

DWEC - Examen 18 de noviembre de 2022

Contenido

Instrucciones	1
Ejercicios.....	2
Ejercicio 1	2
Ejercicio 2	3
Ejercicio 3	4
Ejercicio 4	5

Instrucciones

Lee atentamente las siguientes indicaciones.

Te **recomiendo encarecidamente** que leas el ejercicio completo antes de hacer los distintos apartados indicados, ya que te dará ideas de cómo plantear la solución.

Se te ha facilitado una estructura de carpetas con todos los documentos.

Los documentos **HTML** solo podrás modificarlos para indicar tu nombre y apellidos donde se indica (borra el texto "**Nombre y apellidos**" y pon tus datos) y para añadir las referencias a los archivos JS que corresponda.

NO AÑADAS LIBRERÍAS. Si quieres usar alguna función extra añádela en el JS que corresponda.

Para la realización de este examen dispones de **las tres horas de clase**, y el método de entrega es a través de Moodle. Modifica el nombre de la carpeta a:

DWEC.Ex1.apellido.nombre

Y comprímela a un archivo ZIP, RAR o 7z con el mismo nombre

Por ejemplo, para Antonio Sierra, el archivo comprimido (y el de la carpeta) será:

DWEC.Ex1.sierra.antonio

ES IMPORTANTE QUE, a la hora de entregar, guardéis los cambios, le deis a **Enviar tarea** luego confirméis que queréis enviar el trabajo para su evaluación (ver la imagen siguiente)

☒ Confirmando que este trabajo es de elaboración propia, excepto aquellas partes en las que haya reconocido la autoría de la obra o parte de ella a otras personas.

¿Está seguro que quiere enviar su trabajo para que sea evaluado? Una vez enviado ya no podrá realizar modificaciones.

Continuar

Cancelar

En este formulario hay campos obligatorios

Y confirmar que sale el estado de entrega como enviado para calificar:

Estado de la entrega	Enviado para calificar
----------------------	------------------------

Si no tienes claro algo del enunciado, pregunta. Para la corrección de este examen será **obligatorio** usar los nombres de funciones y parámetros que se indican.

La puntuación será sobre 10 y la de cada ejercicio es la siguiente:

Ejercicio	1	2	3	4
Puntuación Máxima	2,5	2,5	2,5	2,5

Se valorará todo el código, pero en caso de que alguna función o elemento no funcione como se indica se usará el **modificador x0,6** al valor de éste.

Ejercicios

Ejercicio 1

En el HTML vas a encontrar un input de tipo texto (*inOpcion*), un botón (*btnOrdenar*) y una división de salida (*divResultado*).

En el JS encontrarás ya un par de colecciones, una con nombres y otra con números, que no puedes modificar, y la asignación del load del Windows que tampoco puedes modificar.

En la función *manejadorClick* escribe el código necesario para que la función haga lo que se pide y además desarrolla las funciones que se indican en una tabla más adelante.

Funcionamiento de la aplicación

El usuario podrá escoger unas de las siguientes opciones, escribiendo en el input ésta, y en función de cuál escoja se ordenará el array correspondiente con la función sort apropiada y mostrará el resultado por la división de resultado.

Procura que cada vez que se dé al botón de ordenar se borre lo que hubiese escrito antes en la división de resultado.

Vamos a dar por hecho que el usuario es capaz de introducir correctamente una de las siguientes opciones, ya sea las letras mayúsculas A,B y C o los números 1 y 2, por lo que no vamos a verificar la entrada.

Opción	Descripción
A	Ordenará el array de nombres de manera alfabética pero descendientemente.
B	Ordenará el array de nombres de manera alfabética, pero teniendo en cuenta <u>el tercer carácter</u> de cada elemento de la colección.
C	Ordenará el array de nombres de manera alfabética, pero teniendo en cuenta <u>el último carácter</u> de cada elemento de la colección.
1	Ordenará el array de números de manera descendente y dando prioridad a los que son divisibles por 5.
2	Ordenará el array de números de manera ascendente, pero dando prioridad a aquellos cuyo segundo dígito sea un 3.

Funciones a desarrollar

Función	Recibe	Devuelve	Descripción
---------	--------	----------	-------------

manejadorClick			Implementa la lógica de la aplicación como tú creas conveniente.
ordenarA	a,b	number	Será la función sort de la opción A
ordenarB	a,b	number	Será la función sort de la opción B
ordenarC	a,b	number	Será la función sort de la opción C
ordenar1	a,b	number	Será la función sort de la opción 1
ordenar2	a,b	number	Será la función sort de la opción 2

Ejercicio 2

En el HTML vas a encontrar un input de tipo texto (*inEntrada*), un botón (*btnComprobar*) y una división de salida (*divSalida*).

En el archivo JS encontrarás la función *manejadorClick* a partir de la cual has de generar la aplicación que se pide a continuación.

Funcionamiento de la aplicación

El usuario podrás escribir lo que quiera en el input de entrada y al dar al botón *btnComprobar* la aplicación le indicará cierta información de la entrada:

- Si no escribe nada le dirá que no ha introducido nada.
- Si escribe cualquier cosa le indicará si el dato es un texto, un número entero o decimal
- En el caso de ser texto, además, indicará:
 - La longitud del texto
 - El número de vocales que contiene, mayúsculas o minúsculas (ignora las que tengan acento)
 - Si contiene espacios en blanco y el número de ellos que contiene
 - Si contiene comas y el número de ellas que contiene
- En el caso de ser un número entero, además, indicará:
 - Por cuál de todos los números enteros desde 1 hasta uno menos que el mismo es divisible. Por ejemplo, si el número es el 6, comprobará si es divisible entre 1, 2, 3, 4 y 5.
 - Si es un número primo
- Si es un número decimal, además, indicará:
 - El valor de la parte entera
 - El valor de la parte decimal
 - El entero superior más próximo

Procura que cada vez que se pulse el botón para una nueva comprobación se limpie la división de salida de cualquier dato que hubiese previamente.

Ejercicio 3

En el HTML no vas a encontrar nada.

En el archivo JS tampoco vas a encontrar nada.

En este ejercicio has de crear una única función que se describe a continuación.

Nombre	generarMatriz
Argumentos de entrada	Por defecto no tendrá definido ningún argumento, pero podrá aceptar 2 o 3 argumentos de entrada.
Salida	Devolverá la matriz que genere
Funcionamiento	<p>Esta función generará matrices (arrays de distintas dimensiones) en función de lo que se introduzca como argumento de entrada.</p> <p>Si el usuario no introduce ningún argumento, la función, en vez de devolver la matriz, devolverá el string "Error: 0 argumentos".</p> <p>Si el usuario introduce un solo argumento, independientemente del valor de éste, en vez de devolver la matriz, devolverá el string "Error: 1 argumento".</p> <p>Si el usuario introduce dos argumentos o tres:</p> <ul style="list-style-type: none">• Tanto si son dos como si son tres argumentos, verificará que los dos primeros argumentos son números enteros mayores o iguales a 1, y han de ser Number, no valen Strings. Si no lo son, no hará nada más y en vez de devolver la matriz, devolverá el string "Error: no números enteros".• Si hubiese un tercer argumento, se verificará que es un string de longitud 1. Si no lo es, no hará nada más y en vez de devolver la matriz, devolverá el string "Error: separador erróneo".• Si los dos primeros argumentos son números enteros mayores o iguales a 1 creará una matriz de dos dimensiones usando como tamaño de las dimensiones dichos argumentos. El contenido de cada elemento de la matriz será un string con las coordenadas de dicho elemento, usando como separador el tercer argumento, que de no ser asignado se usará por defecto el separador "x". <p>Por ejemplo, si para acceder al elemento tengo que usar [2][3], dicho elemento será un string tal que "2x3", si no hubiese tercer argumento. Si se hubiese indicado como tercer argumento un "-", el elemento anterior hubiese sido "2-3".</p>

Por poner un par de ejemplos, te voy a tratar de mostrar el resultado de hacer un par de llamadas con los argumentos correctos:

Llamada	Return
generarMatriz(2,3)	<pre>[["0x0" "0x1" "0x2"] ["1x0" "1x1" "1x2"]]</pre>
generarMatriz(2,3,".")	<pre>[["0.0" "0.1" "0.2"] ["1.0" "1.1" "1.2"]]</pre>

Si escribes código para hacer verificaciones, bórralo antes de entregar el ejercicio.

Ejercicio 4

En el HTML vas a encontrar una división (*divSalida*) donde mostrarás todos los mensajes necesarios.

En el archivo JS encontrarás el manejador del load, pero completamente limpio. Tienes libertad absoluta para modificar este load y añadir las funciones que más te gusten.

Funcionamiento de la aplicación

Deberás generar un número entero aleatorio entre 2 y 6, ambos inclusive y sacar por la división de salida un mensaje que ponga:

“Vas a tener que esperar X segundos”

Donde la “X” será un contador desde el número obtenido antes hasta 0. Obviamente, dicho contador se actualizará cada segundo.

Cuando llegue a 0 el contador, se repetirá el proceso con un nuevo número aleatorio entre 2 y 6.

Cada vez que generes el número aleatorio saca por consola el valor de dicho número. No pongas ningún texto más, solo el valor.

Y así indefinidamente.