

Apellidos:
Nombre: UVUS:



Bases de datos (GE/DGME)
Cuarta prueba de evaluación alternativa

24/05/2023

Ejercicio 1 (1 punto) Partiendo del archivo `ejercicio1.html` que se le proporciona, adapto para que aplique los siguientes mecanismos empleando Bootstrap 5:

- (0,6) Sin necesidad de alterar la estructura general del documento, incorpore los elementos y clases necesarias para que, en lugar de disponer de simples enlaces que mueven la página hacia arriba o hacia abajo, emplee alguno de los mecanismos de navegación vistos en clase (barra, pestañas o botones/píldoras), de modo que siempre estemos visualizando la opción seleccionada, quedando oculto el resto.
- (0,4) Emplee los elementos necesarios para disponer los controles de formulario en filas y columnas bien alineadas, como en la imagen de la figura (ignore la estética de los elementos):

Controles de formulario HTML5

Nombre:	<input type="text"/>	Correo Electrónico:	<input type="text"/>
URL:	<input type="text" value="Escribe la URL de tu página web person."/>	Fecha:	<input type="text" value="dd/mm/aaaa"/>
Patrón	<input type="text"/>	País:	<input type="text"/>
		<input type="button" value="Insertar"/> <input type="button" value="Actualizar"/> <input type="button" value="Eliminar"/>	

En el caso de que el dispositivo sea pequeño, debería verse todo en una columna.

- (0,3 opcional, solo para compensar errores en apartados anteriores) (0,05) Aplique alguna clase de Bootstrap para separar los elementos de los bordes laterales del navegador. (0,1) Aplique los cambios necesarios a la tabla para que pase a tener la estética de las tablas de Bootstrap, incluyendo la alternancia de colores en las filas y el resaltado al pasar sobre ellas. (0,15) Aplique las clases necesarias a los controles y botones, para que adopten la estética de Bootstrap.

Ejercicio 2 (1 punto) Partiendo del archivo `ejercicio2.html` que se le proporciona, aplicar una hoja de estilo generada a partir de código SASS, que incorpore los siguientes estilos (aplicando de forma adecuada los mecanismos de SASS vistos en el tema, empezando por el anidamiento, cada vez que pueda):

- (0,1) El elemento genérico de bloque cuyo id sea “container” debe tener un borde negro con línea continua de 2 píxeles. Además, debe tener un relleno de 20 píxeles.
- (0,1) Los párrafos hijos directos de dicho elemento deben aparecer en *cursiva*.
- (0,1) Los divs descendientes del *container* deben tener un borde negro con línea discontinua, con el mismo ancho que los bordes anteriores.
- (0,2) Los párrafos descendientes de los anteriores deben tener su texto en rojo, y su primera línea debe estar en **negrita**.
- (0,2) Si no lo ha hecho hasta el momento, adapte el código anterior para incorporar variables para aquellos valores que aparecen recurrentemente.

- f) (0,3) Incorpore algún otro mecanismo parametrizado para generar los bordes, con posibles parámetros con valores por defecto, y de modo que permitan indicar además mediante algún parámetro un posible redondeo al borde. En caso de aplicarse el redondeo, añadir un relleno al elemento. Pruebe este efecto en alguno de los bordes usados.

El resultado debe quedar de forma similar a lo siguiente:

Lorem ipsum dolor sit amet consectetur adipisicing elit. Cumque laboriosam, quidem, delectus voluptates temporibus nisi eius adipisci exercitationem dolor ipsam autem repellendus rerum beatae aspernatur quos, neque ea harum voluptatibus. Dicta porro et deleniti, vel fuga ducimus, fugiat libero dolorum eius aperiam impedit possimus quae? Reiciendis velit eveniet rerum qui.

Deserunt debitis repellat repudiandae quia assumenda a, harum est deleniti nostrum ipsum sit accusantium voluptates cupiditate nam voluptatum, fugit quo. Maiores nemo illum itaque porro debitis aliquid quibusdam reiciendis, ea quis blanditiis laborum perferendis sint dolorum sequi ipsa magnam. Odio, sint culpa. Inventore culpa eos atque quidem nihil voluptatem officiis.

Quisquam provident reiciendis quibusdam, quaerat ratione optio. Fugit alias dolorem minus voluptatem consequuntur temporibus doloremque obcaecati quaerat accusamus eum in, cum nam totam facere aliquid dolores adipisci voluptatibus optio pariatur veritatis debitis? Culpa corporis id ipsa, voluptatum dicta officia voluptatem. Error, itaque! Maxime eos, similique inventore facilis quaerat non odio.

Voluptate omnis dolorem nisi tenetur. Culpa ullam iure beatae, assumenda, sequi voluptatem at iusto incidunt qui debitis quisquam aliquid! Cum repudiandae praesentium obcaecati neque illum esse necessitatibus. Nostrum voluptatem quia voluptates natus quaerat asperiores vel quidem officiis, est magni dicta tempora animi soluta sunt voluptas, delectus accusantium? Laudantium, minus beatae.

Similique ullam iure corrupti libero sint. Eveniet odio facere, necessitatibus id sequi iusto? Iure fugiat repellendus eaque, quis fuga officiis praesentium et consectetur. Molestiae nihil eligendi labore a iure? Nobis, doloremque! Autem est animi, dicta cumque vel perspiciatis sequi voluptatem voluptates ratione corporis ex laborum perferendis ab quo non dignissimos.

Facere velit tempora laborum molestias unde asperiores dolores minus ullam quasi nobis, vero ad perspiciatis enim magnam eos quam? Dicta magnam sed repudiandae eos molestiae quisquam aliquid perspiciatis impedit excepturi asperiores, earum architecto consequatur tempore aperiam quis doloremque cumque numquam, ratione sequi? Corporis magnam dicta, pariatur nisi cumque veritatis et!

Minus eligendi voluptas pariatur quasi dolorum ullam provident quidem velit rerum eos aliquam consectetur sint exercitationem sequi, consequatur aspernatur eaque fugit in eius, non molestias accusantium vel reprehenderit. Facere alias explicabo asperiores consectetur, ad omnis? Placeat dolore illum voluptatum quisquam, porro, dolorum magnam sit voluptate repellendus a quam doloremque odit.

Suscipit, dolores? Sequi fugiat nobis officia minus debitis esse omnis aliquid aperiam animi beatae ex temporibus minima nesciunt et voluptas deleniti vero culpa aliquid similique, assumenda maxime velit! Laborum quis fugiat possunt placeat? Molestiae, id asperiores dolores cupiditate

Ejercicio 3 (1,25 puntos) Partiendo del archivo `ejercicio3.html` que se le proporciona, hacer que al cargar dicha página, ejecute dinámicamente un código que haga lo siguiente:

- (0,25) Solicitar al usuario un número entre el 5 y el 10, con valor por defecto el 5, y hacer lo siguiente:
 - (0,35) Si el número es menor que 8, deberá obtener el producto de los números desde el 4 (incluido) hasta el número indicado.
 - (0,3) En caso contrario, deberá obtener la suma de los números desde el indicado hasta el 11 (incluido).
- (0,35) Tras realizar la operación anterior, deberá pedir confirmación al usuario sobre si se debe mostrar el resultado anterior por mensaje emergente. En caso de rechazarse, se deberá imprimir por consola.

Ejercicio 4 (1,75 puntos) Partiendo del archivo `ejercicio4.html` que se le proporciona, incorporar el código dinámico en cliente para hacer lo siguiente:

- (0,5) Asignar un manejador de evento al select de modo que, al cambiar su valor, asigne al div la clase 1 o la 2 (que encontrará ya definidas en la página), según la opción seleccionada.
- (0,75) Asignar un manejador de evento al botón de modo que, al pulsarlo, añada al principio de la lista el texto indicado en la entrada de texto.
- (0,5) Asignar un mismo manejador de evento a cada uno de los párrafos, de forma que, al pasar sobre cada uno de ellos, muestre en mensaje emergente el contenido en texto del párrafo.

Ejercicio 5 (2 puntos) Partiendo del archivo `ejercicio5.html` que se le proporciona, y tomando inspiración en la siguiente imagen (tomada de [esta web](#)), incorporar el código dinámico en cliente para hacer lo siguiente:

G	Cazasubmarinos
1	Hawker Siddeley Nimrod MR Mk.1
	
Velocidad	926 Km/h
Autonomía	8.340 Km
Potencia	22.160 Kp.
Altura techo	9.600 m.
Longitud	38,63 m.
Envergadura	35,— m.

H	Helicóptero de combate
1	Sikorsky S-67 Blackhawk
	
Velocidad	350 Km/h
Autonomía	1.356 Km
Potencia	3.000 CV
Altura techo	1.980 m.
Longitud	19,56 m.
Rotor Ø	18,90 m.

- (0,3) Crear un array de objetos simples de JavaScript (sin clases), almacenando los datos de algunos vehículos voladores, cada uno con los datos que necesite.
- (0,6) Crear una función que reciba un array de vehículos y dos números, y muestre en el cuerpo de la página un div conteniendo un párrafo con los datos principales de aquellos vehículos cuya velocidad se encuentre entre los dos números indicados.
- (0,1) Llamar a la función anterior, pasándole el array creado.
- (1) Crear una variante de la página creada en los apartados anteriores pero basada en clases. Se deberá distinguir los datos generales de cada vehículo volador de aquellos datos específicos de los aviones (envergadura) o los helicópteros (rotor). Además, se valorará el empleo efectivo de algún atributo privado, getter y setter.

Ejercicio 6 (3 puntos) Cargue en su servidor local MySQL la base de datos `jardineria.sql`. Partiendo de los archivos HTML y PHP que se le han proporcionado, realice las adaptaciones necesarias en la dinámica de cliente y servidor para páginas PHP para disponer de las siguientes funcionalidades:

- a) Desarrollar una función que realice una petición GET al servidor para recuperar todas las oficinas, enviándolas al PHP de vuelta en formato JSON. Al volver al cliente, generar las correspondientes opciones en el select de oficinas, asignando como valor el código de oficina y como texto de la opción la ciudad. Al cargar la página, se debe hacer una llamada a esta función. Indicar mediante comentarios en el código los cambios que implicaría en el cliente y el servidor si deseáramos devolver los datos del servidor al cliente mediante HTML, en lugar de JSON.
- b) Desarrollar una nueva función que tome el valor actual del select de oficinas y realice una petición POST al servidor, enviándole los datos al servidor en formato JSON, para solicitarle los empleados correspondientes a la oficina seleccionada. El servidor deberá devolver un objeto JSON con varios campos, incluyendo las filas recuperadas de la consulta a la base de datos, otro campo con el número de filas, y otro con un atributo booleano con valor TRUE (puede hacer pruebas luego poniéndolo a FALSE). Una vez recibida la respuesta del servidor, se deberá vaciar la lista de empleados. A continuación se deberá comprobar que el valor del atributo booleano es cierto y que el número recibido es mayor que 0. En caso afirmativo, deberá volver a generar los items correspondientes a los empleados. En caso contrario, se deberá mostrar un mensaje emergente indicando que no se ha podido completar la recuperación de los empleados. Garantice que esta función es llamada por primera vez una vez hayamos recuperado las oficinas. Además, se deberá asignar al select un manejador de evento, de modo que, al modificar la oficina seleccionada, se llame a la misma función creada.