

Apellidos y Nombre:

Hora de inicio: 8:30

Duración: 6 horas.

NOTA: Para la realización del examen solo se puede consultar el codecar. No se puede consultar libros, ni apuntes, ni páginas web, ni ficheros de prácticas, ejercicios realizados, ni cualquier otro fichero que contenga código. A la hora de evaluar los ejercicios se tendrá en cuenta la optimización del código y la utilización correcta de los recursos que se pueden utilizar

- 1.-) A partir del fichero **ejercicio01.html**, vamos a realizar un proceso de validación de dicho formulario, dichos formularios no se van a enviar si se produce algún error en el formulario.(Se puntúa sobre 3 puntos).
 - a.-) El **documento HTML NO se puede modificar y NO se pueden utilizar DOM ni jQuery.**
 - b.-) Cuando se pulse el botón de "**Enviar**" vamos a llevar la validación. Se deberán de tener en cuenta las validaciones descritas más adelante. Vamos a mostrar los errores en las cajas de texto que hay a continuación de los elementos a evaluar y cuyo nombre empieza por "err" y pertenecen a la clase error.
 - c.-) Las cajas de texto de la descripción y nombre del artículo solamente van a admitir letras y el espacio.
 - d.-) Cuando nos situemos sobre las cajas de texto el color de los caracteres será azul y el color de fondo será amarillo, cuando abandonemos la caja de texto la visualización será la inicial.

Apellidos y Nombre:

e.-) Para las validaciones deberemos tener en cuenta los siguientes puntos:

i. Validaciones exhaustivas:

1. El **código de exportación** comienza con letras entre 3 y 6, le siguen 4 ó 8 dígitos, a continuación tiene un guión o el carácter "?", continua con tres caracteres que pueden ser "\$", "%", "& ", "*", "/", le siguen entre 2 y 8 caracteres entre letras o dígitos y termina por tres dígitos.
2. El **nombre del artículo** va a empezar por dos letras o números y va a terminar por una letra o el carácter "#", en medio puede tener letras, espacios, el guión, el punto y dígitos, con un número de caracteres comprendidos entre 7 y 13.
3. Las **unidades** va a ser un número entero positivo.

ii. Validaciones mediante expresiones regulares directas:

1. El **código interno** va a empezar por cuatro o siete dígitos a continuación va a tener de cinco a ocho letras, le va a seguir el carácter "\$" o bien el carácter "@", a continuación va a tener dos dígitos y va a terminar con seis caracteres, que pueden ser letras o dígitos.
2. La **descripción** va a empezar por seis letras, va a terminar por dos caracteres que pueden ser letras, dígito o coma y en medio va a contener letras,

Apellidos y Nombre:

números, el guion o los caracteres “\$”, “%”, “/”, con una longitud comprendida entre 13 y 37 caracteres.

3. El **valor** va a contener de cinco a siete dígitos con un valor mínimo de 50000.

iii. Validaciones mediante expresiones regulares del objeto RegExp.

1. El **código internacional** va a contener 10 o 17 letras.
2. Las **características** van a ser un número entero de entre 2 y 7 dígitos.
3. La **fecha de alta** va a contener un fecha correcta en la cual el día y el año van a tener dos dígitos y el año va a contener 2 o 4 dígitos. No se tienen en cuenta los años bisiestos.

iv. Se debe seleccionar un tipo de artículo.

v. Se deben seleccionar al menos dos ciudades donde hay fabricas.

vi. Se deben seleccionar al menos cuatro comunidades autónomas con tienda.

2.-) A partir del fichero **ejercicio02.html** realiza las siguientes acciones utilizando **DOM**. El **código HTML NO se puede modificar y los eventos se deben programar mediante el DOM para cualquier navegador. NO se pueden utilizar arrays para hacer este ejercicio, se debe trabajar directamente sobre los elementos que tenemos.**(Se puntúa sobre 1'5 puntos).

Apellidos y Nombre:

a.-) Cuando se pulse el botón **Poner**, deberemos comprobar que se ha introducido una marca y modelo de coche, se ha seleccionado una opción en la lista de número de unidades (no vale la opción (“–seleccione una opción–”, que tiene como valor la cadena vacía) y se ha marcado un botón de selección, en cuyo caso se deberán realizar las siguientes acciones:

i.-) Con los valores obtenidos vamos a añadir una fila a la tabla (que debe estar ordenada en modo ascendente por la marca y modelo del coche), siempre y cuando la marca y modelo del coche no esté incluido en la tabla. Si la marca y modelo del coche está incluido en la tabla haremos las siguientes acciones:

1. Modificaremos el número de unidades vendidas y el tipo de coche, poniendo los valores que hemos introducido.
2. Incluiremos la marca y modelo del coche en la lista que hay debajo (que debe estar ordenada en modo descendente), lista de marcas y modelos del coche modificados, siempre y cuando no exista previamente en la misma

3.-) A partir del fichero **ejercicio03.html** realiza las siguientes acciones utilizando **jQuery**. El **código HTML NO se puede modificar y los eventos se deben programar mediante jQuery**. (Se puntúa sobre 1'5 puntos).

Apellidos y Nombre:

a.-) Cuando pulsemos en el botón “Incluir individuos” vamos a coger el contenido de las cajas de texto y si no están vacías:

i.-) Vamos a incluir los datos en la tabla (ordenado por familia en orden ascendente y cuando coincida la familia por subfamilia en orden descendente); siempre y cuando no existe un animal en la tabla con la misma familia y subfamilia, en cuyo caso (si existe) sumaremos el número de individuos de la caja de texto al valor que tenemos en la tabla.

ii.-) Además en todos los casos deberemos actualizar los valores que tenemos en el pie de la tabla.

4.-) A partir del fichero **ejercicio04.html** realiza las siguientes acciones **Utilizando jQuery y utilizando el método \$.ajax**. El código HTML y PHP NO se puede modificar y los eventos se deben programar mediante jQuery (Se puntúa sobre 1 puntos)

a.-) Cuando pulsemos el botón “**Realizar cálculo**” vamos a obtener el valor de las cajas de texto de número de caras y vértices de un poliedro. Si no están vacías vamos a crear un objeto **FormData** con los campos **numcaras** (número de caras) y **numvertices** (número de vértices).

b.-) Se va a llamar mediante el método \$.ajax de jQuery al programa **ejercicio04.php** pasandole el objeto **FormData** creado en el apartado anterior.

Apellidos y Nombre:

c.-) Cuando nos devuelve la solicitud, nos devolverá un valor que vamos a mostrar en la caja de texto del número de aristas del poliedro.

5.-) A partir del fichero **ejercicio05.html** realiza las siguientes acciones **DOM y el método fetch**. El código HTML y PHP NO se puede modificar y los eventos se deben programar mediante el DOM para cualquier navegador.: (Se puntúa sobre 1 puntos)

a.-) Cuando se pulse el botón “**Solucionar ecuación**” vamos a comprobar que las cajas de texto tienen valor, en cuyo caso crearemos una cadena XML, en donde el nombre del dataset es **ecuacion**, el nombre del registro **segundo** y los nombre de los campos son **a**, **b** y **c**.

b.-) Llamaremos al programa **ejercicio05.php**, pasándole los datos en XML, utilizando el método **fetch**.

c.-) Nos va a devolver una cadena XML con las soluciones a esa ecuación. Los nuevos campos son **sol1** y **sol2**. Y mostraremos los resultados en las cajas de texto correspondientes.

6.-) A partir del fichero **ejercicio-06.html** realiza las siguientes acciones Utilizando el método **ajax puro y el DOM**. El código HTML y PHP NO se puede modificar y los eventos se deben programar mediante el DOM para cualquier navegador.: (Se puntúa sobre 1 puntos)

a.-) Cuando se pulse el botón “**Obtener distancia**” vamos a comprobar que las cajas de texto tienen valor, en cuyo caso

Apellidos y Nombre:

crearemos un objeto con las propiedades **Velocidad** y **Rozamiento**.

b.-) Llamaremos al programa **ejercicio06.php** pasándole los datos en JSON.

c.-) Nos va a devolver un objeto JSON con la distancia recorrida.

La nueva propiedad del objeto es **Distancia**. Y mostraremos los resultados en la caja de texto correspondiente.