

U1. Navegador.

Lenguajes Cliente y Sintaxis Básica

Desarrollo Web en Entorno Cliente

Índice

- Características Javascript
- Palabras reservadas
- Tipos de dato
- Variables
- Operadores
- Sentencias condicionales
- Bucles

Características de Javascript

- Javascript es un lenguaje de script que se emplea mayoritariamente para la creación de páginas web con contenido dinámico e interactivo.
- Se trata de un lenguaje interpretado directamente desde el cliente que no requiere compilación.
- Su sintaxis se asemeja a la de C++ y Java pero en Javascript los objetos utilizan herencia basada en prototipos.

Características Javascript

- Este lenguaje fue creado en 1995 por Netscape y se estandarizó en 1997 dando lugar a la norma ECMA-262, conocida como ECMAScript.
- Desde el 2012, todos los navegadores modernos soportan completamente ECMAScript 5.1.
- Los navegadores más antiguos soportan por lo menos ECMAScript 3.
- Cada navegador implementa “su propia versión” del estándar ECMA a través de su motor de Javascript por lo que algunos scripts pueden comportarse de diferente manera dependiendo de qué navegador empleemos.

Características Javascript

- Se ejecuta dentro del navegador por lo que se encuentra limitado al entorno del documento en el que se incluye. Eso quiere decir que:
 - No puede cerrar ventanas que no haya abierto.
 - No puede acceder a recursos del sistema (ficheros, configuración, etc).
 - No puede modificar preferencias del navegador, etc.

Integración de Javascript en HTML

- **JavaScript en el mismo documento HTML.**
 - Uso de unas etiquetas predefinidas para marcar el texto (<script> y </script>).
 - Puede incluirse en cualquier parte del documento, aunque se recomienda que se defina dentro de la cabecera del documento HTML.
 - Esta técnica suele utilizarse cuando se definen instrucciones que se referenciarán desde cualquier parte del documento o cuando se definen funciones con fragmentos de código genéricos.

Integración de Javascript en HTML

- **JavaScript en un archivo externo.**
 - Las mismas instrucciones de JavaScript que se incluyen entre un bloque `<script></script>` pueden almacenarse en un fichero externo con extensión `.js`.
 - La forma de acceder y enlazar esos ficheros `.js` con el documento HTML/XHTML es a través de la propia etiqueta `<script>`.
 - No existe un límite en el número de ficheros `.js` que pueden enlazarse en un mismo documento HTML/XHTML.

Integración de Javascript en HTML

- **JavaScript en elementos HTML.**
 - Consiste en insertar fragmentos de JavaScript dentro de atributos de etiquetas HTML de la página.
 - Forma de controlar los eventos que suceden asociados a un elemento HTML concreto.
 - Principal desventaja: el mantenimiento y modificación del código puede resultar más complicado.

Soporte JS navegadores

- Actualmente, la amplia mayoría de navegadores soportan JavaScript no obstante, existe la posibilidad de que algunos usuarios bloqueen el uso de JavaScript porque creen que así navegan de forma más segura.
- Si la página web requiere JavaScript para su correcto funcionamiento, es posible incluir un aviso al usuario indicándole que debería activar JavaScript para disfrutar de la funcionalidad completa de la página.

Como habilitar JavaScript en tu navegador

Hoy en día casi todas las páginas web contienen JavaScript, un lenguaje de programación que se ejecuta en el navegador web del visitante. JavaScript permite crear funcionalidades específicas en las páginas web y si por alguna razón es deshabilitado, el contenido o la funcionalidad de la página puede quedar limitada o no disponible. Aquí puedes encontrar las instrucciones sobre cómo habilitar (activar) JavaScript en cinco de los navegadores más utilizados.



Internet Explorer



Mozilla Firefox



Google Chrome



Opera



Apple Safari

Soporte JS navegadores

- Para ello, se emplea la etiqueta [<noscript>](#) que se incluye en <head> (en HTML5 también es posible ponerla en <body>).

Ejemplo:

```
<head>
  <noscript>
    <p>Bienvenido a Mi Sitio</p>
    <p>La página que estás viendo requiere
    para su funcionamiento el uso de
    JavaScript. Si lo has deshabilitado
    intencionadamente, por favor vuelve a
    activarlo.</p>
  </noscript>
</head>
```

Actualmente, [noscript está en desuso](#) y en su lugar se visualizan u ocultan elementos dependiendo de si requieren o no JS.

Características Javascript

- Los espacios sobrantes y las líneas en blanco no se tienen en cuenta.
- Es Case Sensitive: distingue mayúsculas de minúsculas.
- No es obligatorio acabar las sentencias con ; pero es recomendable.
- Los comentarios son:
Línea: //
Multilinea: /*...*/

Características JavaScript

- Tabulación y saltos de línea – Diferencias:

```
<script
type="text/javascript">va
r i,j=0;
for (i=0;i<5;i++){
alert("Variable i: "+i);
for (j=0;j<5;j++){ if
(i%2==0){
document.write
(i + "-" + j +
"<br>");}}}
```

```
<script type="text/javascript">
  var i,j=0;
  for (i=0;i<5;i++){
    alert("Variable i: "+i;
    for (j=0;j<5;j++){
      if (i%2==0){
        document.write(i + "-" + j + "<br>");
      }
    }
  }
</script>
```

Características de Javascript

- Es un **lenguaje débilmente tipado**. Es decir, al crear una variable, no es necesario indicar el tipo de dato que almacenará.
 - Permite definir variables cuando se inicializan sin declararlas.
 - Una misma variable puede almacenar diferentes tipos de datos durante la ejecución del script. Es decir, el tipo de variable se asigna y modifica dinámicamente.

Palabras reservadas

- Son aquellas que no podemos emplear con libertad porque se utilizan en la propia sintaxis del lenguaje.
- En Javascript son:
break, case, catch, class, continue, default, delete, do, else, finally, for, function, if, in, instanceof, new, return, switch, this, throw, try, typeof, var, void, while, with.
- En www.ecmascript.org es posible consultar todas las palabras reservadas de JavaScript.

Tipos de datos

- Los tipos de datos especifican qué tipo de valor se guardará en una determinada variable.
- Los tres tipos de datos primitivos de JavaScript son:
 - Números.
 - Cadenas de texto.
 - Valores booleanos.

Tipos de datos

- Números:
 - En JavaScript existe sólo un tipo de dato numérico.
 - Todos los números se representan a través del formato de punto flotante de 64 bits.
 - Este formato es el llamado `double` en los lenguajes Java o C++.

Tipos de datos

- Cadenas de texto:
 - El tipo de datos para representar cadenas de texto se llama `string`.
 - Se pueden representar letras, dígitos, signos de puntuación o cualquier otro carácter de Unicode.
 - La cadena de caracteres se debe definir entre comillas dobles o comillas simples.

Tipos de datos

- Cadenas de texto - Secuencias de escape:

Secuencia de escape	Resultado
\\	Barra invertida
\'	Comilla simple
\"	Comillas dobles
\n	Salto de línea
\t	Tabulación horizontal
\v	Tabulación vertical
\f	Salto de página
\r	Retorno de carro
\b	Retroceso

Tipos de datos

- Valores booleanos:
 - También conocido como valor lógico.
 - Sólo admite dos valores: `true` o `false`.
 - Es muy útil a la hora de evaluar expresiones lógicas o verificar condiciones.

Variables

- Se pueden definir como zonas de la memoria de un ordenador que se identifican con un nombre y en las cuales se almacenan ciertos datos.
- El desarrollo de un script conlleva:
 - Declaración de variables.
 - Inicialización de variables.

Variables

- Declaración de variables:
 - Se declaran mediante la palabra clave `var` seguida por el nombre que se quiera dar a la variable.
 - `var mi_variable;`
 - Es posible declarar más de una variable en una sola línea.
 - `var mi_variable1, mi_variable2;`

Variables

- Por defecto, todas las variables son globales. Es decir, si no definimos un ámbito para la variable se ubica en un espacio de nombres común denominado global object.
- En general, el uso de variables globales no es una buena práctica

Variables

- Inicialización de variables:
 - Se puede asignar un valor a una variable de tres formas:
 - Asignación directa de un valor concreto.
 - Asignación indirecta a través de un cálculo en el que se implican a otras variables o constantes.
 - Asignación a través de la solicitud del valor al usuario del programa.
 - Ejemplos:
 - `var mi_variable_1 = 30;`
 - `var mi_variable_2 = mi_variable_1 + 10;`
 - `var mi_variable_3 = prompt('Introduce un valor:');`

Operadores

- JavaScript utiliza principalmente cinco tipo de operadores:
 - Aritméticos.
 - Lógicos.
 - De asignación.
 - De comparación.
 - Condicionales.

Operadores

- Operadores aritméticos:
 - Permiten realizar cálculos elementales entre variables numéricas.

Operador	Nombre
+	Suma
-	Resta
*	Multiplicación
/	División
%	Módulo
++	Incremento
--	Decremento

Operadores

- Operadores lógicos:
 - Combinan diferentes expresiones lógicas con el fin de evaluar si el resultado de dicha combinación es verdadero o falso.

Operador	Nombre
& &	Y
	O
!	No

Operadores

- Operadores de asignación:
 - Permiten obtener métodos abreviados para evitar escribir dos veces la variable que se encuentra a la izquierda del operador.

Operador	Nombre
<code>+=</code>	Suma y asigna
<code>-=</code>	Resta y asigna
<code>*=</code>	Multiplica y asigna
<code>/=</code>	Divide y asigna
<code>%=</code>	Módulo y asigna

Operadores

- Operadores de comparación:
 - Permiten comparar todo tipo de variables y devuelve un valor booleano.

Operador	Nombre
<	Menor que
<=	Menor o igual que
==	Igual
>	Mayor que
>=	Mayor o igual que
!=	Diferente
===	Estrictamente igual
!==	Estrictamente diferente

Operadores

- Operadores condicionales:
 - Permite indicar al navegador que ejecute una acción en concreto después de evaluar una expresión.

Operador	Nombre
? :	Condicional

- Ejemplo:

```
<script type="text/javascript">
  var dividendo = prompt("Introduce el dividendo: ");
  var divisor = prompt("Introduce el divisor: ");
  var resultado;
  divisor != 0 ? resultado = dividendo/divisor :
    alert("No es posible la división por cero");
    alert("El resultado es: " + resultado);
</script>
```

Sentencias condicionales

- Con las sentencias condicionales se puede gestionar la toma de decisiones y el posterior resultado por parte del navegador.
- Dichas sentencias evalúan condiciones y ejecutan ciertas instrucciones en base al resultado de la condición.
- Las sentencias condicionales en JavaScript son:
 - `if`.
 - `switch`.
 - `while`.
 - `for`.

Sentencias condicionales

- `if` – sintaxis (1):

```
if (expresión) {  
    instrucciones  
}
```

Sentencias condicionales

- `if` – sintaxis (2):

```
if (expresión) {  
    instrucciones_si_true  
} else {  
    instrucciones_si_false  
}
```


Sentencias condicionales

- `if` – sintaxis (3):

```
if (expresión_1) {  
    instrucciones_1  
} else if (expresión_2) {  
    instrucciones_2  
} else {  
    instrucciones_3  
}
```

Sentencias condicionales

- `switch` – sintaxis:

```
switch (expresión) {  
    case valor1:  
        instrucciones a ejecutar si expresión = valor1  
        break;  
    case valor2:  
        instrucciones a ejecutar si expresión = valor2  
        break;  
    case valor3:  
        instrucciones a ejecutar si expresión = valor3  
        break  
    default:  
        instrucciones a ejecutar si expresión es diferente a  
        los valores anteriores  
}
```

Bucle while

- `while` – sintaxis:

(1)

```
while (expresión) {  
    instrucciones  
}
```

(2)

```
do {  
    instrucciones  
} while (expresión)
```

Bucle for

- `for` – sintaxis:

```
for(valor_inicial_variable;  
    expresión_condicional;  
    incremento_o_decremento_var) {  
    cuerpo_del_bucle  
}
```