Adrián Polo 16 de octubre de 2024

Adrián Polo 16 de octubre de 2024

Ejercicio 1

1. Visualiza por pantalla todo el contenido del array, separando cada dato en líneas distintas.
2. Añade al array un dato más. (mediante el uso [longitud])
3. Añade al array dos datos más mediante utilizando un solo método
4. Añade un dato más al principio del array.
5. Localiza un cierto dato dentro del array.
6. Elimina los últimos tres datos del array.
7. Crea un sub-array llamado array_recortado con los datos del array elementos, comprendidos entre la posición 4 y 8 (ambos inclusive).
8. Crea un nuevo array llamado elementos_MYCLS con los datos del array elementos en mayúsculas.

Ejercicio 2

1. Leer contenido del fichero
2. Crear array llamado datos, guardar el contenido del fichero en ese array mediante un split y mostrarlo en html.
    a. var datos= new Array();
    b. datos=contenidoFichero.split('/n');
    c. document.getElementById('contenidoArray').innerHTML=datos;

Adrián Polo 16 de octubre de 2024

# Ejercicio 1

Crea un array llamado elementos que tenga 10 datos de tipo string (tipo primitivo).

```
//Crea un array llamado elementos que tenga 10 datos de tipo string (tipo primitivo).
var elementos = ["manzana","banana","naranja","uva","fresa","kiwi","mango","piña","cereza","pera"];
```

Sobre el array creado realiza las siguientes acciones.

1. Visualiza por pantalla todo el contenido del array, separando cada dato en líneas distintas.

```
//Visualiza por pantalla todo el contenido del array, separando cada dato en líneas distintas.
function caso1(){
        var cadena="";
        for(i=0;i<elementos.length;i++){
                cadena+=elementos[i] + "<br>";
        }
        document.getElementById('texto').innerHTML=cadena;
}
```

Recorremos con un for de todos los elementos y lo asignamos a una cadena más un salto de línea para luego mostrarlo con un document.getElementById(‘id puesto en el HTML).

2. Añade al array un dato más. (mediante el uso [longitud])

```
//Añade al array un dato más. (mediante el uso [longitud])
function caso2(){
        elementos[elementos.length]=prompt("Introduzca el dato que quiere meter dentro del array");
}
```

Añadimos un dato al array elementos con el método, elementos.length, ya que el último índice del array es elementos.length-1, es decir, si el array tiene 7 elementos, el último índice es 6 (elementos.length-1), entonces, si queremos añadir un dato más al array seria elementos.length, para ponerlo en la siguiente posición a la última.

3. Añade al array dos datos más mediante utilizando un solo método

```
//Añade al array dos datos más mediante utilizando un solo método.
function caso3(){
	var dato1=prompt("Introduzca el dato que quiere meter dentro del array");
	var dato2=prompt("Introduzca el dato que quiere meter dentro del array");
	elementos.push(dato1, dato2);
}
```

Hacemos dos prompt para recoger los dos datos a introducir y los guardamos dentro de dos variables distintas, para añadirlos al array, se hace mediante el método push, es decir, elementos.push(los datos que quieras introducir separados por comas).

4. Añade un dato más al principio del array.

```
//Añade un dato más al principio del array.
function caso4(){
	elementos.unshift(prompt("Introduzca el dato que quiere meter dentro del array"));
}
```

Para añadir un dato dentro del array, pero al principio, es con el método unshift, es decir, elementos.unshift(el dato que quieras introducir), en este caso hago un prompt para que le salga una alerta al usuario para que introduzca el dato.

5. Localiza un cierto dato dentro del array.

```
//Localiza un cierto dato dentro del array.
function caso5(){
	var datoBuscado = prompt("Introduzca el dato que quiere buscar dentro del array");
	var indice = elementos.indexOf(datoBuscado);
	var cadena="";

	if (indice !== -1)
		cadena="El dato "+datoBuscado+" se encuentra en el índice "+indice+", es decir, el elemento "+(indice+1)+" del array.";
	else
		cadena="El dato "+datoBuscado+" no se encuentra en el array.";

	document.getElementById('texto').innerHTML=cadena;
}
```

Pedimos el dato a buscar dentro del array al usuario con un prompt, luego declaramos una variable con elementos.indexOf(datoABuscar), si encuentra el dato, en la variable se guarda el índice del elemento buscado, sino lo encuentra, en la variable se guarda -1, luego hacemos un if para ver si lo ha encontrado o no, y si lo ha encontrado en que posición lo encontró.

6. Elimina los últimos tres datos del array.

```
//Elimina los últimos tres datos del array.
function caso6(){
        elementos.splice(-3, 3);
}
```

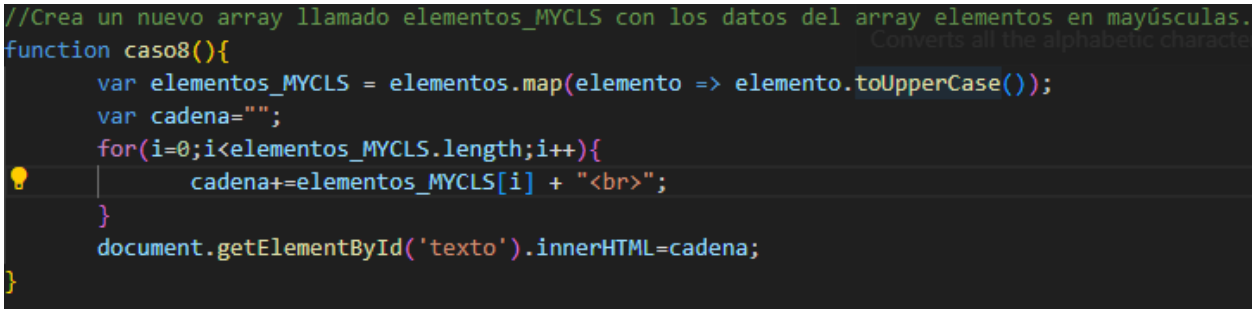
Utilizamos el método splice para eliminar tres datos del array, el primer parámetro del método -3, es decir, comenzará a eliminar desde el tercer elemento desde el final, el segundo parámetro del método es 3, es la cantidad de elementos a eliminar desde la posición dada anteriormente.

7. Crea un sub-array llamado array_recortado con los datos del array elementos, comprendidos entre la posición 4 y 8 (ambos inclusive).

```
//Crea un sub-array llamado array_recortado con los datos del array elementos, comprendidos entre la posición 4 y 8 (ambos inclusive).
function caso7(){
        var array_recortado = elementos.slice(4, 9);
        var cadena="";
        for(i=0;i<array_recortado.length;i++){
                cadena+=array_recortado[i] + "<br>";
        }
        document.getElementById('texto').innerHTML=cadena;
}
```

Creamos la variable array_recortado, que tendrá los datos del array elementos, desde la posición 4 a la 8, ambos incluidos, con el método slice(4,9), hacemos esa operación, desde la posición 4, ya que incluye la posición 4, a la 9, ya que la 9 no esta incluida, luego recorremos el array con un for y los mostramos con saltos en un documento.getElementById().

8. Crea un nuevo array llamado elementos_MYCLS con los datos del array elementos en mayúsculas.



```
//Crea un nuevo array llamado elementos_MYCLS con los datos del array elementos en mayúsculas.
function caso8(){
        var elementos_MYCLS = elementos.map(elemento => elemento.toUpperCase());
        var cadena="";
        for(i=0;i<elementos_MYCLS.length;i++){
                cadena+=elementos_MYCLS[i] + "<br>";
        }
        document.getElementById('texto').innerHTML=cadena;
}
```

Creamos la variable elementos_MYCLS, la que tendrá todo el array elementos pero en mayúsculas. Esa operación se hace con el método toUpperCase(), pero eso sirve para un string, es decir, tendríamos que recorrer el array y hacer ese toUpperCase() para cada elemento, entonces con el metodo elementos.map(), recorre todo el array sin tener que hacer un for. Una vez hecho todo esto, se recorre el array para mostrar ese array creado con saltos de línea en un document.getElementById().

# Ejercicio 2

1. Sigue los pasos del vídeo y genera el código necesario para cargar un fichero de texto.
2. A continuación se va a modificar el código para que en vez de aparecer en la página html el contenido del fichero, se cargue en un array llamado datos. Para ello, se debe añadir la declaración de la variable tipo array:
    - var datos = new Array();
3. Y a continuación, se asigna el contenido de cada línea del fichero al array, teniendo en cuenta hay que dividir en partes todo el string de contenido, como la separación es línea a línea, se utiliza como delimitador '\n'-
    - datos = contenido.split('\n');
4. Por último, visualizamos el array datos.

## 1. Leer contenido del fichero

```
var contenidoFichero="";
document.getElementById('fileInput').addEventListener('change', function(event) {
    const file = event.target.files[0];
    if (file) {
        const reader = new FileReader();

        reader.onload = function(e) {
            contenidoFichero = e.target.result;
        };

        reader.readAsText(file);


    } else {
        console.error('No se ha seleccionado un fichero');
    }
});
```

Creamos una variable contenidoFichero para guardar el contenido del fichero, como el propio nombre de la variable indica, después añadimos un evento para que se active cuando se le añade el fichero a ese input creado en el HTML con id 'fileInput'. La siguiente línea de código ***const file = event.target.files[0];*** accede al primer fichero seleccionado por el usuario (si hay alguno). Comprobamos que realmente ha seleccionado el fichero con el if.

```
reader.onload = function(e) {
        contenidoFichero = e.target.result;
};
```

***reader.onload = function(e) { ... }:*** Asigna una función que se ejecutará cuando el fichero haya sido leído completamente. ***onload*** es un evento que se dispara cuando la operación de lectura se completa.
***contenidoFichero = e.target.result;:*** Aquí, e.target.result contiene el contenido del fichero leído. Este contenido se guarda en la variable contenidoFichero.
***reader.readAsText(file);:*** Inicia la lectura del fichero como texto. Esta es una operación asíncrona; el código no se detiene aquí, y el programa continuará ejecutándose mientras se lee el fichero.

2. Crear array llamado datos, guardar el contenido del fichero en ese array mediante un split y mostrarlo en html.

```
function ejercicio2(){
    var datos = new Array();
    datos=contenidoFichero.split('/n');
    document.getElementById('contenidoArray').innerHTML=datos+'';
}
```

   a. var datos= new Array();

      Creamos una variable datos, que será un array, entonces lo definimos con un array con ***new Array();***

   b. datos=contenidoFichero.split(‘/n’);

      Hacemos que ese array tenga el contenido del fichero separado por los saltos de línea, es decir, por los ***/n***. Eso se hace mediante el método ***split(‘/n’)***, entonces en ese array, se guardan los elementos del fichero separados por delimitadores.

   c. document.getElementById(‘contenidoArray’).innerHTML=datos;

      Mostramos los elementos del array previamente construido con el ***split(‘\n’)***;