

TEMA 8

Almacenamiento de datos en el lado cliente

CUESTIONES. Soluciones.

1 ALMACENAMIENTO WEB

http://www.w3schools.com/html/html5_webstorage.asp

Y también:

<http://www.maestrosdelweb.com/actualidad/tutorial-local-session-storage/>

1.2 Las cookies

¿Qué es una sesión?

Es una técnica para vincular datos de un usuario identificado a lo largo de los distintos accesos a una aplicación web.

¿Qué ocurrirá con una cookie para la que no hemos especificado fecha de expiración?

Se eliminará cuando el usuario cierre todas las instancias del navegador.

¿Cuándo y cómo se envían las cookies al servidor?

La información contenida en la cookie será enviada junto con cada petición HTTP al servidor.

¿Qué ventaja tiene la forma de enviar las cookies al servidor?

Que el servidor tiene acceso inmediato a los datos del cliente.

¿Qué desventaja tiene la forma de enviar las cookies al servidor?

La disminución del rendimiento y las limitaciones de la cantidad de datos a ser almacenados.

1.3 Problemas con las cookies

¿Cuáles son los principales inconvenientes de las cookies?

- Cada navegador tendrá sus propias cookies.
- NO diferencian entre usuarios que utilicen el mismo navegador en una misma sesión del

sistema.

- Son vulnerables a los sniffer.
- Pueden ser modificadas en el cliente, lo cual podría aprovechar vulnerabilidades del servidor.

1.4 Las cookies de Flash

1.5 La especificación web storage de la W3C.

¿Cuáles son los nuevos objetos que incluye HTML5 para el almacenamiento de datos en el cliente?
Los sessionStorage y los localStorage.

¿Por qué, mediante *web storage*, es posible almacenar gran cantidad de información sin afectar al rendimiento de la aplicación web?

Porque la información sólo podrá ser accedida desde el lado del cliente.

SessionStorage

Si tenemos 2 pestañas abiertas en el mismo navegador, ¿podrán compartir un mismo objeto sessionStorage esas 2 ventanas? ¿Por qué?

NO. Porque el objeto sessionStorage se instancia por sesión y ventana, por lo que dos pestañas del navegador abiertas al mismo tiempo y para un mismo sitio web pueden tener información distinta.

¿Qué pasa con la información almacenada en un sessionStorage al cerrar la sesión?
SE pierde la información.

¿Cómo podemos especificar el tiempo de expiración en una sessionStorage?
No se puede.

¿Qué método se utiliza para agregar un nuevo par clave/valor?
setItem()

¿Qué método se utiliza para eliminar un par clave/valor?
removeItem()

¿Qué hace el método clear()?
Borra todos los elementos de la lista sessionStorage.

LocalStorage

¿Para qué está diseñado el mecanismo de almacenamiento localStorage?
Para los datos que se extienden a lo largo de múltiples ventanas y múltiples sesiones.

¿Cuándo se dispara el evento StorageEvent?
Cuando se produce cualquier tipo de cambio en el almacén de datos.

La persistencia de los datos

NOTA: Los apartados que restan del tema los estudiaremos sólo a nivel teórico, no veremos ejemplos ni ejercicios sobre los dos apartados restantes del tema.

2 BASES DE DATOS SQL (STANDARD QUERY LANGUAGE) EN ENTORNO CLIENTE

¿Cuál es la ventaja de tener una base de datos local?

La ventaja de tener una base de datos local en vez de utilizar una base de datos en el servidor es que las bases de datos locales permiten almacenar datos en el cliente teniendo acceso a esta información sin necesidad de estar en línea.

2.2 WEBSQL

Ya no se encuentra en mantenimiento activo por el grupo de trabajo de aplicaciones web de la W3C desde noviembre de 2010.

2.3 INDEXED DATABASE API

Definiciones

¿Qué tipo de datos almacena IndexedDB?

Objetos.

¿Para qué se utilizan las consultas SQL en IndexedDB?

En IndexedDB no se trabaja con el concepto de consultas SQL.

¿Qué se almacena en el Object Store o contenedor?

Almacena pares clave/valor.

¿Qué diferencia hay entre los pares clave/valor almacenados en el Object Store y los que almacenaban en los objetos DOM Storage vistos anteriormente?

La diferencia se encuentra en que los valores, por lo general, serán objetos con una estructura compleja.

¿Qué es la “clave”?

Un valor a partir del cual los objetos almacenados son organizados.

¿Qué permite hacer un índice?

Un índice se crea en un contenedor y permite buscar objetos dentro del mismo en base a los valores que contiene y no solamente en base a la clave del objeto.

Crear, abrir y eliminar una base de datos

¿Cómo funciona el método open()?

Crea la base de datos si esta no existe o la abre en caso de que exista.

¿Qué evento se dispara si el método open() funciona correctamente?

El evento success, y se aplica la función indicada en la propiedad onsuccess.

¿Dónde se “devuelve” la base de datos después de ejecutarse el método open()?

La base de datos estará accesible a través de la propiedad result del evento que se pasa como parámetro a la función indicada en la propiedad onSuccess.

¿Qué error se producirá en caso que intentar borrar una base de datos inexistente?

No se producirá ningún error.

Creación de contenedores e índices

Antes de poder almacenar datos debemos definir un contenedor de objetos ¿cuál sería el equivalente de un contenedor de objetos en una base de datos relacional?

Sería similar a las tablas de una base de datos relacional.

¿Con qué función se puede crear un índice?

Mediante la función createIndex()

¿Para qué se utiliza el método add()?

Para cargar objetos en el contenedor.

Operaciones con la base de datos

¿En base de datos, qué se entiende por transacción?

En base de datos se indica como transacción aun conjunto de operaciones que se aplican formando una unidad de tal manera que dentro de la transacción se mantiene la integridad de los datos, haciendo que estas operaciones no puedan finalizar en un estado intermedio.

¿Qué ocurre si se cancela una transacción antes de que ésta haya terminado completamente?

En caso de que se cancele la transacción en alguna operación intermedia, se empiezan a deshacer las modificaciones aplicadas hasta dejar la base de datos en su estado inicial, como si las operaciones dentro de la transacción nunca se hubiesen realizado.

¿Cuáles son los tres parámetros que forman una transacción en IndexedDB?

Los contenedores implicados en la transacción.

El modo de a transacción.

La instancia de la base de datos de la transacción.

¿Si no se indica ningún contenedor en la transacción, qué contenedores estarán en la misma?

Se asume que todos los contenedores están en la transacción.

¿Qué indica el modo de la transacción?

Indica el tipo de operación a ser realizada y si dos transacciones pueden aplicarse de manera concurrente.

¿En cuántos modos se puede abrir una transacción?

- Read_Only.
- Read_Write.
- Version_Change.

¿Qué significa que abramos una transacción en modo Read_Only?

Solo lectura y sin modificación de datos.

¿Qué significa que abramos una transacción en modo Read_Write?

Permite la lectura y escritura de datos en contenedores existentes.

¿Qué significa que abramos una transacción en modo Version_Change?

Permite cualquier operación, incluyendo la creación y borrado de contenedores e índices.

Este tipo de transacción no se ejecutará de manera concurrente junto con ninguna otra transacción.

La obtención de datos

¿Qué método se utiliza para recuperar un objeto simple de la base de datos?

El método get() con la clave del objeto a ser recuperado.

¿Dónde se encontrará el objeto devuelto con el método get()?

El objeto devuelto se encontrará en la propiedad result del evento.

¿Qué necesitaremos utilizar si queremos obtener más de un objeto?

Es necesario utilizar un cursor para recorrerlos.

¿Con qué método se puede abrir un cursor?

Con el método openCursor()

¿El método openCursor nos devuelve directamente el cursor? ¿Dónde estará el cursor?

El cursor es el resultado del evento en caso de éxito, es decir, se encuentra en la propiedad result del evento.

Una vez obtenido el cursor, ¿qué se encuentra en dicho cursor?

En la variable cursor se encuentra el objeto actual obtenido.

¿Con qué método se puede cargar en el cursor el objeto siguiente?

Con el método continue().

3 APLICACIONES EN CACHE

¿En qué consiste la caché de las aplicaciones web?

Es el proceso de almacenar datos generados dinámicamente para que puedan ser utilizados nuevamente sin necesidad de realizar peticiones al servidor.

3.2 HTML5

¿Cuál es el objetivo de las aplicaciones en caché junto a IndexedDB?

La posibilidad de obtener aplicaciones web y sitios web que funcionen correctamente cuando no tienen conexión con el servidor (modo offline).

¿En qué consiste la funcionalidad de la caché?

La funcionalidad consiste en obtener toda la información del sitio web antes de que sea explícitamente pedida por el usuario de tal manera que una vez almacenadas en la caché de aplicación puedan ser utilizadas en modo fuera de línea.

¿Cuál es la diferencia entre la caché habitual y la caché de aplicación?

Como diferencia principal entre la caché habitual de una página web y la caché de aplicación es que en el segundo caso podremos acceder también a páginas que no hayan sido visitadas anteriormente.

El manifiesto

El manifiesto es un fichero de texto plano. ¿Qué extensión tiene dicho fichero?
“.appcache”

¿Qué contendrá el manifiesto?

Los nombres de los ficheros y recursos en general que serán almacenados por el navegador para ser utilizados en modo fuera de línea.