

Índice

Función <i>generalInputsOperacion()</i>	2
Función <i>iniciaListers()</i>	3
Función <i>muestraBotonGeneraPalabras()</i>	4
Función <i>comprobarResultado()</i>	4
Función <i>cargarPalabras(numeroLetras, numeroColumnas)</i>	5
Función <i>generarTabla(palabras, numeroColumnas)</i>	6
Función <i>generaCSV()</i>	7

Función *generalInputsOperacion()*

Esta función se usa para crear un elemento “div” con sus respectivos elementos “p”, “input” y “button” y añadirlo al body. Los input para los valores de la operación reciben dos *addEventListener()* cada uno; un *evento “keydown”* para asegurar que no se introduzca el carácter “-” y otro *evento “input”* que hace uso de la función *muestraBotonGeneraPalabras()* para comprobar que ambos input tienen algún valor y habilitar el botón para generar la tabla con las palabras. Además de eso se inhabilitan ambos botones, estos solo serán habilitados para su uso en una situación concreta.

Lo no menos importante de la función es que al principio se comprueba si existe ya este div que contiene todos los elementos y la tabla donde se muestra el resultado; además de limpiar la tabla, elimina ambos elementos antes de crearlos de nuevo. Esto permite realizar innumerables operaciones en la misma ejecución del script sin obtener errores ni sobrescribir elementos.

Función *iniciaListers()*

Esta función es la única que se usa en el script del documento HTML. Se encarga de asignarle un *addEventListener()* al input file que obtendrá el archivo con el que operar. Mediante el evento “*change*” se comprueba que se haya cargado un archivo en el input file, entonces se usa la función *generalInputsOperacion()* para crear y añadir los elementos y hace uso de la función *muestraBotonGeneraPalabras()*.

Función *muestraBotonGeneraPalabras()*

Esta función obtiene los elementos input donde se establecen ambos valores para la operación así como el botón para generar la tabla. Esta contiene otra función que se ejecutará mediante un *setInterval(comprobarResultado, 1000)* cada segundo:

Función *comprobarResultado()*

En esta función se comprueba que ambos input tengan un valor. Cuando los inputs reciben un valor se comprueba que el input file tenga un fichero cargado, entonces se detiene el intervalo, se habilita el botón de generar la tabla y se asigna un *addEventListener()* al botón con el evento "click" que hará uso de la función *cargarPalabras(numeroLetras, numeroColumnas)* pasando por parámetros los valores de los input. En caso de que ambos input tengan valor y no se haya cargado un archivo se muestra un mensaje de advertencia y no se habilita el botón. Al tratarse de una función ejecutada mediante un *setInterval()*, mientras los input no tengan valor se seguirá deshabilitando el botón para generar la tabla.

Función *cargarPalabras*(*numeroLetras*, *numeroColumnas*)

Esta función se ejecutará mediante el *evento* “click” del botón de generar la tabla. Obtiene el archivo cargado en el input file y lo lee mediante el objeto FileReader y sus métodos *readAsText*(*archivo*) y *onload* = (ev) =>{}. Con esta última se recoge el resultado del archivo en un string, a partir de él se realizan operaciones para dividir el texto en líneas; eliminar los espacios “ ”, saltos de línea “\n” y retornos de carro “\r”; obtener un nuevo array eliminando los elementos vacíos [“”] y por último recorrer el array en busca de las palabras cuya longitud sea igual a igual a la variable *numeroLetras* obtenida por parámetros y obtener un array de las palabras que cumplen la condición en esa ejecución del script.

Una vez obtenido el array con el resultado correspondiente comprobamos si al menos contiene un elemento, si es así se elimina el mensaje que haya podido mostrarse anteriormente por consultar palabras de las que no exista esa longitud; se vuelve a limpiar y eliminar la tabla por si se ejecuta más de una vez el script, y por último se hace uso de la función *generarTabla*(*palabrasFiltradas*, *numeroColumnas*).

En caso de que no haya elementos en el array se crea un elemento “h3” y se muestra un mensaje informando que no hay palabras con esa longitud y a los 5 segundos se elimina.

Función *generarTabla*(*palabras*, *numeroColumnas*)

En esta función se crea un elemento “table”, mediante dos bucles for se recorre el array de palabras y el número de columnas, por cada vuelta del primero se creará una fila y por cada vuelta del segundo tantas columnas como *numeroColumnas*. Se añaden las columnas a las filas y se añaden las filas a la tabla. Al terminar se añade la tabla al body y se habilita el botón generar CSV.

Función *generaCSV()*

Esta función se ejecuta con el *evento* “click” del botón generar CSV cuyo `addEventListener()` se encuentra en la función *generalInputsOperacion()*.

Primero de todo, por si esta función se usa más de una vez durante la ejecución del script, se comprueba si existe el elemento cuya función será descargar el fichero. Si es así se elimina el objeto *URL* generado con el *Blob* que contiene el elemento en la propiedad “href” y también elimina el elemento.

Su función es obtener el elemento “table” con los datos, recorrerlo y obtener el contenido de cada columna y añadir una “,” entre ellos si el elemento que obtiene no es el último de una fila. Por cada fila se añadirá un salto de línea “\n” al string que almacena el contenido para establecer una estructura de ficheros .csv. Una vez se han obtenido los datos se genera un objeto *Blob* con el string resultante y se le asigna el *tipo* “text/csv”; se crea un elemento “a” y se asigna su *atributo* “href” con el método *createObjectURL(blob)* de la clase *URL* que obtendrá el *Blob* creado con el resultado; se asigna su otro *atributo* “download” al que le indicaremos el nombre del archivo “palabras + ‘numeroLetras’ + .csv” donde *numeroLetras* será obtenido de la longitud de cualquier elemento de la tabla.

Por último se añade el elemento “a” al body y se establece un segundo de retraso para que de tiempo de realizar los cambios antes de producir el *evento* “click” al elemento que hemos creado para obtener la descarga.