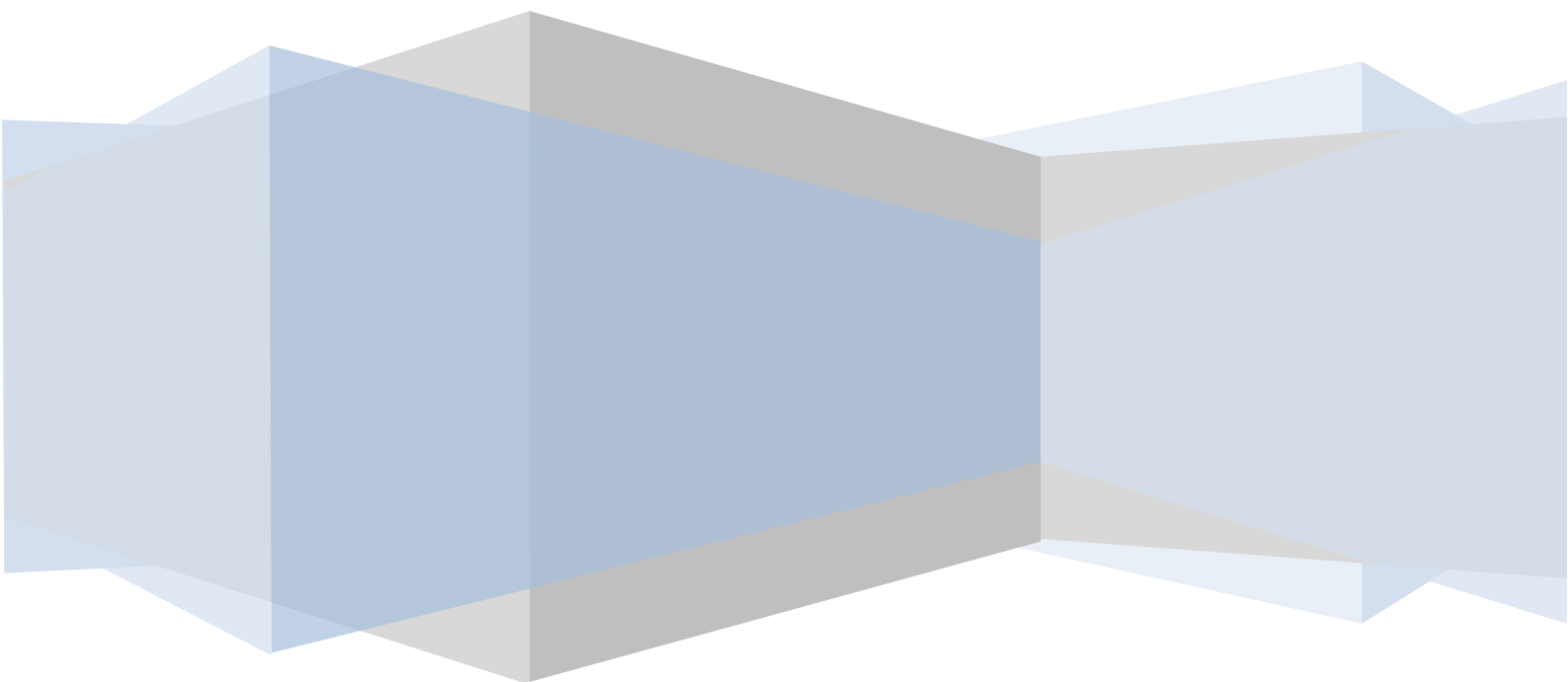


Tarea 1 UT4

Desarrollo web en entorno cliente

Ivan De Las Heras Martín



ÍNDICE

Contenido

INTRODUCCIÓN	2
EJERCICIO 1.....	3
EJERCICIO 2.....	6
BIBLIOGRAFÍA.....	9

INTRODUCCIÓN

En esta práctica realizaremos 2 ejercicios, poniendo en práctica los conocimientos adquiridos en JavaScript.

En este caso se explicarán debajo de forma más detallada las necesidades de los ejercicios:

EJERCICIO 1

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

Sobre el array creado realiza las siguientes acciones:

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.

```
<html>
  <head></head>
  <body>
    <script type="text/javascript">

      let elementos = ["posi-
cion1","posicion2","posicion3","posicion4"
,"posicion5","posicion6","posicion7","posicion8","posicion9","posicion10"
];

      document.write("<h2>EJERCICIO 1: Visualiza por pantalla todo
el contenido del array, separando cada dato en líneas distintas.</h2>");
      document.write(elementos);

      elementos[elementos.length] = "posicion11";
      document.write("<h2>EJERCICIO 2: Añade al array un dato más.
(mediante el uso [longitud]</h2>");
      document.write(elementos);

      elementos.push("posicion12","posicion13");
      document.write("<h2>EJERCICIO 3: Añade al array dos datos más
mediante utilizando un solo método.</h2>");
      document.write(elementos);

      let datoPrincipio = elementos.unshift("posicion0");
      document.write("<h2>EJERCICIO 4: Añade un dato más al princi-
pio del array.</h2>");
      document.write(elementos);
```

```
        document.write("<h2>EJERCICIO 5: Localiza un cierto dato dentro del array.</h2>");
        document.write(elementos.indexOf("posicion7"));

        elementos.length = elementos.length - 3;
        document.write("<h2>EJERCICIO 6: Elimina los últimos tres datos del array.</h2>");
        document.write(elementos);

        document.write("<h2>EJERCICIO 7: Crea un sub-array llamado array_recortado con los datos del array elementos, comprendidos entre la posición 4 y 8 (ambos inclusive).</h2>");
        let array_recortado = elementos;

        for(var i = 0; i < array_recortado.length; i++){
            if(i >= 4 && i <= 8){
                array_recortado.push(elementos[i]);
                document.write(array_recortado[i]);
            }
        }

        elementos = ["posicion1","posicion2","posicion3","posicion4",
,"posicion5","posicion6","posicion7","posicion8","posicion9","posicion10"
];

        document.write("<h2>EJERCICIO 8: Crea un nuevo array llamado elementos_MYCLS con los datos del array elementos en mayúsculas.</h2>");
        let elementos_MYCLS = elementos;
        for(var i = 0;i < elementos.length; i++){
            elementos_MYCLS[i] = elementos_MYCLS[i].toUpperCase();
        }
        document.write(elementos_MYCLS);
</script>

</body>

</html>
```

En este ejercicio prácticamente se utiliza un mismo array para realizar todas las partes de dicho ejercicio, para el último punto es necesario volver a reiniciar los valores del array a sus valores originales debido a que entran en conflicto la parte siete y ocho.

En vez de utilizar alert() para mostrar el resultado en pantalla se utiliza document.write() para una mejor visualización de los resultados.

Imagen final del resultado obtenido:

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

posicion1,posicion2,posicion3,posicion4,posicion5,posicion6,posicion7,posicion8,posicion9,posicion10

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

posicion1,posicion2,posicion3,posicion4,posicion5,posicion6,posicion7,posicion8,posicion9,posicion10,posicion11

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

posicion1,posicion2,posicion3,posicion4,posicion5,posicion6,posicion7,posicion8,posicion9,posicion10,posicion11,posicion12,posicion13

EJERCICIO 4: Añade un dato más al principio del array.

posicion0,posicion1,posicion2,posicion3,posicion4,posicion5,posicion6,posicion7,posicion8,posicion9,posicion10,posicion11,posicion12,posicion13

EJERCICIO 5: Localiza un cierto dato dentro del array.

7

EJERCICIO 6: Elimina los últimos tres datos del array.

posicion0,posicion1,posicion2,posicion3,posicion4,posicion5,posicion6,posicion7,posicion8,posicion9,posicion10

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

posicion4,posicion5,posicion6,posicion7,posicion8

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

POSICION1,POSICION2,POSICION3,POSICION4,POSICION5,POSICION6,POSICION7,POSICION8,POSICION9,POSICION10

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.

Ej4_2.html

```
<html>
  <head>
    <meta charset="utf-8">
    <title>Ejercicio 4_2</title>
    <link rel="stylesheet" href="estilos.css">
    <script src="archivos.js">

    </script>
  </head>
  <body>
    <div id="contenedor">
      <section id="zonaFormulario">
        <form name="formulario">

          <p>Archivos: <br> <input type="file" name="archivos"
id="archivos"></p>

        </form>
      </section>

      <section id="zonaDatos">

        No hay información

      </section>
    </div>

  </body>
</html>
```

El archivo html contiene dos secciones, una con un botón de tipo input file el cual permite agregar cualquier archivo a un formulario.

La segunda sección es donde se volcará la información de dicho archivo.

archivos.js

```
function comenzar(){
    //identificar la sección de datos
    zonaDatos = document.getElementById("zonaDatos");
    //identificar el botón input file
    var archivos = document.getElementById("archivos");
    //al pulsar o cambiar de estado se llamará a la función procesar()
    archivos.addEventListener("change",procesar,false);
}

function procesar(e){
    //e es el botón y tiene la propiedad files que es un array de los
    //archivos seleccionados
    var archivos = e.target.files;
    //cogemos el primer archivo del array
    var miArchivo = archivos[0];
    //lector permite leer la información del objeto devuelto
    var lector = new FileReader();

    //leemos el fichero como texto con codificación utf-8
    lector.readAsText(miArchivo, "utf-8");
    //al cargar el lector se llama a la función mostrarWeb()
    lector.addEventListener("load",mostrarEnWeb, false);
}

function mostrarEnWeb(e){
    //e es el lector y se le da lo que haya leído del fichero
    var resultado = e.target.result;

    var datos = new Array();
    datos = resultado.split('\n');
    //visualizar array completo en la página
    //en la sección datos incluya todo en código HTML
    zonaDatos.innerHTML = datos;
    //visualizar línea por línea en consola
    for(var linea of datos) {
        console.log('[linea]', linea)
    }
}

//al cargar la ventana se llamará a la función comenzar()
window.addEventListener("load",comenzar,false);
```

En el archivo .js realizaremos las operaciones necesarias para recoger los datos del fichero y mostrarlos en un array en la página web.

estilos.css

```
@charset "utf-8";

#contenedor{
    width: 100px;
    position: relative;
    margin: auto;
}

#zonaFormulario{
    float: left;
    padding: 30px;
    border: 2px solid #F00;
}

#zonaDatos{
    float: left;
    width: 500px;
    margin-left: 20px;
    padding: 20px;
    border: 2px dashed #0000FF;
}

.directorio{
    color: #0000FF;
    font-weight: bold;
    cursor: pointer;
}
```

Estilos.css contiene toda la parte visual del archivo html

BIBLIOGRAFÍA

[W3Schools](#)

[DelfStack](#)

Apuntes de clase.