

25-10-2023

Estructura del lenguaje JavaScript

Tarea 02

ROBERTO RODRÍGUEZ JIMÉNEZ
roberto.rodjim.1@educa.jcyl.es

Contenido

Tarea online DWEC02.....	2
¿Qué contenidos o resultados de aprendizaje trabajaremos?.....	2
Resultados de Aprendizaje.....	2
Contenidos.....	2
1.- Descripción de la tarea	3
Caso práctico	3
¿Qué te pedimos que hagas?	3
Apartado 1	3
Apartado 2	3
2.- Información de interés.....	4
Recursos necesarios	4
Consejos y recomendaciones	4
3.- Evaluación de la tarea	4
Criterios de evaluación implicados.....	4
¿Cómo valoramos y puntuamos tu tarea?	4
RESOLUCIÓN.....	5
Apartado 1.....	5
Apartado 2.....	7
Código.....	9
index.html	9
tablaMultiplicar.js.....	11
estilos.css.....	13

Tarea online DWEC02

Título de la tarea: Estructura del lenguaje JavaScript

Unidad: DWEC02

Ciclo formativo y módulo: Desarrollo de Aplicaciones Web - Desarrollo web en entorno cliente

Curso académico: 2020/2021

¿Qué contenidos o resultados de aprendizaje trabajaremos?

Resultados de Aprendizaje

- RA2: Escribe sentencias simples, aplicando la sintaxis del lenguaje y verificando su ejecución sobre navegadores Web.

Contenidos

1. Fundamentos de JavaScript.
 - 1.1. Comentarios en el código.
 - 1.2. Constantes y variables.
 - 1.3. Tipos de datos.
 - 1.3.1. Conversiones de tipos de datos.
 - 1.4. Operadores.
 - 1.4.1. Algunos operadores de interés.
 - 1.5. Condiciones y bucles.
 - 1.5.1. Estructuras de control.
 - 1.5.2. Bucles.
 - 1.6. Ejemplo sencillo con JavaScript.

1.- Descripción de la tarea

Caso

práctico

En BK Programación han decidido ponerse manos a la obra y Antonio comienza a estudiar los fundamentos del lenguaje JavaScript, (siempre bajo la tutoría de Juan, que estará ahí para ayudarle en todo momento).

En la unidad anterior Antonio analizó las posibilidades de los lenguajes de script, decidió qué lenguaje emplearía para la programación en el entorno cliente y vio cómo insertar en la página HTML dicho lenguaje de script.

Cómo han decidido emplear el lenguaje Javascript, lo primero que hará Antonio es ver los fundamentos de dicho lenguaje, y cuál es la estructura básica para comenzar a programar lo más pronto posible.

Ada está muy entusiasmada con este proyecto y está convencida de que Antonio será capaz de hacer ese trabajo sin ningún problema.

¿Qué te pedimos que hagas?

Apartado 1

Realizar una pequeña aplicación en JavaScript que muestre lo siguiente:

- ✓ Tabla de multiplicar del 7.
- ✓ Tabla de sumar del 8.
- ✓ Tabla de dividir del 9.

Utilizar los tres tipos de bucles que hay en JavaScript (para cada número un tipo de bucle diferente).

Apartado 2

Sabiendo que cuando desplazamos 1 bit a la derecha dividimos un entero por 2 y cuando lo desplazamos a la izquierda estamos multiplicando por 2. Tu aplicación también debe imprimir el resultado de las siguientes operaciones empleando desplazamiento de bits:

- ✓ $125 / 8 =$
- ✓ $40 \times 4 =$
- ✓ $25 / 2 =$
- ✓ $10 \times 16 =$

2.- Información de interés

Recursos necesarios

Editor web para teclear el código de la aplicación y un navegador web para probar su funcionamiento.

Consejos y recomendaciones

Al final de la unidad 2 dispones de un ejemplo básico que te puede servir de referencia para comenzar con tu aplicación.

3.- Evaluación de la tarea

Criterios de evaluación implicados

- a) Se ha seleccionado un lenguaje de programación de clientes Web en función de sus posibilidades.
- b) Se han utilizado los distintos tipos de variables y operadores disponibles en el lenguaje.
- c) Se han identificado los ámbitos de utilización de las variables.
- d) Se han reconocido y comprobado las peculiaridades del lenguaje respecto a las conversiones entre distintos tipos de datos.
- e) Se han utilizado mecanismos de decisión en la creación de bloques de sentencias.
- f) Se han utilizado bucles y se ha verificado su funcionamiento.
- g) Se han añadido comentarios al código.
- h) Se han utilizado herramientas y entornos para facilitar la programación, prueba y depuración del código.
- i)

¿Cómo valoramos y puntuamos tu tarea?

Rúbrica de la tarea

Tabla de multiplicar del número 7 utilizando los tres tipos de bucles.	2 puntos.
Tabla de multiplicar del número 8 utilizando los tres tipos de bucles.	2 puntos.
Tabla de multiplicar del número 9 utilizando los tres tipos de bucles.	2 puntos.
Resultado de las operaciones empleando desplazamiento de bits.	2,5 puntos.
Claridad y presentación de los resultados, los comentarios utilizados en el código y su indentación.	1,5 puntos.

RESOLUCIÓN

Apartado 1

Realizar una pequeña aplicación en JavaScript que muestre lo siguiente:

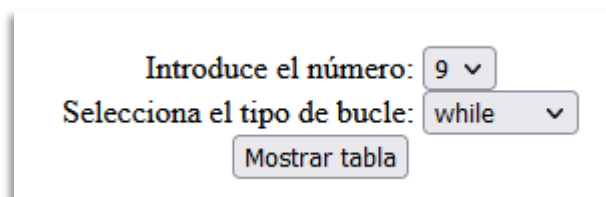
- ✓ Tabla de multiplicar del 7.
- ✓ Tabla de sumar del 8.
- ✓ Tabla de dividir del 9.

Utilizar los tres tipos de bucles que hay en JavaScript (para cada número un tipo de bucle diferente).

Para la realización del punto 1 se ha conservado la estructura del ejercicio del temario.

En *index.html* se ha cambiado el *input* por un *select* para limitar los números a los pedidos (7, 8 y 9).

Para seleccionar el tipo de bucle se ha añadido un nuevo *select*.

Un formulario web con un título "Introduce el número:" seguido de un select con el valor "9". Debajo, "Selecciona el tipo de bucle:" seguido de un select con el valor "while". En la parte inferior hay un botón que dice "Mostrar tabla".

1 Formulario para las tablas

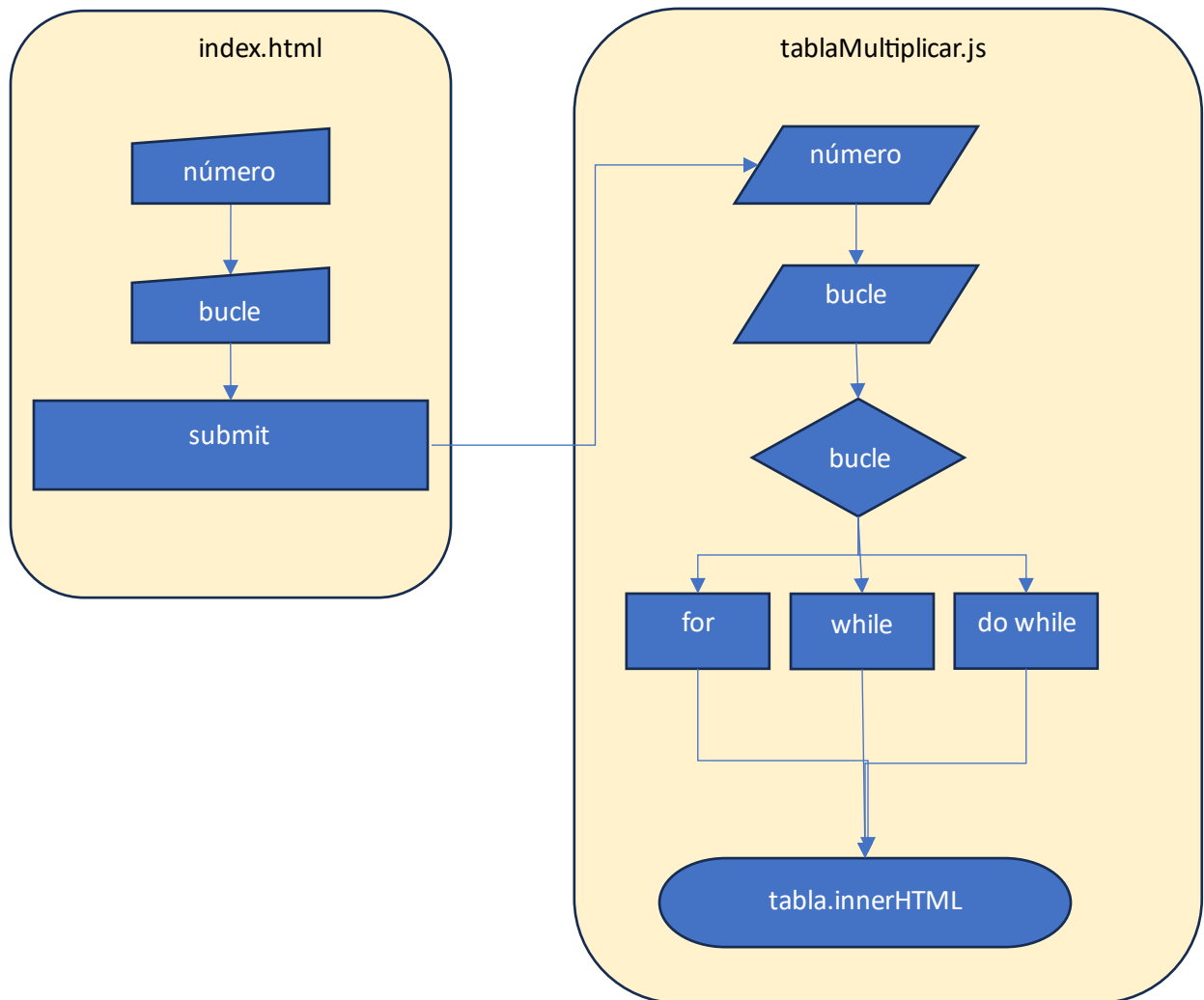
Se ha añadido una hoja de estilo mínima para organizar un poco los componentes.

El código JavaScript se ha modificado para añadir una función dependiendo del tipo de bucle buscado.

```
30 > /**...
36 > function bucleFor(multiplicando) {...
45 > }
46
47 > /**...
53 > function bucleWhile(multiplicando) {...
64 > }
65
66 > /**...
72 > function bucleDoWhile(multiplicando) {...
83 > }
```

2 Declaración de las funciones de bucle

Funcionamiento:



Después de seleccionar las opciones de número y bucle y enviar el formulario, la constante *mostrarTabla* ejecuta una función anónima en la que se obtienen los valores del formulario.

Dependiendo el valor que tenga *bucle*, mediante un *switch*, se llama a una función u otra que devolverá el código HTML que deberá insertarse en la tabla.

Apartado 2

Sabiendo que cuando desplazamos 1 bit a la derecha dividimos un entero por 2 y cuando lo desplazamos a la izquierda estamos multiplicando por 2. Tu aplicación también debe imprimir el resultado de las siguientes operaciones empleando desplazamiento de bits:

- ✓ 125 / 8 =
- ✓ 40 x 4 =
- ✓ 25 / 2 =
- ✓ 10 x 16 =

En *index.html* se muestra un listado con las operaciones y un botón para ejecutarlas.

Debajo se muestra el proceso de desplazamiento de los bits en cada una de las operaciones con el resultado que deben arrojar.

Operaciones con operadores binarios

125 / 8 =
0111 1101 (125) >> 0011 1110 (62) >> 0001 1111 (31) >> 0000 1111 (15) >> 0000 0111 (7)

40 * 4 =
0010 1000 (40) << 0101 0000 (80) << 1010 0000 (160)

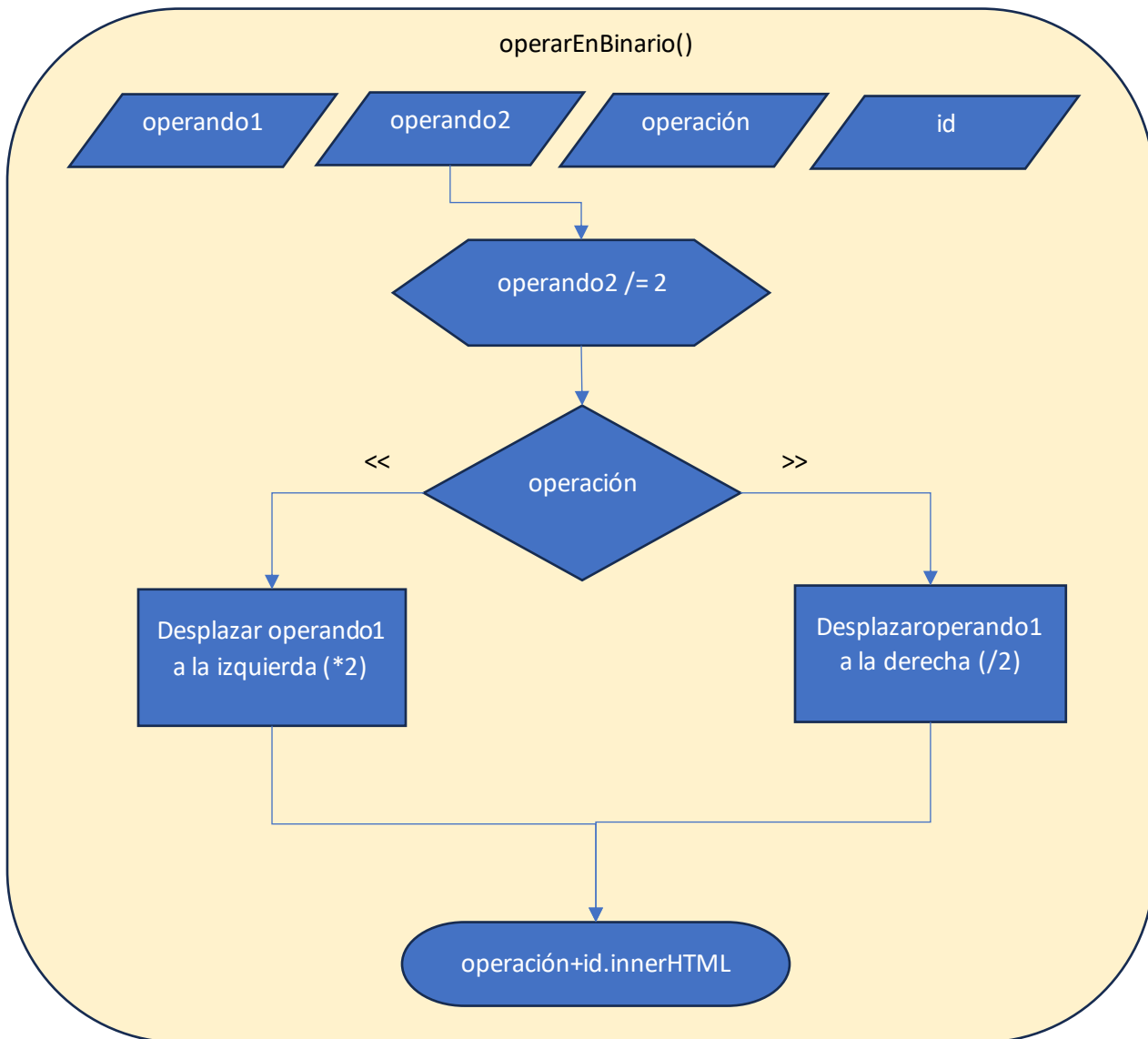
25 / 2 =
0001 1001 (25) >> 0000 1100 (12)

10 * 16 =
1010 (10) << 0001 0100 (20) << 0010 1000 (40) << 0101 0000 (80) << 1010 0000 (160) << 0001 0100 0000 (320) << 0010 1000 0000 (640) << 0101 0000 0000 (1280) << 1010 0000 0000 (2560)

3Operaciones con desplazamiento de bits

Funcionamiento:

El botón "Calcular" llama a la función *operarEnBinario()*, pasándole los operandos, el tipo de operación y el id del lugar en el que mostrar el resultado.



Código

index.html

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>DAW02 - Tarea</title>
    <link rel="stylesheet" href="css/estilos.css" />
  </head>
  <body>
    <h2>DAW02 - Tarea</h2>
    <h1>Tabla de multiplicar usando diferentes bucles</h1>

    <form onsubmit="mostrarTabla()">
      <label for="numero">Introduce el número: </label>
      <!-- <input type="number" name="numero" id="numero" required /> -->
      <select name="numero" id="numero">
        <option value="7" selected>7</option>
        <option value="8">8</option>
        <option value="9">9</option>
      </select>
      <br>
      <label for="bucle">Selecciona el tipo de bucle: </label>
      <select name="bucle" id="bucle">
        <option value="for" selected>for</option>
        <option value="while">while</option>
        <option value="doWhile">do while</option>
      </select>
      <br>
      <input type="submit" name="submit" id="submit" value="Mostrar tabla" />
    </form>
    <div id="tabla"></div>

    <hr />

    <h2>Operaciones con operadores binarios</h2>

    <dl>
      <dt>
        125 / 8 =
        <span id="operacion1">
          <button onclick="oprearEnBinario(125,8,'/',1)">calcular</button>
        </span>
      </dt>
      <dd>
```

```

0111 1101 (125) >> 0011 1110 (62) >> 0001 1111 (31) >> 0000 1111 (15) >> 0000 0111
(7)
</dd>
<dt>
  40 * 4 =
  <span id="operacion2">
    <button onclick="oprearEnBinario(40,4,'*',2)">calcular</button>
  </span>
</dt>
<dd>
  0010 1000 (40) << 0101 0000 (80) << 1010 0000 (160)
</dd>
<dt>
  25 / 2 =
  <span id="operacion3">
    <button onclick="oprearEnBinario(25,2,'/',3)">calcular</button>
  </span>
</dt>
<dd>
  0001 1001 (25) >> 0000 1100 (12)
</dd>
<dt>
  10 * 16 =
  <span id="operacion4">
    <button onclick="oprearEnBinario(10,16,'*',4)">calcular</button>
  </span>
</dt>
<dd>
  1010 (10) << 0001 0100 (20) << 0010 1000 (40) << 0101 0000 (80) << 1010 0000 (160)
<< 0001 0100 0000 (320) << 0010 1000 0000 (640) << 0101 0000 0000 (1280) << 1010 0000 0000
(2560)
</dd>
</dl>

<script type="text/javascript" src="js/tablaMultiplicar.js"></script>
</body>
</html>

```

tablaMultiplicar.js

```
const mostrarTabla = () => {
  this.event.preventDefault();
  const numero = Number(document.getElementById("numero").value);
  const bucle = document.getElementById("bucle").value;

  let tabla = document.getElementById("tabla");
  let tablaMultiplicar = `

## Tabla de multiplicar del numero ${numero} y bucle ${bucle}</h2>`; tablaMultiplicar += "<ul>"; // Seleccionamos una función para cada uno de los // bucles que se quiera ejecutar. switch (bucle) { case "for": tablaMultiplicar += bucleFor(numero); break; case "while": tablaMultiplicar += bucleWhile(numero); break; case "doWhile": tablaMultiplicar += bucleDoWhile(numero); break; default: } tablaMultiplicar += "</ul>"; tabla.innerHTML = tablaMultiplicar; }; /** * Devuelve la table del número pasado como parámetro usando * el bucle for. * @param {} numero el número sobre el que recorrer la tabla * @returns el resultado de la tabla */ function bucleFor(multiplicando) { let resultado = ""; let max = 10; for (let multiplicador = 0; multiplicador <= max; multiplicador++) { resultado += `<li>${multiplicando}*${multiplicador} = ${ multiplicando * multiplicador }</li>`; } return resultado; } /** * Devuelve la table del número pasado como parámetro usando


```

```

* el bucle while.
* @param {} numero el número sobre el que recorrer la tabla
* @returns el resultado de la tabla
*/
function bucleWhile(multiplicando) {
  let resultado = "";
  let multiplicador = 0;
  let max = 10;
  while (multiplicador <= max) {
    resultado += `<li>${multiplicando}*${multiplicador} = ${
      multiplicando * multiplicador
    }</li>`;
    multiplicador++;
  }
  return resultado;
}

/**
* Devuelve la table del número pasado como parámetro usando
* el bucle do while.
* @param {} numero el número sobre el que recorrer la tabla
* @returns el resultado de la tabla
*/
function bucleDoWhile(multiplicando) {
  let resultado = "";
  let multiplicador = 0;
  let max = 10;
  do {
    resultado += `<li>${multiplicando}*${multiplicador} = ${
      multiplicando * multiplicador
    }</li>`;
    multiplicador++;
  } while (multiplicador <= max);
  return resultado;
}

/**
* Función que devuelve el resultado de la operación mediante el
* desplazamiento de bits.
*
* @param {*} operando1
* @param {*} operando2
* @param {*} operacion si es multiplicación o división
* @param {*} id identificación de la operación
*/
function opearEnBinario(operando1, operando2, operacion, id) {
  // Dividimos el operando2 para obtener el número de desplazamientos.
  operando2 /= 2;

  // Obtener el resultado dependiendo de la operación
  let resultado =
    operacion === "*" ? operando1 << operando2 : operando1 >> operando2;

```

```
// Re-escribir el contenido del span
document.getElementById("operacion" + id).innerHTML = resultado;
}
```

estilos.css

```
body{
    text-align: center;
}

form,
#tabla{
    width: 300px;
    margin: 0 auto;
}

ul{
    list-style: none;
}

dd{
    margin-bottom: 20px;
    font-size: 0.8rem;
}
```