1 Implementando JavaScript

1.1 Sumario

1.2 Incorporación de JavaScript en un documento HTML

El código JavaScript se puede incorporar al documento mediante tres técnicas diferentes, que podemos ver a continuación.

1.2.1 El código se puede insertar en un elemento por medio de atributos (en línea)

Podemos ver un ejemplo donde se crea una web interactiva con JavaScript. Esta página tiene dos párrafos, si hacemos *click* en el primero sale una ventana de alerta, si hacemos *click* en el segundo no pasa nada:
Solución:

```
<!DOCTYPE html>
<html>
<head>
<title>JavaScript</title>
</head>
<body>
<section>
<p onclick="alert('Hiciste click!')">Haz click aquí</p>
<p>No puedes hacer click aquí</p>
</section>
</body>
</html>
```

El atributo **onclick** es parte de una serie de atributos provistos por HTML para responder a eventos. La lista de atributos disponibles es extensa, pero se pueden organizar en grupos dependiendo de sus propósitos.

Los atributos más usados asociados con el ratón son:

- **onclick** ? Click con el botón izquierdo.
- **ondblclick** ? Doble click con el botón izquierdo del ratón.
- **oncontextmenu** ? Click con el botón derecho del ratón.
- **onmousedown** ? Pulsar el botón izquierdo o derecho del ratón.
- **onmouseup** ? Liberar el botón izquierdo del ratón.
- **onmouseenter** ? Cuando el ratón se introduce en el área ocupada por el elemento.
- **onmouseleave** ? Cuando el ratón abandona el área ocupada por el elemento.
- **onmouseover** ? Cuando el ratón se mueve sobre el elemento o cualquiera de sus elementos hijos.
- **onmouseout** ? Cuando el ratón abandona el área ocupada por el elemento o cualquiera de sus elementos hijos.
- **onmousemove** ? Cuando el ratón se encuentra sobre el elemento y se mueve.
- **onwheel** ? Cada vez que se hace girar la rueda del ratón.

Mostramos ahora los atributos disponibles para responder a eventos generados por el teclado. Estos tipos de atributos se aplican a elementos que aceptan una entrada del usuario, como los elementos **<input>** y **<textarea>**.

- **onkeypress** ? Responde al evento keypress.
- **onkeydown** ? Responde al evento keydown.
- **onkeyup** ? Responde al evento keyup.

Tenemos otros dos atributos asociados al documento:

- **onload** ? Responde al evento *load* --- al cargar la página.
- **onunload** ? Responde al evento *unload* --- al abandonar la página.

1.2.2 Incorporar al documento como contenido del elemento *<script>*

Para trabajar con códigos extensos y personalizar las funciones, tenemos que agrupar el código con el elemento **<script>**. El elemento **<script>** actúa igual que el elemento **<style>** para CSS, organizando el código en un solo lugar y afectando al resto de los elementos en el documento usando referencias.

Este elemento **<script>** y su contenido se pueden ubicar en cualquier parte del documento, pero, normalmente, se introducen justo antes del cierre del elemento **<body>**, para que ya se encuentren cargados en el navegador todos los elementos de la página antes de ejecutarlo.

Podemos ver un ejemplo sencillo que, al abrir el documento en un navegador, debería verse una ventana emergente con el mensaje ?Bienvenido!?.

Solución:

```
<!DOCTYPE html>
<html>
<head>
<title>JavaScript</title>
</head>
<body>
<section>
<p>Hola</p>
</section>

<script type="text/javascript">
 alert('¡Bienvenido!')
</script>

</body>
</html>
```

1.2.3 Cargar desde un archivo externo

Si utilizamos el mismo código en más de un documento, introducir JavaScript con el elemento **<script>** no es práctico. En este caso, lo mejor, es introducir el código JavaScript en un archivo externo y luego cargarlo desde los documentos que lo requieren. Para este propósito, el elemento **<script>** incluye el atributo **src**.

El elemento **<script>** del siguiente documento carga el código JavaScript desde un archivo llamado **micodigo.js**.

```
...
    <script src="micodigo.js"></script>
</body>
...
```

El contenido de *micodigo.js* será simplemente:

```
alert('¡Bienvenido!');
```

1.2.4 Ejemplos

Nota: Para realizar estos primeros ejercicios sobre el núcleo de JavaScript, las salidas, además de utilizar el objeto **alert()** para verlas, también podemos utilizar la "consola" del navegador (por ejemplo, la de **Chrome** la podemos ver pulsando **F12**).

Podemos ver un ejemplo sencillo que, al abrir el documento en un navegador, debería verse en la consola un mensaje ?¡Bienvenido!?.

Solución:

```
<!DOCTYPE html>
<html><head>
    <title>JavaScript</title>
</head>
<body>
    <section><p>Hola</p></section>
    <script type="text/javascript">
        console.log('¡Bienvenido!');
    </script>
</body>
</html>
```

Nota: Aunque veremos más adelante el DOM (*Document Object Model*) de JavaScript y cómo crear funciones, podemos adelantar un pequeño código que nos permitirá interactuar con el navegador para comprobar, de un modo cómodo, las salidas de las funciones a programar y, además, poder dar valor a variables de un forma dinámica.

Podemos ver un ejemplo sencillo donde aparecen dos cajas de texto, un botón para ejecutar una función de JavaScript y una etiqueta donde se muestra la salida al pulsar dicho botón. En este caso, la operación que realiza la función es una simple suma de números.

Utiliza el siguiente ejemplo como base para realizar las siguientes funciones que se piden, en ellas nos centraremos en aprender el núcleo de JavaScript: tipos de variables, condiciones, bucles, etc. Más adelante nos pararemos a ver en profundidad las funciones, el DOM, etc., en estos momentos ese no es el objetivo.

Solución:

```
<html lang="es">
<head>
  <meta charset="UTF-8">
  <title>JavaScript</title>
</head>
<body>
  <fieldset>
    <legend>Escribe dos números para sumarlos</legend>
    <label for="txtN1">Número 1:</label>
    <input type="number" id="txtN1" name="txtN1">
    <br><br>
    <label for="txtN2">Número 2:</label>
    <input type="number" id="txtN2" name="txtN2">
    <br><br>

    <input type="button" name="Calcular" onclick="calcular()" value="Calcular">
    <br><br>
    <div id="txtSol"></label>
  </fieldset>

  <script type="text/javascript">
    function sumar(n1, n2) {  //Función suma de dos números
      return parseInt(n1) + parseInt(n2); }

    function calcular() { //Función calcular() que llama sumar() y muestra resultado
      //Seleccionamos la caja de texto por su "id" y, de ella, el valor escrito en su interior
      var n1 = document.getElementById('txtN1').value;
      var n2 = document.getElementById('txtN2').value;
      //Seleccionamos el elemento "label" por su "id"
      //e insertamos la solución como código html en su interior
      document.getElementById('txtSol').innerHTML = sumar(n1, n2); }
  </script>

</body>
</html>
```

1.3 Comentarios en el código

A la hora de programar en cualquier lenguaje, es muy importante que comentes tu código. Los comentarios son sentencias que el intérprete de JavaScript ignora. Sin embargo estas sentencias permiten a los desarrolladores dejar notas sobre cómo funcionan las cosas en sus *scripts*.

Los comentarios ocupan espacio dentro de tu código de JavaScript, por lo que cuando alguien se descargue vuestro código necesitará más o menos tiempo, dependiendo del tamaño de vuestro fichero. Aunque esto pueda ser un problema, es muy recomendable el que documentes tu código lo máximo posible, ya que esto te proporcionará muchas más ventajas que inconvenientes.

JavaScript permite dos estilos de comentarios:

- Un estilo consiste en dos barras inclinadas hacia la derecha (sin espacios entre ellas), y es muy útil para comentar una línea sencilla. JavaScript ignorará cualquier carácter a la derecha de esas barras inclinadas en la misma línea, incluso si aparecen en el medio de una línea.

```
// Este es un comentario de una línea
var nombre="Marta" // Otro comentario sobre esta línea
```

```
// Podemos dejar por ejemplo
//
// una línea en medio en blanco
```

- Para comentarios más largos, por ejemplo de una sección del documento, podemos emplear en lugar de las dos barras inclinadas el /* para comenzar la sección de comentarios, y */ para cerrar la sección de comentarios.

  Por ejemplo:

```
/* Ésta es una sección
de comentarios
en el código de JavaScript */
```

  O también:

```
/* ----------------------------------
función imprimir()
Imprime el listado de alumnos en orden alfabético
------------------------------------*/
function imprimir()
{
// Líneas de código JavaScript
}
```

Volver