

Apellidos:

Nombre: UVUS:



Bases de datos (GE/DGME)

08/06/2022

Segundo parcial oficial - Parte 1

Cuestión 1 (1 punto) ¿Cómo es el proceso desde que escribimos una dirección en un navegador hasta que recibimos la respuesta? Ilustre gráficamente los pasos y descríbalos brevemente.

Ejercicio 2 (5 puntos) Desarrolle el código HTML correspondiente a los siguientes apartados (con las páginas que necesite, mientras no haya funcionalidades en conflicto):

- a) (1) Un elemento genérico de bloque con una clase contenedor. Dentro del mismo, además de una cabecera de segundo nivel con un identificador, incorporar un par de párrafos, con texto generado automáticamente. Incluya dentro del texto algunos enlaces: uno a un servidor externo, otro con una dirección relativa dentro de la carpeta donde tiene el archivo y otro con una dirección absoluta a partir de la raíz de su servidor.
- b) (1) Una lista ordenada con las personas célebres a quienes más admire. Para cada una de ellas, proporcionar una lista no ordenada con los valores o las cualidades que le llevan a incluirlas en esta lista. Indique los nombres de las personas como importantes, y las cualidades como destacadas.
- c) (1) Una tabla que muestre información sobre asignaturas con sus siglas, nombre completo, curso y número de créditos. La tabla debe tener todas sus secciones principales, filas con campos de cabecera resaltados y algunas filas de datos con celdas simples. Además, incluya al final en el pie una fila con una primera celda "Total" que ocupe todas las columnas menos la última (que debe mostrar el número de filas).
- d) (1) Un formulario para almacenar asignaturas, con la información de la tabla anterior. El formulario debe enviar datos mediante una petición POST a cierta página php de su servidor. Debe incluir etiquetas de texto asociados a los controles, controles básicos apropiados para los campos indicados y un desplegable para seleccionar el curso.
- e) (1) Incorpore elementos de lo que denominamos HTML semántico, para marcar adecuadamente distintos tipos de apartados de su página, elementos para interconectar partes de la misma, etc. Explique brevemente el cometido de los elementos empleados.

Ejercicio 3 (4 puntos) Desarrolle el código CSS necesario para lo siguiente:

- a) (0,75) Asignar al elemento con identificador “coloreado” un color de fondo amarilla y letra negrita. Además, a todo elemento de clase “principal”, asignarle un borde continuo azul de 1 píxel, ligeramente redondeado.
- b) (0,75) Aplicar a los párrafos descendientes de un elemento “div” que tenga la clase ‘contenedor’ un espacio interior que sea cinco veces el ancho de la letra M. Modifique la manera de interpretar el tamaño del elemento, de modo que en lugar de considerar solo el contenido se incluya también en las dimensiones (ancho y alto) el espacio interior y el grosor del borde.
- c) (0,75) Asigne a todos los párrafos hermanos posteriores de otro párrafo la mitad del ancho disponible (a ser posible, con unidad relativa dependiente del espacio visible actual de la página) y céntrelos horizontalmente.
- d) (0,75) Dejar a todo elemento de tipo celda de una tabla (resaltados o no, con selector múltiple) un espaciado exterior empleando una medida absoluta. Cambie además el color de fondo de la última fila del cuerpo de la tabla.
- e) (0,5) Ocultar todo elemento de clase “ocultar” cuando pasemos con el ratón sobre él, de modo que se mantenga sin ocupar el espacio que ocuparía el elemento.
- f) (0,5) Aplique un estilo a un div cuyo id sea “barra-fija”, de modo que este elemento quede posicionado en la parte superior de la pantalla, separada 2 píxels de la parte superior y 3 del lado izquierdo, y que permanezca visible independientemente de que hagamos scroll hacia abajo de la página y de que dicho elemento se encuentre en su código HTML en una posición más baja

Apellidos:
Nombre: UVUS:



Bases de datos (GE/DGME)
Segundo parcial oficial - Parte 2

08/06/2022

Ejercicio 1 (3,25 puntos) Aplique las técnicas necesarias en una web para dotarlo de los siguientes elementos haciendo uso de los mecanismos empleados en clase:

- a) (1,25) Haciendo uso de CSS Grid, disponer en una página una sección de 200px de ancho, y otras dos que ocupen todo el espacio restante, siendo ambas del mismo tamaño, independientemente de si cada elemento contenga márgenes, espaciados internos o bordes en sus elementos. Además, las secciones deberán separarse en algunos píxeles. Dentro del grid, incluya en el elemento de la izquierda un contenedor con una serie de botones dispuestos verticalmente, empleando Flexbox. En las dos secciones restantes, incorpore una serie de párrafos generados automáticamente.
- b) (1) Haciendo uso de Bootstrap 5, crear una página con una barra de navegación interna, de modo con cada opción visualizando la sección de la página que corresponda y se oculte el resto, conteniendo cada sección algún contenido mínimo para poder probar que todo funciona correctamente. Aparte de la navegación, ponga algún ejemplo de código aplicando elementos de estilo propios de Bootstrap a componentes de una web, preferentemente a tablas, formularios, textos o tarjetas.
- c) (1) Cree una nueva nueva página que defina su estilo aplicando SASS. Defina un *mixin* al que le pase parámetros (con valores por defecto incluidos) relacionados con color de fondo, características de borde y espaciado interno, de forma que asigne los parámetros a propiedades CSS adecuadas. Llame al mixin desde distintas reglas, con una llamada por defecto y otra pasando valores a la llamada. Emplee anidamiento dentro de SASS, incluyendo distintas reglas anidadas que afecten a hijos, a descendientes y al elemento actual pero incluyendo alguna pseudoclase.

Ejercicio 2 (6,75 puntos) Desarrolle el código necesario para crear lo siguientes elementos estáticos y dinámicos en una web:

- a) (1,5) Hacer una página tal que, al cargarla, ejecute dinámicamente un código que haga lo siguiente:
 - Generar un número aleatorio, empleando la función `Math.random()`. Dependiendo de su valor:
 - Si este es mayor que 0.5, deberá solicitar al usuario un número natural. A continuación, calculará la suma de los números naturales hasta ese número. Por último, emitirá un mensaje emergente indicando el cálculo realizado. Ejemplo: si el número suministrado es el 6, se deberá mostrar algo así como “La suma de los números naturales hasta el 6 es 21”.
 - Si el número es menor o igual que 0.5, deberá calcular el producto de los números del 1 al 10 y sacarlo por consola.
- b) (2) Cree una nueva página que contenga un control de número (con valores posibles entre 1 y 3), y un contenedor con dos párrafos con el texto que quiera, todos teniendo la misma clase html. La página debe hacer lo siguiente (en un código Javascript asociado a la página, en un archivo aparte):

- Definir una función que no reciba parámetros de entrada y, en función de su valor:
 - Si el valor es 1, crear en la página un nuevo párrafo con el texto que quiera, y con la misma clase que incluyó en los párrafos anteriores.
 - Si el valor es 2, eliminar el primer párrafo de la clase atribuida a sus párrafos.
 - Si el valor es 3, añadir a sus párrafos una copia de alguno de sus párrafos existentes.
- Hacer que esta función se ejecute por primera vez cuando se cargue la página.
- Asignar un manejador al evento de cambiar el valor del control anterior, de modo que se tome la opción seleccionada y llame a la función anteriormente creada.

c) (2,25) Vamos a trabajar con arrays y objetos básicos de JavaScript. Haga lo siguiente en una nueva página:

- Cree un array con una lista de cadenas denominada `aficiones`.
- Cree una función que reciba un array `a` y un número `n`, y devuelva las primeras `n` aficiones contenidas en el array.
- Cree en HTML un control de formulario que permita seleccionar un número natural, y un botón al que asociar un manejador de evento que haga lo siguiente:
 - Recuperar el valor del control de formulario.
 - Obtener un array con aficiones que igualen o superen el número de créditos indicado en el control de formulario. Tenga presente que ya debe disponer del objeto global `asignaturas` y el valor anterior, y que con la función definida anteriormente puede filtrar la lista de asignaturas.
 - Recorrer el array obtenido y generar una lista HTML con las aficiones que contenga.
 - Además, para la primera de las aficiones seleccionadas, crear un objeto básico que contenga su nombre y el número de caracteres (recuerde la propiedad `length` para cadenas) y que imprima por consola este objeto.

d) (1) Haga lo siguiente en una nueva página, en relación con clases y objetos:

- Cree una clase de JavaScript para representar películas, con atributos para indicar el título, año, los nombres de l@s director@s y l@s principales intérpretes. Incluya un constructor con parámetros adecuados para recopilar esos datos. Nota: aunque no es necesario saberlo en el constructor, comentar que cada intérprete o director/a sería un objeto básico conteniendo nombre y nacionalidad.
- Incluya un método en la clase que indique si hay más de un director.
- Incluya otro método en la misma que permita añadir un/a nuevo/a intérprete a los ya existentes.
- Cree alguna instancia de película y código que emplee los métodos creados. Finalmente, imprima un mensaje en la consola mostrando la información principal contenida en el objeto.

Apellidos:

Nombre: UVUS:



Bases de datos (GE/DGME)

08/06/2022

Segundo parcial oficial - Parte 3

Ejercicio 1 (3,25 puntos) Cree páginas .php para resolver los siguientes apartados:

- a) (0,75) Cree una página en el servidor, incluyendo una mezcla de código HTML estático y código PHP dinámico que devuelva partes de la página resultante. En la parte dinámica, cree variables para almacenar el nombre de un libro, su número de páginas, e indicar si tiene tapa dura o no. Prepare una cadena mezclando texto, HTML y valores de las variables para imprimir a la salida el nombre del libro y su número de páginas. Además, si el número de páginas es mayor que 500 y la tapa es dura, deberá imprimir también a la salida un texto que indique que el libro debe ser bastante pesado.
- b) (1,25) Crear un array para almacenar series, de modo que de cada serie se almacene tanto su nombre como el número de temporadas que tiene. Recorrer este array e imprimir, para cada una de las series, un texto que indique su nombre. Además, aprovechar el recorrido para ir calculando la suma del número de temporadas de todas las series incluidas en su array. Imprimir posteriormente un nuevo párrafo de texto que indique cuál es la cantidad de temporadas que suman las series de su array.
- c) (1,25) En un archivo distinto, crear una función que, dado un array de series, lo recorra para encontrar la serie con más temporadas. La función deberá devolver el array asociativo que constituya esta serie más extensa. Llamar a esta función pasándole un array de series e imprimir un objeto JSON con el contenido devuelto por la función.

Ejercicio 2 (2 puntos) Ayudándose del archivo proporcionado `ej2.html`, crear el código necesario para resolver lo siguiente:

- a) (0,5) Completar el formulario para que envíe directamente los datos del formulario a un nuevo archivo php, haciendo uso de un tipo de botón adecuado, sin emplear JavaScript. La petición a realizar debe ser de tipo POST.
- b) (1,5) En el servidor, crear el código necesario para recibir los datos proporcionados por el formulario (texto 1, texto 2 y tipo de salida) y procesarlo dependiendo del tipo de salida indicada. Si el tipo de salida es HTML, deberá imprimirse una lista html con dos ítems, uno con el texto 1 y otra con el texto 2. Si es de tipo JSON, deberá generar un array asociativo con dos campos, uno para cada texto, e imprimirlo a la salida con el formato adecuado.

Ejercicio 3 (4,75 puntos) Ayudándose del archivo proporcionado `ej4.html`, y tras importar en phpMyAdmin la base de datos de coches proporcionada, crear el código necesario para resolver lo siguiente:

- a) (2) Preparar el código necesario tanto en JavaScript como en PHP para que, al cargar la página HTML, se llame al servidor mediante una petición GET para solicitarle los nombres de las ciudades (disponibles en la tabla CIUDAD de su base de datos). La parte de servidor deberá recibir la petición GET, conectar con la base de datos, recuperar todos los nombres de ciudades mediante una consulta no preparada, e imprimir el código HTML que genere las opciones del control desplegable de ciudades. De vuelta al código JS, se deberá tomar el texto HTML devuelto por el servidor y cargarlo dentro del elemento del DOM correspondiente al control desplegable, de forma que el usuario de la página visualice en el desplegable las ciudades existentes en la BD.
- b) (2,75) Completar el formulario para que envíe, a través de JavaScript, una petición POST al servidor al hacer click sobre un nuevo botón del formulario. La petición deberá ser asíncrona, como hemos visto en clase, y enviar los datos del formulario de la forma que estime conveniente. En el servidor, se deberán recibir los datos del formulario, conectar con la base de datos de coches, e insertar el registro en la tabla MARCA mediante una consulta preparada. Tras la misma, se deberá devolver un objeto JSON con un objeto que incluya un campo por cada dato que recibió de la petición, un dato lógico indicando que la transacción tuvo éxito y un mensaje indicando que “Todo fue Ok”. Terminar el código del cliente para que, una vez recibida la respuesta del servidor, se extraiga el mensaje recibido del servidor y se visualice en algún elemento del DOM.