Desarrollo Web en entorno cliente

Sintaxis, formularios y eventos

Ejercicios de exámenes (E) y prácticas (P) de cursos anteriores

Ejercicio 1. Búsqueda de palabras (E)

El programa debe encontrar cadenas de caracteres en el texto del archivo .js de datos (No es válido copiar el contenido del archivo de datos en el js del programa.)

Tras introducir una cadena de caracteres y pulsar sobre "Iniciar la búsqueda" aparece el texto "REALIZANDO BÚSQUEDA..." durante dos segundos en el espacio de "avisos". Tras este tiempo aparece en el espacio "consola", para cada ocurrencia, la línea y la posición que ocupa el texto, un párrafo para cada una.

***Nota**: para realizar búsquedas válidas utilizar la cadena "ma"*

Se tiene que cumplir que:

- Si no hay texto de búsqueda se da el aviso de "DEBES INTRODUCIR UN TEXTO A BUSCAR"
- Si el texto no existe el aviso es "ESE TEXTO NO EXISTE"

Ejercicio 2. Carrusel de anuncios (E)

Este ejercicio quiere simular la activación de un carrusel de anuncios. Para ello hay que pulsar sobre el número correcto que lo activa. Una vez activado, van rotando cuatro anuncios en el espacio "carrusel" cambiando textos y colores de fondo.

Hay que cumplir lo siguiente para que funcione el carrusel:

1. Pulsar en "Comenzar". Aparece el panel con los números. (el estilo que hace aparecer el panel es *visibility* con valor *visible*)
2. El número para hacerlo funcionar es el 4. Si no es el correcto, aparece el texto "CLAVE INCORRECTA".
3. Cuando se pulsa el número 4 ocurre que:

- El botón de inicio cambia de texto a "Parar carrusel".
- Desaparecen los números
- En el espacio "carrusel", cada medio segundo, cambia el texto y el color de fondo. Las marcas y los colores vienen dados en el archivo .js en *fondos* y *marcas*

4. Al pulsar "Parar carrusel", se detiene la rotación de anuncios y se vuelve al estado inicial.

Ejercicio 3. Búsqueda de precios (E)

El programa debe encontrar precios en los productos del archivo .js de datos (No es válido copiar el contenido del archivo de datos en el js del programa.) Los precios vienen dados con identificador, precio y nombre de producto, separados por comas.

Tras introducir un rango de precios y pulsar sobre "Iniciar la búsqueda" aparece el texto "REALIZANDO BÚSQUEDA..." durante dos segundos en el espacio de "avisos". Tras este tiempo aparece en el espacio "consola", para cada producto,

- una línea con el precio original y el precio con iva (21%).
- otra línea con el identificador del producto, el nombre y el precio con iva redondeado

***Nota**: para realizar búsquedas válidas utilizar el rango entre 100 y 300€. Los precios son todos menores a 1000*

Se tiene que cumplir que:

- Si no se introduce alguno de los dos valores se da el aviso de "DEBES INTRODUCIR UN RANGO CORRECTO DE PRECIOS"
- Si no hay productos en el rango el aviso es "NO HAY PRODUCTOS EN ESE RANGO DE PRECIOS"

Ejercicio 4. Agenda deportiva (E)

Este ejercicio quiere simular un cartel para un evento deportivo. Para ello hay que pulsar sobre el color correcto que lo activa. Una vez activado, el evento va rotando en cuatro espacios del espacio "carrusel", cambiando el texto y el color de fondo de posición cada medio segundo.

Hay que cumplir lo siguiente para que funcione el anuncio del evento:

1. Pulsar en "Comenzar". Aparece el panel con los colores. (el estilo que hace aparecer el panel es *visibility* con valor *visible*)
2. El color para hacerlo funcionar es el dorado. Si no es el correcto, aparece el texto "OPCIÓN NO VÁLIDA" en el espacio "avisos".
3. Cuando se pulsa sobre el color dorado ocurre que:

- El botón de inicio cambia de texto a "Parar anuncios".
- Los mensajes de avisos desaparecen.
- Desaparecen también la selección de colores.
- En el espacio "carrusel", cada medio segundo, cambia de posición el anuncio con su color de fondo. El texto del anuncio y los colores a usar vienen dados en el archivo .js en *literal*, *starcolor* y *defaultcolor*.

4. Al pulsar "Parar anuncios", se detiene el anuncio y se vuelve al estado inicial.

Ejercicio 5. Búsqueda de comerciales (E)

El programa debe encontrar los comerciales de empresas que hay registradas en una provincia.

El listado donde buscar está en el archivo .js de datos. Cada comercial vienen dados por su nombre completo, la empresa para la que trabaja, y la provincia. Los campos están separados por comas. (No es válido copiar el contenido del archivo de datos en el js del programa.)

Tras introducir los caracteres de búsqueda de la empresa, y la provincia, hay que pulsar sobre "Iniciar la búsqueda". La búsqueda de empresa es "contiene los caracteres" en el nombre de la empresa.

(Empresas válidas: Toyota, Microsoft, Facebook, Samsung, Coca-Cola)

Como resultado aparecen en el espacio "consola" los nombres completos de los comerciales, la empresa para la que trabaja y la provincia. Cada comercial aparece en una línea (párrafo)

Se tiene que cumplir que:

- Se realizan las búsquedas cuando los dos campos a rellenar no estén vacíos.
- Si no se introducen caracteres de búsqueda aparece un aviso de "DEBES INTRODUCIR UNA EMPRESA"
- Si no hay seleccionada provincia, el aviso es "SELECCIONA UNA PROVINCIA"
- Si no hay comerciales que cumplan con la búsqueda el aviso es "NO HAY COMERCIALES EN ESA PROVINCIA"

Estos avisos aparecen en el espacio llamado "avisos" del archivo html.

Ejercicio 6. Calcula el instante (E)

Este ejercicio quiere simular un juego para calcular el instante en que va a iluminar un objeto. El funcionamiento es el siguiente:

- Al pulsar sobre el botón de "Comenzar", aparece y se activa, una fila de pequeños cuadros (20) que van cambiando de color progresivamente. También aparece y se activa un botón que sirve para parar el cambio de colores de los cuadros. Al llegar al final de la fila cambia el color entre seis colores predefinidos.

El estilo que hace aparecer los cuadros es *visibility* con valor *visible*.

- Al pulsar con el ratón sobre el cuadro con el símbolo **O** se para la rotación de los colores. Aparece un mensaje en el espacio de avisos "TU NIVEL ES X" siendo X el número del cuadro sobre el que se ha pulsado dividido por 2. Como hay 20 cuadros, los niveles son progresivos cada 0,5 de 0 a 10. *(**Nota**: no es necesario el control del límite del último cuadro, pero sí que la rotación sea correcta)*

- Para volver a activar la rotación de colores de los cuadros, se pulsa de nuevo el botón **O**.
- Al llegar al final de la fila de cuadros, que tienen todos el mismo color, comienzan a cambiar a otro color desde el principio. El aviso que hubiese escrito desaparece.

Los colores a usar vienen dados en el archivo .js en la variable *colorines*.

- El botón "Comenzar" al ser pulsado y empezar el juego, cambia de texto a "Parar" para parar el juego.

Para facilitar el ejercicio, los cuadros cambian de color cada medio segundo (se aumenta la velocidad con menos tiempo).

Ejercicio 7. Búsqueda de número premiado (E)

El programa debe encontrar números premiados a partir de una lista que se proporciona en el archivo .js de datos (No es válido copiar el contenido del archivo de datos en el js del programa.)

Hay tres posibilidades de búsqueda, por número completo (5 cifras), por terminación (últimas 2 cifras) y por reintegro (última cifra). Tras introducir el número y la opción de búsqueda, hay que pulsar sobre "Iniciar la búsqueda".

Como resultado aparece, en el espacio "consola", los números que cumplen con la búsqueda, la provincia donde ha sido vendido y el premio de cada número, un párrafo para cada uno.

***Nota**: para realizar búsquedas válidas utiliza los números que aparecen en las imágenes, (o busca en el archivo de datos)*

Se tiene que cumplir que:

- Si no hay número a buscar se da el aviso de "DEBES INTRODUCIR UN NÚMERO A BUSCAR"
- Si no se ha seleccionado opción, se da el aviso de "SELECCIONA NÚMERO, TERMINACIÓN O REINTEGRO"
- Si no hay número que cumpla la búsqueda el aviso es "ESE NÚMERO NO TIENE PREMIO"

Estos avisos aparecen en el espacio llamado "avisos" del archivo html.

Ejercicio 8. Pantones de colores (E)

Este ejercicio ofrece una muestra de una gama de distintos colores, sobre la base de tres colores básicos, azul, rojo y verde. Para cada color básico hay otros cuatro de esa gama, que van a ir rotando en los cuadros de abajo.

- Al pulsar sobre el botón "Comenzar Muestra", aparecen los otros dos paneles, un panel de colores para seleccionar el color básico, y otro con cuatro cuadros, de nombre "gama", que muestran los colores de cada gama.
- Al pulsar "Comenzar Muestra" los colores comienzan a rotar, empezando por el color por defecto (no se nota porque es el mismo color en los cuatro).
- Al pasar por encima de uno de los tres colores básicos, empiezan a rotar los colores de la gama en los cuatro cuadros puestos en fila. Rotan cada medio segundo, cambiando sucesivamente.
- Al pasar por encima de otro de los colores básicos se cambia la gama de colores.
- El botón "Comenzar Muestra", al activar la rotación, cambia de texto a "Parar Muestra" y permite detener la rotación, volviendo a la situación inicial.

El estilo que hace aparecer los cuadros es *visibility* con valor *visible*.

Los colores a usar vienen dados en el archivo .js en las variables *pantonXXXX* donde XXXX es el color principal.

***Nota**: Cuidado! para volver a activar un intervalo, hay que parar el anterior; y para volver a activar un evento hay que haberlo desactivado antes.*

Ejercicio 9. Listado de direcciones (P)

Para este ejercicio son necesarios los archivos *direcciones.html*, *direcciones.js* y *direcciones_datos.js*.

El ejercicio consiste en dos tareas:

1. A partir de los datos del archivo *direcciones_datos.js*, obtener en la variable *listaregistros* un conjunto de objetos que constan de tres campos: nombre, apellidos y direccion.
2. Al seleccionar cualquiera de los objetos radio, se muestra una tabla dentro del div de identificador *lienzo*, donde cada fila muestra el campo de los objetos elegidos (nombre, apellido, direccion o todas las columnas). No se muestran las filas con campos con valores nulos.

Notas:

1. Hay que utilizar las funciones enunciadas en *direcciones.js*
2. No se permite cambiar el fichero *direcciones.html*

Ejercicio 10. Destacar direcciones (P)

Para este ejercicio son necesarios los archivos *direcciones2.html* y *direcciones_datos.js*. Hay que completar el *direcciones2.js*. No es necesario haber realizado el ejercicio anterior.

Cada línea del fichero de datos es considerado como un registro. Se pide:

- Escribir, dentro del div con id *lienzo*, cada registro en un párrafo (<p>) separando nombre, apellido y dirección. No se escriben los registros con algún campo nulo.
- Activar eventos de forma que al pasar por encima de cada registro se cambie el color del fondo del párrafo. Al pasar a otro párrafo, el color vuelve a blanco.

Ejercicio 11. Listado de mascotas (P)

Para realizar los ejercicios se proporciona el fichero *listamascotas.js*, que contiene una lista de objetos de animales (nombre, raza, color, fecha de nacimiento).

1. Dibujar la tabla de datos

- Escribir una tabla con todos los datos de las mascotas que no contengan algún campo vacío.
- La tabla debe estar dentro del elemento div de identificador *lienzo*.
- La tabla contiene una cabecera con los nombres de los campos.

2. Selección de filas

A partir de los objetos de formulario (checkbox y date) se va a reescribir la tabla del ejercicio 1 con las mascotas que cumplen uno, o los dos, criterios de búsqueda.

- La búsqueda por raza se realiza al elegir cualquiera de ellas. Se escriben todos los animales que pertenecen a las marcadas.
- La búsqueda por fecha se hace al pulsar en la imagen de la lupa. Hay que escribir ambas fechas, si falta alguna se avisa al usuario y no se cambia la tabla.
- Si se pulsa la lupa sin haber elegido fechas, se realiza una búsqueda por las razas seleccionadas. Y si no hay ninguna, se escribe completa (sin vacíos).

3. Resaltado de filas

En este apartado se quiere que, al pasar el ratón por las filas de la tabla, la fila cambie de color de fondo. Al pasar a otra fila, vuelve a su color inicial.

4. Añadir estilos

Este apartado se puede realizar mientras se realizan los otros, o bien, implementarlo al final. Se quiere que

- La tabla tenga un borde de 1px, sólido
- Las filas vayan alternando de colores de fondo, las pares en *beige*, las impares en *bisque*.

Ejercicio 12. Listado de coches (P)

Para realizar los ejercicios se proporciona el fichero *listacoches.js*, que contiene una lista de objetos de coches (matricula, marca, color, motor).

1. Dibujar la tabla de datos

- Escribir una tabla con todos los datos de los coches que no contengan algún campo vacío.
- La tabla debe estar dentro del elemento div de identificador *lienzo*.
- La tabla contiene una cabecera con los nombres de los campos.

2. Selección de filas

A partir de los dos objetos de formulario (selección y texto) se va a reescribir la tabla del ejercicio 1 con los coches que cumplen uno, o los dos, criterios de búsqueda.

- La búsqueda por marca de coche se realiza al elegir una de ellas
- La búsqueda por matrícula se hace al ir introduciendo caracteres en el campo de texto. Se buscan matrículas que contengan la cadena escrita en cualquier posición de la matrícula.

3. Resaltado de filas

En este apartado se quiere que, al pasar el ratón por las filas de la tabla, la fila cambie de color de fondo. Al pasar a otra fila, vuelve a su color inicial.

4. Añadir estilos

Este apartado se puede realizar mientras se realizan los otros, o bien, implementarlo al final. Se quiere que

- La tabla tenga un borde de 1px, sólido
- Las filas vayan alternando de colores de fondo, las pares en *beige*, las impares en *bisque*.

Ejercicio 13. Sorteo de la primitiva (E)

Este ejercicio consiste en una simulación del sorteo de la primitiva obteniendo, y mostrando, la combinación ganadora.

El programa debe cumplir:

1. Crear el boleto en el lugar destinado (div boleto) con la disposición de los números tal como aparece en la imagen de inicio.
2. Activar el botón **Sortear** para que al pulsar se genere la combinación ganadora.
3. Al pulsar *Sortear*, los números ganadores aparecen con fondo rojo en el boleto y en la tabla de números premiados.
4. Si se vuelve a pulsar *Sortear*, aparecen los nuevos números desapareciendo el resultado anterior.

Ejercicio 14. Desactivar la cuenta atrás (E)

En el estado inicial de la página del ejercicio se muestra un cuadro que simula una bomba con un reguero de cuadros simulando una mecha.

Al pulsar el botón **Comenzar**

- El botón *Comenzar* se desactiva
- Comienza la cuenta atrás de **10 segundos**
- Aparece el botón *Desactivar*

En cada segundo sucede

- La cuenta atrás, que se muestra en el cuadro azul, disminuye en un segundo.
- Un cuadro de la mecha cambia su color a negro, simulando que ha ardido.

Al pulsar el botón **Desactivar** se vuelve al estado inicial:

- *Comenzar* en rojo y activo,
- la mecha y la bomba en sus colores iniciales (aparecen en el archivo .css)
- La cuenta atrás vuelve a 10
- El botón *Desactivar* se oculta

Al finalizar la cuenta atrás y llegar a 10 segundos

- La bomba cambia de color a *firebrick*
- Aparece el letrero de explosión (Boom!!!)
- Se para la cuenta atrás

Ejercicio 15. Descubrir el pin de una tarjeta (E)

El ejercicio simula la situación en que un usuario quiere visualizar el pin de su tarjeta bancaria, una vez introduce unas claves conocidas. En este caso hay que introducir la palabra **javascript** y seleccionar sólo el año **2012** para activar el botón **Mostrar Pin** (estas claves están en el archivo .js)

Hay que verificar el formulario que contiene los datos correctos

Al pulsar el botón **Mostrar Pin**

- Se levanta la *"cortina"* que oculta el pin
- Comienza una cuenta atrás de **5 segundos**

En cada segundo el cuadro azul muestra los segundos que quedan para ver el pin.

Al finalizar la cuenta atrás y llegar a 0

- La *cortina* vuelve a cubrir el pin
- Se para la cuenta atrás
- El cuadro azul queda en blanco

Si no se cumplen las claves del formulario en el cuadro azul (para los segundos) aparece el mensaje **Error**

Ejercicio 16. Votación del público (P)

A partir de los votos del público, se quiere visualizar el resultado de la votación, con los participantes en orden de mayor a menor número de votos.

Al mostrarlos en el navegador se quiere:

- El total de votos
- Los participantes, en orden decreciente de votos. Se muestra el nombre del participante y el número de votos.
- Una gráfico de barras horizontales que muestre el porcentaje de votos de cada participante.
- Hay que fijar el número de participantes (5 por ejemplo)

Opción A.

- El número de votantes es fijado por programa.
- El número de votantes es aleatorio entre un rango (p.ej.: entre 2000 y 3500 votantes). Se puede considerar que el resto hasta el máximo son votos blancos o abstención.
- El número de votos de cada participante es aleatorio
- Hay que tener en cuenta que, para cada participante, los votos posibles van decreciendo con los ya asignados.

Opción B.

Suponemos que cada participante es una lista de personas. Se quiere ver cuántos integrantes de cada lista consiguen plaza.

Además de las condiciones de la opción A, tenemos que

- Las plazas a asignar en total son 20.
- El número de votos para cada lista es aleatorio. Si estos votos son menos del 5% de los votos, no consiguen plaza.

17. Sorteo del Euromillón (E)

Este ejercicio consiste en una simulación del sorteo del Euromillón obteniendo, y mostrando, la combinación ganadora.

El programa debe cumplir:

1. Crear el boleto en el lugar destinado (div boleto) con la disposición de los números tal como aparece en la imagen de inicio **(atención a la última columna)**
2. Activar el botón **Busca tu Millón** para que al pulsar se genere la combinación ganadora.
3. Al pulsar *Busca tu Millón*, los números ganadores aparecen con fondo dorado *rgb(248, 218, 47)* en el boleto y en la tabla de números premiados.
4. Si se vuelve a pulsar *Busca tu Millón*, aparecen los nuevos números desapareciendo el resultado anterior.

18. Alineaciones de una Final Four (P)

Se quiere tener un cuadro de controles que muestren en el navegador los integrantes de los cuatro equipos que participan en una Final Four.

En una cabecera tenemos los nombres de los cuatro equipos. Debajo, en un cuadro a la izquierda, se ven los integrantes de los equipos; y al mismo nivel, a la derecha los datos de cada jugador.

Inicialmente, sólo se ven los nombres de los equipos y los cuadros de jugadores y datos de jugador en blanco.

- Al pasar el ratón por el nombre del equipo, se ven los nombres de los jugadores de ese equipo. Desaparecen al dejar de pasar por el nombre de equipo.
- Al pulsar el nombre de un equipo, los jugadores quedan fijados en el cuadro izquierdo.
- Al pasar el ratón por el nombre de un jugador (una vez fijados), aparecen sus datos en la derecha.
- Al pulsar el nombre de un jugador, quedan fijados los datos de ese jugador. Aparece un cuadro con el texto **Liberar**
- Al pulsar sobre *Liberar* volvemos a la situación inicial.
- Los datos de los equipos y de los jugadores se asignan por programa.