

# SINTAXIS, PROPIEDADES Y MÉTODOS DE APOYO

Palabras reservadas más comunes	
Nombre	Uso
var	Utilizada para declarar una variable.
if .. else	Estructura condicional.
for	Estructura de repetición. Se ejecutará mientras la condición sea verdadera,
do ... while	Estructura de repetición. Se ejecutará mientras la condición sea verdadera.
switch	Sentencia condicional.
break	Termina un switch o un bucle.
continue	Sale del bucle y se coloca al comienzo de este.
function	Declara una función.
return	Sale de una función.
try – catch	Utilizadas para el manejo de excepciones.
let	Declaración de variable.
const	Declaración de constantes.

Categorías de operadores en JavaScript	
Tipo	Qué realizan
Comparación.	Comparan los valores de 2 operandos, devolviendo un resultado de true o false (se usan extensivamente en sentencias condicionales como if... else y en instrucciones bucle).  ==   !=   ===   !==   >=   <=
Aritméticos.	Unen dos operandos para producir un único valor que es el resultado de una operación aritmética u otra operación sobre ambos operandos.  +   -   *   /   %   ++   --   +valor   -valor
Asignación.	Asigna el valor a la derecha de la expresión a la variable que está a la izquierda.  =   +=   -=   *=   /=   %=   <=   >=   >>=   >>>=   &=    =   ^=   [ ]
Boolean.	Realizan operaciones booleanas aritméticas sobre uno o dos operandos booleanos.

	&&          !
Bit a Bit.	Realizan operaciones aritméticas o de desplazamiento de columna en las representaciones binarias de dos operandos.  &             ^      ~      <<      >>      >>>
Objeto.	Ayudan a los scripts a evaluar la herencia y capacidades de un objeto particular antes de que tengamos que invocar al objeto y sus propiedades o métodos.  .      []      ()      delete      in      instanceof      new      this
Misceláneos.	Operadores que tienen un comportamiento especial.  ,      ?:      typeof      void

Operadores de comparación en JavaScript			
Sintaxis	Nombre	Tipos de operandos	Resultados
==	Igualdad.	Todos.	Boolean.
!=	Distinto.	Todos.	Boolean.
====	Igualdad estricta.	Todos.	Boolean.
!==	Desigualdad estricta.	Todos.	Boolean.
>	Mayor que .	Todos.	Boolean.
>=	Mayor o igual que.	Todos.	Boolean.
<	Menor que.	Todos.	Boolean.
<=	Menor o igual que.	Todos.	Boolean.

Operadores aritméticos en JavaScript			
Sintaxis	Nombre	Tipos de Operando	Resultados
+	Más.	integer, float, string.	integer, float, string.
-	Menos.	integer, float.	integer, float.
*	Multiplicación.	integer, float.	integer, float.
/	División.	integer, float.	integer, float.
%	Módulo.	integer, float.	integer, float.
++	Incremento.	integer, float.	integer, float.
--	Decremento.	integer, float.	integer, float.
+valor	Positivo.	integer, float, string.	integer, float.
-valor	Negativo.	integer, float, string.	integer, float.

Operadores de asignación en JavaScript			
Sintaxis	Nombre	Ejemplo	Significado
=	Asignación.	x = y	x = y
+=	Sumar un valor.	x += y	x = x + y
-=	Substraer un valor.	x -= y	x = x - y
*=	Multiplicar un valor.	x *= y	x = x * y
/=	Dividir un valor.	x /= y	x = x / y
%=	Módulo de un valor.	x %= y	x = x % y
<<=	Desplazar bits a la izquierda.	x <<= y	x = x << y
>=	Desplazar bits a la derecha.	x >= y	x = x > y
>>=	Desplazar bits a la derecha rellenando con 0.	x >>= y	x = x >> y
>>>=	Desplazar bits a la derecha.	x >>>= y	x = x >>> y
&=	Operación AND bit a bit.	x &= y	x = x & y
=	Operación OR bit a bit.	x  = y	x = x   y
^=	Operación XOR bit a bit.	x ^= y	x = x ^ y
[ ]=	Desestructurando asignaciones.	[a,b]=[c,d]	a=c, b=d

Operadores de boolean en JavaScript			
Sintaxis	Nombre	Operandos	Resultados
&&	AND.	Boolean.	Boolean.
	OR.	Boolean.	Boolean.
!	Not.	Boolean.	Boolean.

Tabla de valores de verdad del operador AND			
Operando Izquierdo	Operador AND	Operando Derecho	Resultado
True	&&	True	True
True	&&	False	False
False	&&	True	False
False	&&	False	False

Tabla de valores de verdad del operador OR			
Operando Izquierdo	Operador OR	Operando Derecho	Resultado
True		True	True
True		False	True
False		True	True
False		False	False

### ? : (operador condicional)

Este operador condicional es la forma reducida de la expresión `if ... else`.

La sintaxis formal para este operador condicional es:

**condición ? expresión si se cumple la condición: expresión si no se cumple;**

## Condicionales y bucles.

### Construcción `if`

Sintaxis:

```
if (condición)
{
    // instrucciones a ejecutar si se cumple la condición
}
```

### Construcción `if ... else`

Sintaxis:

```
if (condición)
{
    // instrucciones a ejecutar si se cumple la condición
}
else
{
    // instrucciones a ejecutar si no se cumple la condición
}
```

### Sentencia switch.

Sintaxis:

```
switch (expresion) {
    case valor: sentencia
    break;
    case valor: sentencia
    break;
    case valor: sentencia
    break;
    case valor: sentencia
    break;
    default: sentencia
}
```

**Bucle for.**

Sintaxis:

```
for (expresión inicial; condición; incremento)
{
// Instrucciones a ejecutar dentro del bucle.
}
```

**Bucle while().**

Sintaxis:

```
while (condición)
{
// Instrucciones a ejecutar dentro del bucle.
}
```

**Bucle do ... while().**

Sintaxis:

```
do {
// Instrucciones a ejecutar dentro del bucle.
}while (condición);
```

Propiedades del objeto Number	
Propiedad	Descripción
constructor	Devuelve la función que creó el objeto Number.
MAX_VALUE	Devuelve el número más alto disponible en JavaScript.
MIN_VALUE	Devuelve el número más pequeño disponible en JavaScript.
MAX_SAFE_INTEGER	El valor máximo para realizar cálculos con seguridad de que los resultados son correctos en JavaScript.
MIN_SAFE_INTEGER	El valor mínimo para realizar cálculos con seguridad de que los resultados son correctos en JavaScript.
NaN	La propiedad <i>NaN</i> ( <i>Not a Number</i> ) indica que el valor de un número no es un número válido. <i>Number.isNaN(NaN)</i> ; sería la forma correcta de comprobarlo.
NEGATIVE_INFINITY	Representa a infinito negativo (se devuelve en caso de overflow).
POSITIVE_INFINITY	Representa a infinito positivo (se devuelve en caso de overflow).
prototype	Permite añadir nuestras propias propiedades y métodos a un objeto.