Tema 3

El lenguaje de programación de clientes

MEDAC
Instituto Oficial de Formación Profesional

El lenguaje de programación de clientes

Aprendiendo Javascript

- Tipos de datos
- Variables
- Comentarios y etiquetas
- Conversión de tipos de datos

MEDAC

Javascript

- Javascript es un lenguaje de programación utilizado para crear pequeños programas encargados de realizar acciones dentro del ámbito de una página web
- Con Javascript dotaremos de interactividad a nuestras páginas. Además, podemos crear efectos especiales, controlar las diferentes partes del navegador, validar entradas de datos..., en resumen, aumentar la funcionalidad de nuestras páginas web.
- Es el navegador del cliente (el que visualiza la página) el encargado de interpretar las instrucciones Javascript y ejecutarlas para realizar todo lo que le indiquemos, de modo que el mayor recurso, y tal vez el único, con que cuenta este lenguaje es el propio navegador.

Javascript

- En la actualidad JavaScript se considera un lenguaje de programación seguro:
  - no tiene acceso a la memoria ni a la CPU.
  - no tiene acceso directo al sistema operativo.
  - Es posible que JavaScript interactúe con la cámara o con el micrófono de nuestro sistema. Sin embargo, para poder realizar esta tarea, primero debe solicitar permiso de forma explícita al usuario.
  - no puede acceder al contenido de una ventana o pestaña diferente a la que está ejecutando si no provienen del mismo dominio. (Same Origin Policy)
- A la hora de comunicar con el servidor, JavaScript nos proporciona mecanismos de comunicación asíncronos o síncronos.
- Como el protocolo HTTP no puede guardar el estado de una comunicación, JavaScript ofrece mecanismos para guardar datos como cookies y almacenamiento local.

Formas de incluir JavaScript

La forma correcta de incluir Javascript en una página HTML es utilizando la etiqueta <script> en el documento. Se pueden incluir tantas etiquetas <script> como se quiera en un documento.

```
<script type="text/javascript">
    código JavaScript
</script>
```

HTML 5

```
<script>
    código JavaScript
</script>
```

MEDAC

Formas de incluir JavaScript

¿En qué parte del documento HTML colocamos esa etiqueta? En el <head>. Es posible hacerlo en el body pero a la larga el código será más difícil de mantener y cambiar

```
<!DOCTYPE html>
<html>
    <head>
        <title>Titulo de la página</title>
        <!-- etiquetas META -->
        <!-- enlaces a archivos CSS -->
        <!-- enlaces a recursos (pej: favicon) -->
        <script type="text/javascript">
            código JavaScript
        </script>
    </head>
    <body>
         contenido de body (código HTML)
    </body>
</html>
```

MEDAC

Formas de incluir JavaScript

Existe otra forma aún más adecuada de utilizar JavaScript en nuestra página: Haciendo uso del atributo src de la etiqueta <script> .Este atributo permite incluir un archivo externo que sólo contiene código JavaScript y que será incluido en la página.

```
<!DOCTYPE html>
<html>
    <head>
        <title>Titulo de la página</title>
        <!-- etiquetas META -->
        <!-- enlaces a archivos CSS -->
        <!-- enlaces a recursos (pej: favicon) -->
        <script type="text/javascript" src="ruta relativa
                        archivo.js></script>
    </head>
    <body>
         contenido de body (código HTML)
    </body>
</html>
```

MEDAC

Javascript en un fichero independiente

- Aunque en esta etiqueta no tiene nada en su interior, la etiqueta <script> debe cerrarse.

- Se enlaza un archivo externo, por tanto, aparte de la página HTML debemos crear el archivo .js con el código que necesitemos.

- El archivo externo que se enlaza es un archivo de texto que contiene SOLO código JavaScript (nada de HTML o CSS) , y cuyo nombre acaba con la extensión js.

Formas de incluir JavaScript

También podemos incluir código JavaScript como respuesta a algún evento

```
<button onclick="alert('Has pulsado el botón')">Pulsa</button>
```

MEDAC

Sintaxis básica

- Comentarios: parte de código que no es interpretada por el navegador y cuya utilidad radica en facilitar la lectura al programador

Existen dos tipos de comentarios en el lenguaje:

- La doble barra (//): sirve para comentar una línea de código.
- Los signos /* para empezar el comentario y */ para terminarlo: Este tipo de comentario es capaz de generar bloques

- Sensible a mayúsculas y minúsculas
- La separación de instrucciones se puede hacer con ; o con salto de línea. Se recomienda usar ; por semejanza a JAVA, PHP, etc..

Ejercicio (*)

Añadir a una página HTML un script que muestre un alert de la siguiente forma:

- Añadimos el alert de forma interna sin función
- Añadimos el alert en un evento en un botón
- Igual que el punto anterior pero usamos una función
- Igual pero la función en un archivo externo

Ejercicio (*)

Crear una página con una imagen y un botón. Añadir el código necesario para que cuando pulse el botón salga un alert (“hola”) y cuando pase el ratón encima de la foto (onmouseover) salga un alert(“adios”).

Variables

- Usada para almacenar valores que pueden alterarse durante su tiempo de vida
- Identificador
  - Puede contener letras, dígitos y guiones bajos
  - Debe comentar por letra ó por los caracteres _ ó $
  - No puede contener espacios
  - No se puede usar palabras reservadas del lenguaje
  - Distingue entre mayúsculas y minúsculas
- Sintaxis para la declaración

```
var miVariable;
let nombre, nombre2, nombre3;
```

MEDAC

Variables

- var miVariable;

Ámbito de función o global.

- Si una variable var se declara dentro de una función, solo es accesible dentro de esa función
- Si una variable var se declara fuera de una función será accesible en todo el ámbito global
- Incluso si se declara dentro de un bloque (if, for…) esa variable sigue pudiendo ser accedida desde fuera de ese bloque

```
function testVar() {
  if (true) {
    var x = 10;
  }
  console.log(x); // 10 (var ignora el bloque)
}
testVar();
```

MEDAC

Variables

- let miVariable;

Ámbito de bloque

- Las variables declaradas con let solo son accesibles dentro del bloque donde se declaran (dentro de llaves {} como en if, for, while, etc.).
- Esto evita el acceso accidental fuera del bloque.

```
function testLet() {
  if (true) {
    let y = 20;
  }
  console.log(y); // Error: y is not defined (porque let respeta el bloque)
}
testLet();
```

MEDAC

Asignación

- Javascript tiene un sistema de tipado débil, es decir, no tenemos que declarar el tipo de variable antes de su utilización. El motor de Javascript averigua el tipo de una variable cuando se está ejecutando.
- A diferencia de otros lenguajes como JAVA, podemos asignar valores de diferente tipo a nuestras variables cuando nos haga falta. Si bien esto es “legal” en Javascript, no es recomendable a nivel de coherencia en el código y se considera una mala costumbre de programación.

```
var revisado = true;
var edad = 32;

revisado = "hola";
edad = 57;
```

MEDAC

Tipos de datos primitivos

- Number: Números enteros o reales, positivos o negativos. Solo existe un tipo de dato para todo lo que sea un número. Podemos trabajar además en tres bases:
  - Base 10: sistema decimal habitual
  - Base 8: octal (dígitos del 0 al 7). Se representan comenzando por 0. P Ej. 045
  - Base 16: hexadecimial (0-9, A-F). Se representan comanzando por 0x Ej. 0x3EF

Tipos de datos

- Number:
  - Existen 3 números especiales
    - Infinity: representa infinito (resultado cuando nos salimos en una operación de rango 64 bits. P ej 10 elevado a 100 ó división por 0)
    - -Infinity: infinito negativo
    - NaN: “Not a Number”: Cuando realizamos operaciones numéricas con resultados anómalos (0/0, Infinito-Infinito, dividir un número entre una cadena…)

Tipos de datos

- Boolean: true/false
- String: Cadenas de caracteres que representa texto independientemente de su longitud. Se representa entre “ (no existe el tipo char)
    - Caracteres de escape: caracteres especiales que sirven para cambiar el comportamiento en una cadena de texto

Salto de línea	\n
Comilla simple	\'
Comilla doble	\"
Tabulador	\t
Retorno de carro	\r
Avance de página	\f

Retroceder espacio	\b
Contrabarra	\\

MEDAC

Template Strings

Es una nueva manera de definir las cadenas de caracteres que aporta una serie de ventajas con respecto a la forma clásica. En lugar de “ vamos a usar la tilde invertida `

```
var otroTexto = `Pepe se va a pescar`;
var masTexto = `23%%$ Letras & *--*`;
```

- Ahorramos caracteres de escape: podemos usar “ y ‘ sin necesidad de usar caracteres adicionales

MEDAC

Template Strings

- Podemos usar el salto de línea como elemento de la cadena y será respetado

```
var queHacer =
  `comprar aceite
   comprar filetes
   calentar la plancha
   freír cebolla
  `;
```

Template Strings

- Interpolación de variables: podemos colocar el nombre de una variable dentro de la cadena usando la nomenclatura ${nombre_variable} y automáticamente será sustituido por su valor correspondiente.

```
var nombre = "Sr. Hormiga";
var asignatura = "Programación JS";
var texto = `El profesor de la asignatura ${asignatura}
será: ${nombre}`;

console.log(texto);
```

Valores vacíos

En Javascript existen dos valores especiales que se usan para indicar la ausencia de un valor significativo: undefined y null

- Undefined y null : es el “valor” que tendrán por defecto todas aquellas variables que se definan y a las que no se les asigne ningún valor significativo. Significan lo mismo. Existen dos valores para representar lo mismo (error en el diseño original de Javascript)

Conversión de tipos de datos

En JavaScript la conversión de tipos es automática la mayor parte del tiempo y se realiza correctamente. Por ejemplo, cuando mostramos un valor por pantalla, el lenguaje es capaz de convertir la variable a cadena caracteres.

Cuando JavaScript no pueda realizar la conversión de tipos, es necesario acudir a la conversión explícita. Al usar este tipo de conversión, el programador es el responsable del resultado obtenido.

```
var age = Number("Valor textual");
alert(age); // NaN, conversion
```

Código 4. Error de conversión

MEDAC

Conversión de tipos de datos

```
alert( Boolean(1) ); // true
alert( Boolean(0) ); // false
alert( Boolean("hola") ); // true
alert( Boolean("") ); // false
```

Código 5. Conversión de números, cadena y boolean.

MEDAC

Instituto Oficial de Formación Profesional