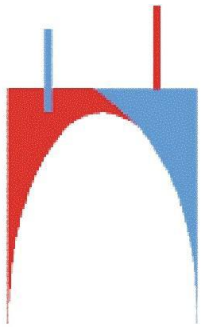

Instituto de Enseñanza Secundaria

Santiago Hernández

DESARROLLO WEB EN ENTORNO CLIENTE

(Curso 2021 – 2022)

UD3. JAVASCRIPT AVANZADO. ALMACENAMIENTO.

ALMACENAMIENTO EN JS

Hechos

- Necesidad: La interacción del cliente con la web requerirá muchas veces almacenar información.
- Ventajas:
  - datos accesibles por el navegador en todo momento.
  - se minimiza la información en tránsito
  - se puede seguir trabajando sin acceso al servidor (off-line)
- En función de la cantidad de información a almacenar tenemos entre otros:
  - Cookies: pequeños archivos almacenados como string.
  - Almacenamiento web (Web Storage).
  - Caché del navegador.

COOKIES

https://www.ionos.es/digitalguide/hosting/cuestiones-tecnicas/cabecera-http/

- Información enviada o recibida en las cabeceras HTTP y que **queda almacenada localmente** durante un tiempo determinado
- Permite almacenar datos temporales (hasta que el usuario las borra)
- Cadena de caracteres con la estructura nombre=valor que el navegador puede almacenar localmente en el ordenador del usuario a petición del servidor o del documento.
- Su contenido se reenvía al servidor con cada nueva petición.
- Capacidad:
  - Cookies en un navegador: 300 al menos.
  - Tamaño de cada cookie: 4kB.

COMUNICACIÓN CLIENTE - SERVIDOR

Sin cookies

- Usuario envía una solicitud
- Servidor envía la información y se olvida
- Usuario vuelve a solicitar la web
- Servidor lo trata como la primera vez

Con cookies

- Usuario envía una solicitud
- Servidor envía información + cookies
- Usuario vuelve a solicitar la web con cookies
- Servidor analiza las cookies identificando al usuario y adaptando la respuesta en consecuencia.

COOKIES

Ventajas

- Recordar personalización de interfaces (presentación y funcionalidad)
- Seguimiento de un usuario a través de un sitio web (cesta de la compra)
- Seguimiento por diferentes sitios web (cookies de terceros) → permiten crear perfiles para las compañías de publicidad

Inconvenientes

- Pérdida de privacidad al crearse un perfil de usuario que se envía al servidor
- Identificación inexacta cuando se comparte cuenta de usuario
- Posible manipulación fraudulenta

COOKIES

Acceso desde JS

- Parámetros de cookies opcionales:
  - **expires**: hasta cuando dura la cookie. Prevalece sobre max-age.
  - **max-age**: Duración máxima de la cookie en segundos.
  - Si no se establece ningún atributo de duración, desaparece al terminar la sesión.
  - **path:** Ruta para la cuál la cookie es válida. Si no se especifica nada será válida para el directorio actual.
  - **domain**: Dominio para el cuál la cookie es válida. Por defecto el subdominio actual. Por seguridad no se permite crear cookies para dominios diferentes al que crea la cookie.
  - **secure**: Si se indica sólo será válida para conexiones encriptadas (HTTPS)

https://cybmeta.com/que-son-las-cookies-y-como-funcionan

COOKIES

Acceso desde JS

- Acceso a las cookies de una web:
  - document.cookie;
- Creación de cookies:
  - document.cookie = “usuario=Javier“;
- Codificación de cookies (si es necesario, acorde a su ubicación en la cabecera HTTP):
  - encodeURIComponent(): valores dados a la pareja nombre=valor
  - Date.toUTCString(): fechas en formato UTC
- Eliminación de cookies: https://cursosgs.es/dwec/2017/11/02/2-15-cookies/
  - document.cookie = "dwec1=;max-age=0;path=/dwec";

WEB STORAGE

ALMACENAMIENTO WEB

Características

- Capacidad hasta 4MB (frente a los 4KB de las cookies)
- La información que almacenan nunca se envía al servidor.
- Permiten almacenar texto y multimedia.
- La información se almacena por dominio.
- La información se almacena por pares clave:valor.
- Opciones:
  - LocalStorage: los datos se guardan hasta que el navegador los elimina.
  - SessionStorage: los datos se guardan hasta que la pestaña del navegador se cierra.

WEB STORAGE

Acceso desde JS

- El Web Storage dispone de 3 métodos de acceso:
  - setItem: crea una pareja clave:valor:
    - localStorage.setItem("ciudad","Zaragoza");
  - getItem: obtiene el valor de un ítem a partir de su clave:
    - var a = localStorage.getItem("ciudad"); //almacena el string “Zaragoza” en a
  - removeItem: elimina un ítem a patir de su clave:
    - localStorage.removeItem("ciudad");
- Ejemplo: https://codepen.io/flippedsandra/pen/XWJWpmp?editors=0011

ARCHIVOS JSON

Características

- JavaScript Object Notation (JSON) es un formato basado en texto estándar para representar datos estructurados en la sintaxis de objetos de JavaScript.
- Permite transmitir datos en aplicaciones web.
- Un archivo JSON es un string con formato similar a la estructura de un objeto inmutable.
- Puede usarse de forma independiente, no sólo con JavaScript.
- Una cadena convertida a un objeto nativo se conoce como parsear (parsing).
- Los archivos JSON tienen la extensión .json.

importante lo que está en el recuadro

Diferencias entre objetos y arrays: http://qbit.com.mx/blog/2013/01/13/arreglos-vs-objetos-en-javascript-diferencias/
Convertir de JSON a objeto JavaScript y viceversa------ https://lenguajejs.com/javascript/caracteristicas/json/

ACCESO A ARCHIVOS EN EL CLIENTE

FileReader

- Objeto que permite leer ficheros almacenados en el cliente de forma asíncrona:
  - var lector = new FileReader();
- File: tipo de objeto para acceder al contenido de un archivo.
- Por motivos de seguridad, el archivo a leer por JavaScript debe ser siempre seleccionado por el usuario, y su lectura irá asociada a un evento.
- Los formularios nos ofrecen la entrada de tipo file para manejar archivos:
  - <input type="file" id="archivo" onchange="abrirArchivo(this.files);">
- Para abrir el archivo en modo lectura con nuestro lector:
  - lector.readAsText(archivo);
- Para acceder al contenido:
  - var contenido = lector.result;