**Examen 2ª evaluación. DAW2 (presencial)- 20 de marzo de 2018**

***MATERIAL ENTREGADO***

Una carpeta comprimida con un archivo html, 4 archivos js, 1 archivo css y un archivo de vídeo:

* *Archivo Donante.js: con el código de la definición del objeto Donante. Se entrega la definición del objeto con sus propiedades y métodos. También contiene métodos adicionales que facilitan la resolución del problema.*
* *Archivo InicializarDatos. js con variables globales a utilizar en el ejercicio.*
* *Archivo donaciones.js vacío que es donde tenéis que hacer vuestro programa. Es el único que hay que modificar, aunque para hacer una de las pruebas habrá que modificar la variable visitas en el archivo inicializarDatos.js.*
* *Archivo videoFuncionamiento.wmv donde se ha capturado el funcionamiento del programa. Ahí podéis ver cómo debe comportarse el programa ante la interacción del usuario. Hay que visualizarlo con el Windows Media Player.*
* *Archivo donaciones.html con:*
  + *los 4 archivos js mencionados enlazados.*
  + *el archivo CSS enlazado.*
  + *los objetos del DOM necesarios para el funcionamiento de la aplicación.*

***REQUISITOS A CUMPLIR***

**IMPORTANTE: No se corregirá ningún examen que no cumpla con los siguientes requisitos:**

1. *Solo se puede usar el break en el comando switch.*
2. *Solo se puede emplear el modelo de eventos del W3C visto en la unidad 5.*
3. *Tabulación y estructura correcta.*
4. *Comentarios en variables, bloques de código y prototipos de funciones.*

***INTERFAZ (Diseño- archivo campanaDonacion.html)***

El archivo dispone de todos los objetos necesarios con los que el usuario podrá interactuar.

Algunos están inactivos, otros están ocultos y se habilitan o muestran según el comportamiento que se puede visualizar en el vídeo.

Los objetos de los que dispone la interfaz son:

1. Área de texto para mostrar el nombre del instituto al que se está realizando la visita y el nombre del responsable de la cruz roja.
2. Botón para comenzar el proceso de donación.
3. Etiquetas: una por cada objeto de la interfaz.
4. Objeto input para introducir el DNI del donante.
5. Botón para comprobar DNI del donante para evitar que done 2 veces. No hay que realizar validación del formato del DNI, ni que el DNI sea correcto.
6. Objetos select para: grupos sanguíneos, edades y pesos. Habrá que crear dinámicamente sus opciones según las restricciones del enunciado indicadas más adelante.
7. Botones de opción para la selección del sexo del donante.
8. Botón para hacer la donación.
9. Elemento div para la lista de las donaciones realizadas. Habrá que agregar de forma dinámica una línea de texto por cada donación realizada con los datos de dicha donación.
10. Botón para terminar el proceso de realizar donaciones.
11. Botón para visualizar las estadísticas.
12. Elemento div para añadir las estadísticas de las donaciones realizadas por cada grupo sanguíneo. Habrá que agregar de forma dinámica una línea con la estadística de cada grupo sanguíneo incluyendo grupo, nº de donaciones realizadas y litros donados de dicho grupo empleando los métodos del DOM.
13. Elemento div para añadir las estadísticas de las donaciones realizadas agrupadas por sexo. Habrá que agregar de forma dinámica una fila con la estadística de sexo incluyendo el sexo y el nº de donaciones realizadas por donante de dicho sexo empleando los métodos del DOM.

***SE PIDE: Enunciado***

Realizar una aplicación que permita realizar, en el instituto que corresponda según la fecha actual, donaciones de sangre por donantes que cumplen los requisitos del enunciado y, permita visualizar las donaciones realizadas, las estadísticas de donaciones realizadas por grupo sanguíneo (grupo, nº de donaciones y litros donados) y por sexo (nº de donaciones realizadas por hombres y por mujeres). La aplicación deberá almacenar como cookies, una vez presentadas las estadísticas, el número de donaciones realizadas, el nombre del instituto y la fecha de realización de la visita.

Indicaciones que se pueden ver en el vídeo:

* El autobús tiene una serie de visitas programadas en diferentes días a diferentes institutos.
* Cada uno de esos días hay un responsable de la cruz Roja que viaja en el autobús supervisando todas las operaciones realizadas.

Al arrancar la aplicación solo puede verse, en la zona habilitada para ello, el nombre del instituto y la persona responsable de la cruz roja para ese día. De ser el caso correcto, también estará visible el botón que permite comenzar las donaciones y se creará el objeto de tipo Donante con los datos iniciales disponibles.

*Si arrancásemos la aplicación en un día que no figura en los datos iniciales debería verse en dicha zona "¡¡¡HOY DESCANSAMOS!!!" y nada más. Podéis hacer la prueba cambiando la fecha en datosIniciales.js*

* Cada donación se reflejará de forma inmediata en la tabla destinada para ello.
* Al finalizar la visita quedarán a la vista los datos de:
  + La visita reflejados en el área de texto.
  + La tabla con las donaciones realizadas y el botón que permite ver las estadísticas que al pulsarlo debe:
    - Mostrar la zona con la estadística de donaciones realizadas por grupo sanguíneo.
    - Mostrar la zona con la estadística de donaciones realizadas por sexo.
  + Grabar en una cookie los datos generales de la visita y el nº de donaciones realizadas.

*RESTRICCIONES SOBRE EL ENUNCIADO*

* *Se consideran donantes válidos todas aquellas personas mayores de edad (edad>=18) con un peso mayor de 50 kg.* 
  + *Se establece un límite máximo de edad en 65 años.*
  + *Se establece un límite máximo de peso en 120 kg.*
* *Cada persona solo puede donar solo una vez en ese día. (controlado por el DNI del donante)*
* *La cantidad extraída a cada donante es de 450cc. (1000cc = 1 litro)*

NOTA: NOSE PIDE VALIDACIÓN DE DNI (ni formato, ni validación de letra)

***INTERFAZ (Comportamiento: Cómo debe comportarse la interfaz)***

El resto del comportamiento se podrá proyectar en el cañón del aula o verse en el vídeo.

Al visualizar el vídeo se verá la interacción del usuario con el programa para determinar:

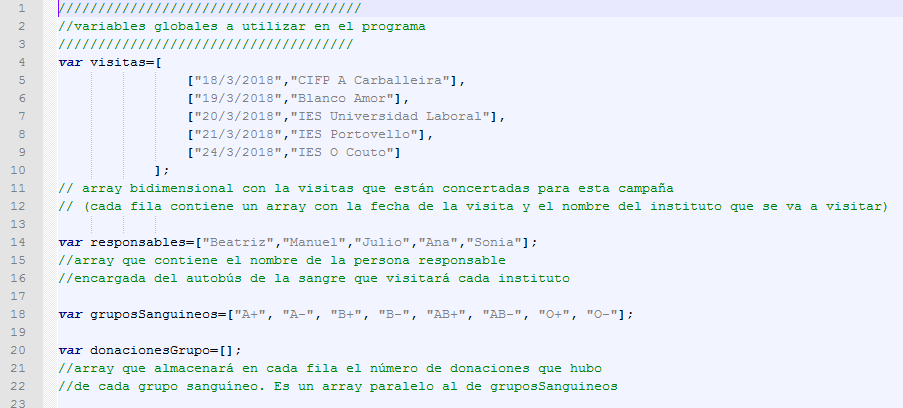
* qué está visible en cada momento
* qué, estando visible, está habilitado en cada momento.
* qué mensaje tiene que salir en cada momento dependiendo de la operación realizada y según las circunstancias.

***ARCHIVO: campanaDonacion.html***

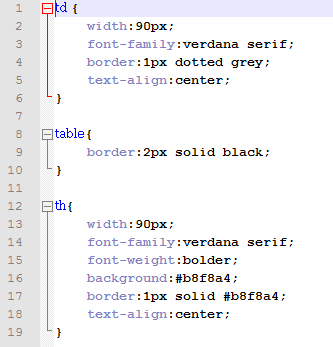




***ARCHIVO: InicializarDatos.js***

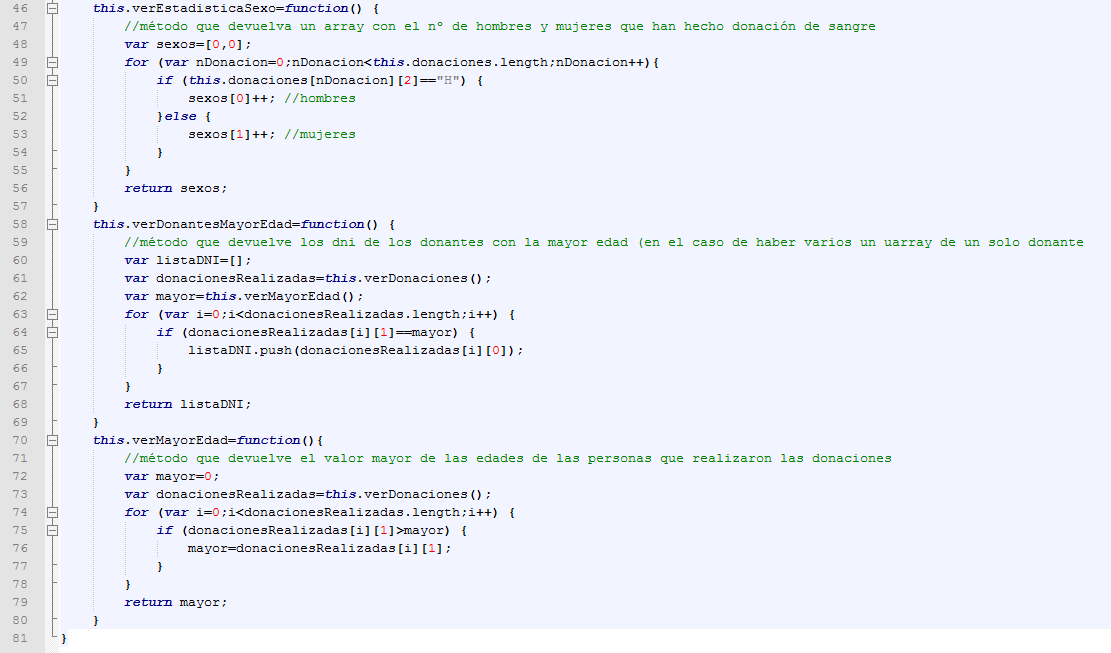


***ARCHIVO: misEstilos.css***



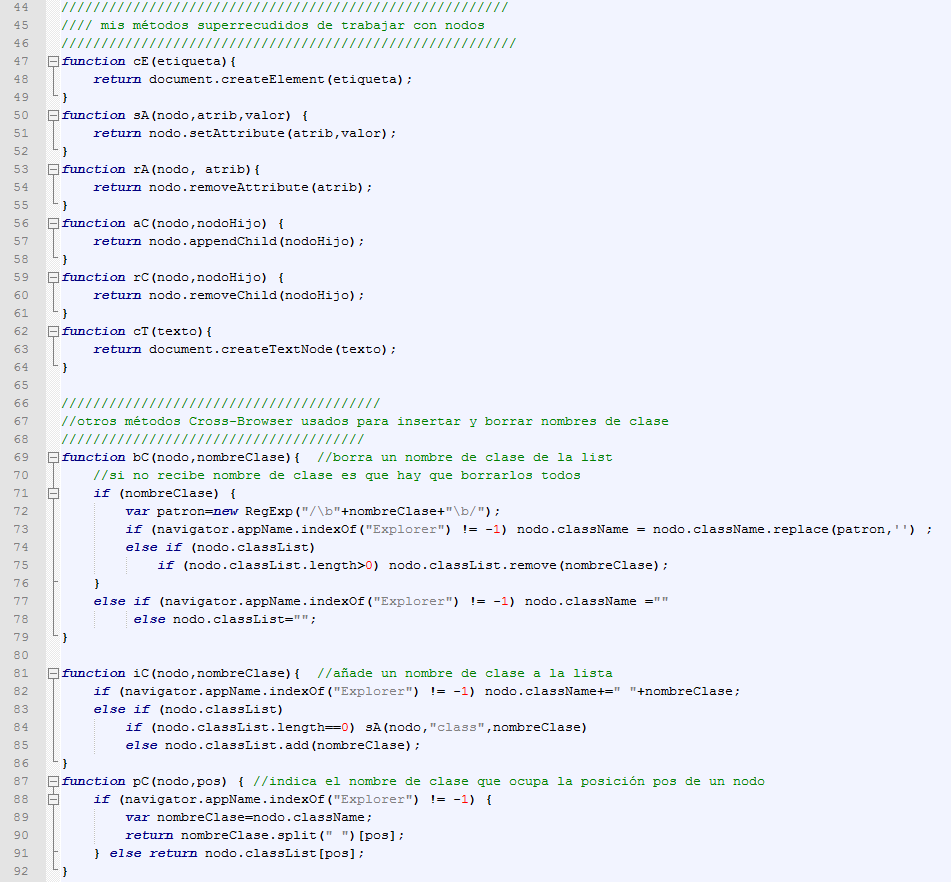
***ARCHIVO: Donante.js***

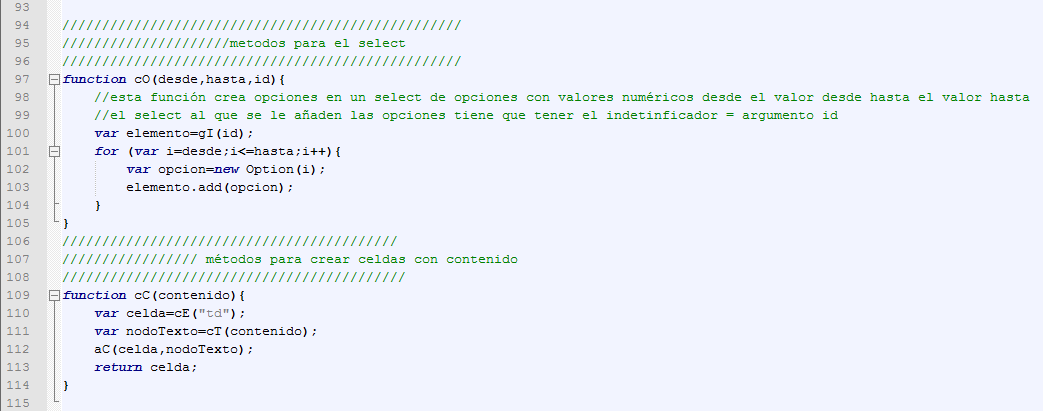




***ARCHIVO: miLibreria.js***

******

******

******

***PUNTUACIÓN.***

***Nombre del alumno/a***

|  |  |  |
| --- | --- | --- |
| Apartado | Puntos | Comentarios |
| Control de la fecha para la visualización de los datos de la visita a realizar o día de descanso | 1 | Para 10.  Si no se sabe emplear el objeto Date crear una variable de tipo string con la fecha de hoy que coincida con uno de los valores de la tabla visitas y trabajar con ella |
| Control de la visibilidad/invisibilidad de las diferentes partes del documento | 1 | Obligatorio (25%) |
| Control de la habilitación/inhabilitación de los objetos visibles | 1 | Obligatorio (25%) |
| Creación y uso del objeto de tipo Donante a través de sus métodos (encapsulamiento) | 1 | Obligatorio |
| Creación de las opciones de peso y edad usando el método de la librería | 1 | Obligatorio (50%) |
| Creación de las opciones de grupo sanguíneo | 1 | Para 10  Si no se sabe hacer y se quiere comprobar el funcionamiento del programa crearlas directamente en el HTML |
| Control del DNI repetido | 1 | Para 10 |
| Control de datos del donante completos para poder realizar la donación | 1 | Obligatorio |
| Redacción de mensajes de alerta y confirmación | 1 | Obligatorio (50%) |
| Creación y control de los eventos | 1 | Obligatorio (50%) |
| Aplicación cross-browser | 1 | Obligatorio |
| Creación de las tablas con métodos del DOM | 3 | Obligatorio al menos 1 |
| Calculo de estadísticas empleando los métodos del objeto Donante | 2 | Obligatorio al menos 1 |
| Cambiar estilo en las filas de donaciones donde los donantes tienen una edad igual a la mayor | 1 | Para 10 |
| Almacenamiento de las cookies | 1 | Obligatorio |
| Comentarios en el código.  Para qué vale cada variable.  Para qué vale cada función. | 2 | Obligatorio (50%) |
| Sin errores de sintaxis que se podrían depurar fácilmente con el depurador del navegador | 1 | Obligatorio |
| TOTAL | 21 | Se calculará el equivalente a 10 puntos |

Obligatorios 10 puntos de 21