



MEDAC

Instituto Oficial de Formación Profesional

EL LENGUAJE DE PROGRAMACIÓN DE CLIENTES

LENGUAJE DE MARCAS

1º DAM

Curso 2021-2022

TEMA 2. LENGUAJES Y HERRAMIENTAS DE PROGRAMACIÓN EN CLIENTES WEB

APARTADO 2.FUNDAMENTOS DEL LENGUAJE



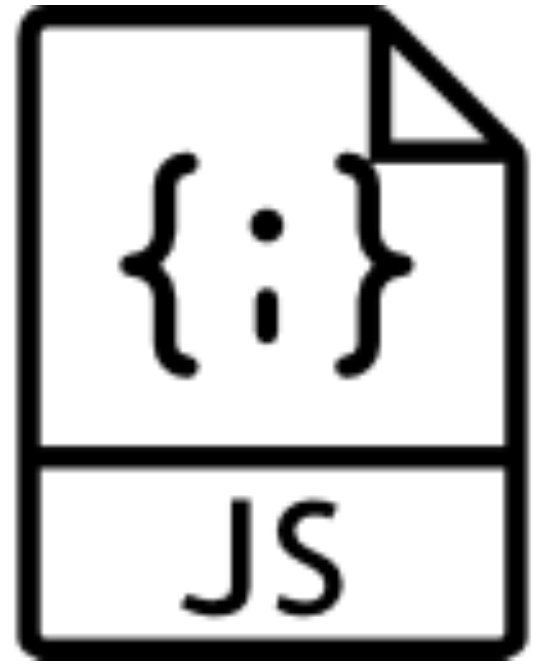
¿Qué es JavaScript?

JavaScript es un **lenguaje de programación de scripts** (secuencia de comandos) **orientado a objetos**. Es un lenguaje interpretado por el navegador, que nos permite crear páginas dinámicas en las que el usuario puede interactuar fácilmente.



TEMA 2. LENGUAJES Y HERRAMIENTAS DE PROGRAMACIÓN EN CLIENTES WEB

APARTADO 2.1 RELACIÓN CON HTML



Para que podamos introducir código JavaScript en un documento web disponemos de las siguientes alternativas:

- **Embeber el código en el documento HTML:** de esta forma podemos añadir el código JavaScript usando las etiqueta `<script></script>`.
- **Mediante archivos externos:** así podemos apoyarnos en archivos externos que sólo contendrían código JavaScript.
- **A través de elementos HTML:** esto se suele usar para añadir referencias a funcionalidades de JavaScript.



EL OPERADOR DE ASIGNACIÓN

En JavaScript funciona igual que en otros lenguajes. En una asignación contamos con 2 partes: a la **izquierda** tenemos una **variable** y a la **derecha** el **valor** asignado.

Aquí podemos observar tres asignaciones y tres declaraciones de variables.

```
var x;  
var y = 6;  
x = 5  
var z = x + y;  
Código 1. Asignaciones
```

- En la línea 2 se declara y se asigna a la variable Y el valor 6.
- En la línea 3 se asigna a la variable X el valor 5.
- En la línea 4 se asigna a la variable Z el resultado de sumar X e Y.



TIPOS DE DATOS

En JavaScript se definen distintos tipos de datos:

- Números.
- Cadenas de caracteres.
- Objetos, etc.

```
var temp = 16 + "Gato";  
Código 2. Distintos tipos de datos
```

Aquí podemos ver que se declara la variable **temp** y se le asigna el resultado de sumar **16** y **<<Gato>>**.

En este caso, podemos dudar del resultado, ya que **16** es un valor entero y **<<Gato>>** es una cadena de caracteres.

La forma en la que JavaScript interpreta esto genera la siguiente cadena **<<16Gato>>**.

TEMA 2. LENGUAJES Y HERRAMIENTAS DE PROGRAMACIÓN EN CLIENTES WEB

APARTADO 2.2 ASPECTOS BÁSICOS DEL LENGUAJE

TIPOS DE DATOS

Para programar cálculos podemos hacer:

```
var resultado= 3+2;  
alert(resultado);
```

→ Muestra 5

```
var num1=3, num2=2, resultado  
resultado= num1*num2  
alert(resultado);
```

→ Muestra 6

```
var divisor=3, resultado1, resultado2, resultado3;  
resultado1=(16+8)/2-2;  
resultado2= resultado1/ divisor;  
resultado3=resultado1%divisor;  
alert(resultado2);  
alert(resultado3);
```

→ 10

→ 3.33

Resto de la división: 1

```
var numero=3;  
numero+= 5  
alert(numero);
```

→ 8



TEMA 2. LENGUAJES Y HERRAMIENTAS DE PROGRAMACIÓN EN CLIENTES WEB

APARTADO 2.2 ASPECTOS BÁSICOS DEL LENGUAJE



TIPOS DE DATOS

Concatenación:

```
var texto= "Hola";  
texto+="tu"  
alert(texto);
```

→ Hola tu

También usaremos para mostrar resultados el `document.write("Lo que queramos");`

Para interactuar con el usuario usaremos **prompt()**

```
var NombreUsuario=prompt('Introduce el nombre:');  
alert(NombreUsuario);
```

Esta página dice

Bea

Aceptar

Esta página dice

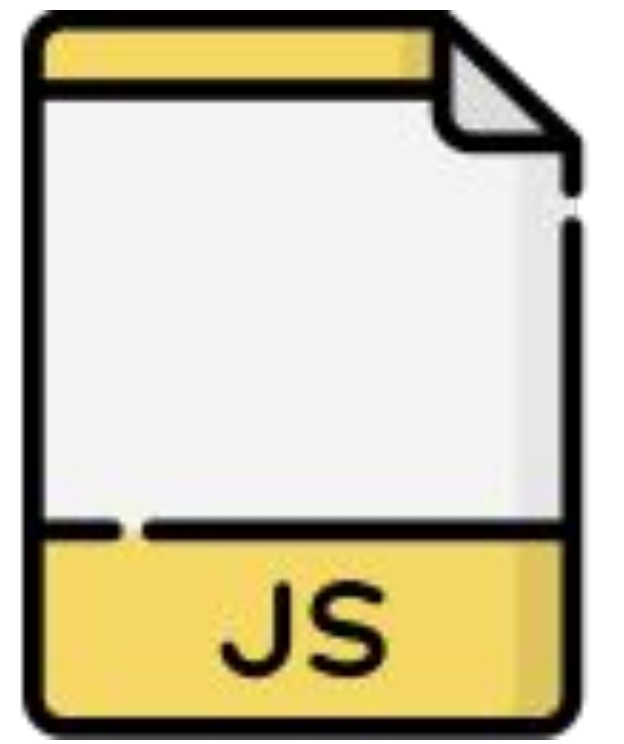
Introduce el nombre:

Bea

Aceptar Cancelar

TEMA 2. LENGUAJES Y HERRAMIENTAS DE PROGRAMACIÓN EN CLIENTES WEB

APARTADO 4. SISTEMA DE TIPADO



JavaScript incluye un sistema de tipado débil, es por esto que **no es necesario** declarar el **tipo** de variable antes de usarla, ya que el motor de JavaScript puede averiguar el tipo de una variable cuando se esté ejecutando. Por esto podemos tener la misma variable con distintos tipos.

En el sistema de tipos existen los tipos “primitivos”, que son los que se proporcionan con el lenguaje. En el caso de JavaScript, los tipos primitivos son:

- Boolean: puede almacenar valores lógicos, es decir, true o false.
- Null: este tipo contiene el valor nulo.

TEMA 2. LENGUAJES Y HERRAMIENTAS DE PROGRAMACIÓN EN CLIENTES WEB

APARTADO 4. SISTEMA DE TIPADO

- Undefined: es el tipo que tienen aquellas variables que no hayan sido definidas previamente.
- Number: este tipo puede almacenar cualquier número, sin diferenciar entre enteros, flotantes,...
- String: nos permite almacenar cadenas de caracteres. Una característica de JavaScript es que las cadenas de caracteres son inmutables, es decir, una vez que tienen valor no puede modificarse. Pero, si es posible crear otra cadena que modifique la actual.



Como JavaScript declara variables sin tipo, es posible que en un momento determinado no sepamos cómo operar con esta variable. En este caso JavaScript nos proporciona el operador **typeof**, que si lo usamos podemos saber el tipo de la variable.

TEMA 2. LENGUAJES Y HERRAMIENTAS DE PROGRAMACIÓN EN CLIENTES WEB

APARTADO 4.1 CONVERSIÓN ENTRE TIPOS DE DATOS

En JavaScript la conversión de tipos es automática la mayor parte del tiempo y se realiza correctamente.

EXAMPLE

Por ejemplo: cuando mostramos un valor por pantalla, el lenguaje es capaz de convertir la variable a cadena de caracteres.

Otras veces, el operador que estemos usando fijará el tipo de dato resuelto.

Cuando JavaScript no pueda realizar la conversión de tipos , tendremos que acudir a la conversión explícita. Al usar este tipo de conversión, somos nosotros como programadores, los responsables del resultado obtenido.

```
var age = Number("Valor textual");  
alert(age); // NaN, conversion  
Código 4. Error de conversión
```

Aquí no es posible convertir un valor textual a un número. Por esta razón se produce un error en la conversión de tipos.

La conversión de tipos de valores lógicos se considera más sencilla.

TEMA 2. LENGUAJES Y HERRAMIENTAS DE PROGRAMACIÓN EN CLIENTES WEB

APARTADO 4.1 CONVERSIÓN ENTRE TIPOS DE DATOS

Aquí se muestra la conversión de números y cadena de caracteres a valores lógicos. Como vemos no se produce error porque la conversión siempre es posible. Aunque esto no significa que el valor sea el esperado.

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

Código 5. Conversión de números, cadena y boolean.

EXAMPLE

Por ejemplo, la conversión 1 devuelve “verdadero” y la 0 devuelve “falso”. Además , la conversión de “hola” devuelve verdadero y la conversión de la cadena vacía devuelve “falso. En el caso de los valores numéricos puede tener sentido pero, la conversión de cadenas puede inducir a error.

TEMA 2. LENGUAJES Y HERRAMIENTAS DE PROGRAMACIÓN EN CLIENTES WEB

APARTADO 5. COMENTARIOS

Para comentar el código, contamos con los <<comentarios>>. JavaScript proporciona dos tipos de comentarios:

→ Comentario de línea: es un comentario que se refiere a una única línea.

```
var z = 2 ; // Declara z y le asigna 2.  
var y = z + 2; // Declara y, y le asigna z + 2  
Código 6. Comentarios de línea
```

→ Comentario multilínea: es un comentario capaz de generar bloques.

```
/* 1. Declara z y le asigna 2.  
2. Declara y, y le asigna z + 2  
*/  
var z = 2 ;  
var y = z + 2;  
Código 7. Comentario en bloque
```


TEMA 3. OPERADORES DEL LENGUAJE Y BUCLES

APARTADO 5. OPERADORES ARITMÉTICOS Y DE COMPARACIÓN

Los **operadores aritméticos** son los que nos permiten realizar operaciones con números, los más usados en JavaScript son los siguientes:

Operador	Descripción
+	Operador suma
-	Operador resta
*	Operador de producto
**	Operador de potencia
/	Operador división
%	Operador módulo.
++	Operador de incremento
--	Operador de decremento

TEMA 3. OPERADORES DEL LENGUAJE Y BUCLES

APARTADO 5. OPERADORES ARITMÉTICOS Y DE COMPARACIÓN

Los **operadores de comparación** nos van a permitir realizar comparaciones entre distintas expresiones o variables. Estos son los más usados en JavaScript:

Operador	Descripción
==	Operador de igualdad
===	Igualdad de valor y de tipo
!=	Operador distinto
!==	Operador distinto valor o tipo
>	Mayor que
<	Menor que
>=	Mayor o igual que
<=	Menor o igual que

TEMA 3. OPERADORES DEL LENGUAJE Y BUCLES

APARTADO 6. OPERADORES DE ASIGNACIÓN Y LÓGICOS

También podemos usar la asignación para hacer varias operaciones en una única sentencia:

Operador	Ejemplo	Equivalencia
=	A = B	A = B
+=	A += B	A = A + B
-=	A -= B	A = A – B
*=	A *= B	A = A * B
/=	A /= B	A = A / B
%=	A %= B	A = A % B
**=	A **= B	A = A ** B

Operador	Descripción
&&	Y lógico
	O lógico
!	Negación lógica



Palabras reservadas en JavaScript

Palabra reservada	Descripción
function	Permite declarar una función
for	Permite crear un bloque que se repita tantas veces como la condición sea cierta
break	Permite salir de un bloque ejecución
switch	Permite elegir qué bloque de sentencias ejecutar a partir de una condición
do...while	Ejecuta un bloque de sentencias y lo repite tantas veces como la condición sea cierta
If...else	Permite ejecutar un bloque de sentencias u otro dependiendo de una condición

Tabla 1. Principales palabras clave de JavaScript





Sentencia if-else

Esta sentencia tiene dos partes: una cuando se cumple la condición y otra en caso de no cumplirse que es opcional.

```
var result = 0
if (today > 10) {
    result++;
}else{
    result--;
}
```

Código 4. Sentencia if-else

En este ejemplo, podemos ver que **result** valdrá uno si se cumple la condición o -1 si no se cumple la condición.

Como en el resto de lenguajes la sentencia condicional la podemos anidar y tener varias condiciones unas tras otras.

TEMA 3. OPERADORES DEL LENGUAJE Y BUCLES

APARTADO 8. CONDICIONALES

Sentencia switch

Permite realizar distintas acciones acordes a diferentes condiciones. Se recomienda usar **switch** cuando el nivel de anidamiento en sentencias condicionales simples es muy elevado.

```
switch (today) {  
  case 0:  
    day = "Lunes";  
    break;  
  case 1:  
    day = "Martes";  
    break;  
  default: {  
    day = "error";  
    break;  
  }  
}
```

Código 5. Ejemplo de funcionamiento sentencia switch

Dependiendo del valor que tenga la variable del bloque switch se ejecutará el caso asociado.

En este caso están definidas las opciones 0 y 1 y, en caso contrario, el valor del día será error.



TEMA 3. OPERADORES DEL LENGUAJE Y BUCLES

APARTADO 3. SENTENCIAS

Cadenas

En JavaScript las cadenas se utilizan para manejar series de caracteres. Las cadenas van definidas con comillas dobles/simples al inicio y al final.



```
var moto1 = "Yamaha TMAX"; // Dobles comillas  
var moto2 = 'Yamaha TMAX'; // Comillas simples  
Código 1. Cadenas de caracteres
```

Bloques de código

En JavaScript las sentencias pueden ir en bloques de código que están delimitados por llaves. Existen bloques anónimos y bloques que tienen un nombre asignado.

```
function test(){  
  var a, b, c;  
  a = 5;  
  b = 6;  
  c = a + b;  
}  
Código 2. Bloque de código con nombre
```

TEMA 3. OPERADORES DEL LENGUAJE Y BUCLES

APARTADO 9. BUCLES



¿Cuál es el bucle más común en todos los lenguajes de programación?

BUCLE FOR

```
for (i = 0; i < 5; i++) {  
    text += "El número es " + i + "<br>";  
}  
Código 6. Bucle for
```

En este ejemplo se ejecuta la primera sentencia que pone el contador a 0, luego comprueba que $i < 5$ y finalmente cada vez que se ejecute el bloque dentro del bucle se incrementará en 1 el contador.

TEMA 3. OPERADORES DEL LENGUAJE Y BUCLES

APARTADO 9. BUCLES

BUCLE WHILE



```
while (i < 10) {  
    text += "El número es " + i;  
    i++;  
}
```

Código 7. Ejemplo texto de tipo de bucle

En este ejemplo se ejecutará el bloque de código mientras se cumpla la condición de que $i < 10$.

TEMA 3. OPERADORES DEL LENGUAJE Y BUCLES

APARTADO 9. BUCLES

BUCLE DO-WHILE



```
do {  
    text += "El número es " + i;  
    i++;  
}  
while (i < 10);
```

Código 8. Ejemplo bucle do-while

En este ejemplo se ejecuta el bucle siempre al menos una vez ya que la condición se comprueba una vez ejecutado el bloque. El resto de funcionamiento es igual que el bucle while.

TEMA 3. OPERADORES DEL LENGUAJE Y BUCLES

APARTADO 9. BUCLES

parseInt(x)

Convierte un argumento de tipo cadena y devuelve un entero.

En JavaScript, por ejemplo, si pedimos dos números por pantalla y los queremos sumar el “+” los concatena, entonces debemos especificar que son números para que los sume por ejemplo usando el `parseInt()`, `parseFloat()`,....



MEDAC

Instituto Oficial de Formación Profesional

- Muchas gracias -
