

<button id="boton-guardar">Guardar</button>

<hr />

Nombre almacenado:  
<label type="text" id="nombre"></label>  
                          
Apellido almacenado:  
<label "text" id="apellido"></label>  
<button id="boton-cargar">  
Cargar elementos  
</button>  
</center>

<hr />

</body>  
</html>  
[/javascript]