

*Manejo de la
sintaxis del
lenguaje*

--

*Asignaciones,
operadores y
comentarios*

Asignaciones	4
Asignación directa e indirecta.	4
Asignaciones mediante solicitud de del valor	4
ALERT	4
CONFIRM	5
PROMPT	6
Conversión de tipos	6
Conversión automática	6
Forzar conversión	7
Función isNaN	7
Operadores	7
Operadores aritméticos	7
Operadores lógicos	8
Operadores de asignación	8
Operadores de comparación	9
Operadores condicionales	9
Comentarios en JavaScript	9
Comentario de una línea	9
Comentario en múltiples líneas	10
Inserción automática del ;	10

1. Asignaciones

Podemos realizar la asignación de 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.

A continuación, podemos ver los ejemplos respectivos de cada caso:

```
var mi_variable_1 = 30;  
var mi_variable_2 = mi_variable_1 + 10;  
var mi_variable_3 = prompt('Introduce un valor:');
```

a. Asignación directa e indirecta.

No presentan mayor problema. Simplemente indicar que se pueden utilizar los operadores de asignación, utilizar la barra escape como por ejemplo poder utilizar símbolos de la tabla unicode.

Operador	Nombre	Descripción
+=	Suma y asigna	Ejecuta una suma y asigna el valor al operando de la izquierda.
-=	Resta y asigna	Ejecuta una resta y asigna el valor al operando de la izquierda.
*=	Multiplica y asigna	Ejecuta una multiplicación y asigna el valor al operando de la izquierda.
/=	Divide y asigna	Ejecuta una división y asigna el valor al operando de la izquierda.
%=	Módulo y asigna	Ejecuta el módulo y asigna el valor al operando de la izquierda.

```
console.log("/u{1F48B}");
```

<https://unicode-table.com/es/search/?q=1f48c>

b. Asignaciones mediante solicitud de del valor

Para ello disponemos de 3 tipos de cuadros de diálogo.

i. ALERT

Muestra un mensaje con un botón de aceptar cuya única función es cerrar el cuadro de diálogo y continuar con el contenido de la página. Se puede incluir tanto en <head> como en <body>

```
<script>
```

```
alert("Esto es un mensaje, no hace nada más");

</script>

<html>
<head>
    <title>Ejemplo diálogos javascript</title>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
</script>
alert("Esto es un mensaje, no hace nada más");
</script>
</head>
<body>
    <h1>
        ESTO ES UNA PÁGINA DE EJEMPLO
    </h1>
</body>
</html>
```

ii. CONFIRM

Muestra un mensaje pero con dos botones: aceptar y cancelar. Podemos programarlo para que realice una acción diferente según el botón que decida pulsar.

```
<script>
var confirmacion = confirm("Pulsa el botón que quieras");
if(confirmacion){
    alert("Has pulsado aceptar");
} else {
    alert("Has pulsado cancelar");
}
</script>

<html>
<head>
    <title>Ejemplo diálogos javascript</title>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <script>
        if(confirm("Pulsa el botón que quieras")){
            alert("Has pulsado aceptar");
        } else {
            alert("Has pulsado cancelar");
        }
    </script>
</head>
<body>
    <h1>
```

```
ESTO ES UNA PÁGINA DE EJEMPLO
</h1>
</body>
</html>
```

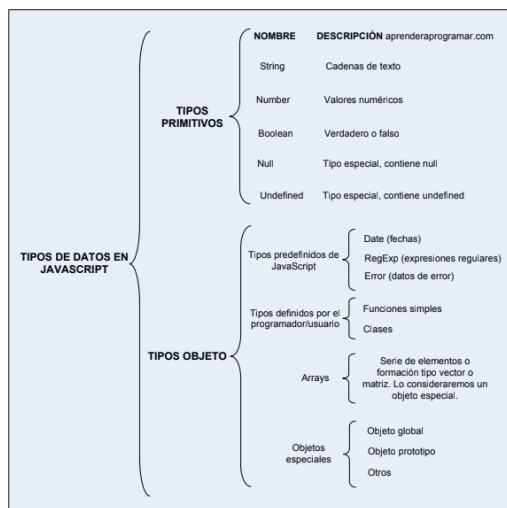
iii. PROMPT

Este último tipo también tiene botón para aceptar y cancelar, pero además incluye un cuadro de texto en el que el usuario puede introducir información. El segundo parámetro es opcional. Permite indicar el valor por defecto que se asignará a la variable si el usuario pulsa aceptar sin escribir nada en la caja de texto.

```
<script>
let dmail = prompt("Escribe tu dirección de correo", "correo@dominio.com");
if(mail != null){
    alert("Has escrito " + dmail);
} else {
    alert("No has escrito nada");
}
</script>
```

2. Conversión de tipos

Cuadro resumen de tipos de datos



JavaScript intenta convertir automáticamente los tipos de datos, pero es posible convertir un texto a número mediante las funciones `Number`, `parseInt` y `parseFloat`. Para convertir a texto disponemos de `String`.

a. Conversión automática

```
console.log('2' * 3); // 6
console.log('2' + 3); // 23
```

```
console.log('Hola' * 3) ;// NaN
```

b. Forzar conversión

```
let x='2';  
let y='3';  
console.log(Number((x) + Number(y)); //5  
console.log(x + y); //23  
console.log(Number("Hola")); //NaN
```

```
let x=2;  
let y=3;  
console.log(String(x) + String(y)); //23  
console.log(parseInt("1011",2); //11  
console.log(parseInt("22.5veces"); //22
```

c. Función isNaN

```
console.log(isNaN('Hola' * 3) ; // true  
console.log(isNaN('5' * 3) ; // false  
console.log(isNaN(NaN) ; // true
```

3. Operadores

a. Operadores aritméticos

Operador	Nombre	Descripción
+	Suma	Efectúa la suma entre los operandos.
-	Resta	Efectúa la resta entre los operandos.
*	Multiplicación	Efectúa la multiplicación entre los operandos.
/	División	Efectúa la división entre los operandos.
%	Módulo	Extrae la parte entera del resultado de la división entre los operandos.
++	Incremento	Permite incrementar un valor.
--	Decremento	Permite decrementar un valor.

b. Operadores lógicos

Operador	Nombre	Descripción
&&	Y	Ejecuta la operación booleana AND sobre los valores. Devuelve true solo si todos los valores son true. Devuelve false en caso contrario.
	O	Ejecuta la operación booleana OR sobre los valores. Devuelve true en el caso en que al menos uno de los valores sea true. Devuelve false en caso contrario.
!	No	Invierte el valor booleano de su operando.

c. Operadores de asignación

Operador	Nombre	Descripción
+=	Suma y asigna	Ejecuta una suma y asigna el valor al operando de la izquierda.
-=	Resta y asigna	Ejecuta una resta y asigna el valor al operando de la izquierda.
*=	Multiplica y asigna	Ejecuta una multiplicación y asigna el valor al operando de la izquierda.
/=	Divide y asigna	Ejecuta una división y asigna el valor al operando de la izquierda.
%=	Módulo y asigna	Ejecuta el módulo y asigna el valor al operando de la izquierda.

d. Operadores de comparación

Operador	Nombre	Descripción
<	Menor que	Verifica si el operando a la izquierda del operador es menor que el operando de la derecha. Devuelve <code>true</code> en ese caso o <code>false</code> en caso contrario.
<=	Menor o igual que	Verifica si el operando a la izquierda del operador es menor o igual que el operando de la derecha. Devuelve <code>true</code> en ese caso o <code>false</code> en caso contrario.
==	Igual	Verifica si los dos operandos son iguales. Devuelve <code>true</code> en ese caso o <code>false</code> en caso contrario.
>	Mayor que	Verifica si el operando a la izquierda del operador es mayor que el operando de la derecha. Devuelve <code>true</code> en ese caso o <code>false</code> en caso contrario.
>=	Mayor o igual que	Verifica si el operando a la izquierda del operador es mayor o igual que el operando de la derecha. Devuelve <code>true</code> en ese caso o <code>false</code> en caso contrario.
!=	Diferente	Verifica si los dos operandos son diferentes. Devuelve <code>true</code> en ese caso o <code>false</code> en caso contrario.
===	Estrictamente igual	Verifica si el operando a la izquierda del operador es igual y del mismo tipo de datos que el operando de la derecha. Devuelve <code>true</code> en ese caso o <code>false</code> en caso contrario.
!==	Estrictamente diferente	Verifica si el operando a la izquierda del operador es diferente y/o de tipo diferente que el operando de la derecha. Devuelve <code>true</code> en ese caso o <code>false</code> en caso contrario.

Ejemplo de comparación estricta:

("2"===2) devuelve false.

e. Operadores condicionales

Operador	Nombre	Descripción
?:	Condicional	Si la expresión antes del operador es verdadera, se utiliza el primer valor a la derecha. En caso contrario se utiliza el segundo valor a la derecha.

4. Comentarios en JavaScript

a. Comentario de una línea

Para agregar comentarios a una línea de código debemos anteponer `//`. Por ejemplo:

// Este es un comentario de una línea

b. Comentario en múltiples líneas

Si queremos agregar comentarios más extensos podemos hacer uso de `/* */`

```
/* Este es un comentario  
de varias  
líneas */
```

Uno de los usos más extendidos de los comentarios es para generar documentación, para clarificar el código para tu yo del futuro o para cualquier otra persona que use tu código.

5. Inserción automática del ;

En general las sentencias deben terminar con un punto y coma que delimita el final de una instrucción. No obstante, en caso de que “se olvide” insertar el punto y coma delimitador, el intérprete JavaScript lo insertará automáticamente siempre que le sea posible, facilitando que el código se ejecute.