

CONVERSIÓN DE TIPOS DE DATOS

“

A través de funciones y métodos de Javascript, podemos alterar el **tipo de dato** que está almacenando una **variable**.

De esta forma, nos **aseguramos** de estar trabajando con un tipo de **dato determinado**.





1.

**AVERIGUANDO
EL TIPO DE DATO**

typeof

Usando el operador **typeof** podemos saber qué tipo de dato está almacenado en una variable.

typeof devuelve un **string** indicando el tipo de dato que almacena la variable.

typeof nombreVariable

operador

Se escribe antes del nombre de la variable.

variable

Nombre de la variable a la cual vamos a consultar el tipo de dato

OPERANDO CON **typeof**

{ }

```
var nombre = 'Dylan';  
console.log(typeof nombre); // devuelve 'string'
```

{ }

```
var llueve = true;  
console.log(typeof llueve); // devuelve 'boolean'
```

{ }

```
var edad = 23;  
console.log(typeof edad); // devuelve 'number'
```

{ }

```
var resultado;  
console.log(typeof resultado); // devuelve 'undefined'
```

2.

MÉTODOS DE CONVERSIÓN

Ahora sí, conozcamos
algunos de los métodos que
nos da Javascript para poder
convertir distintos tipos de
datos.

Number()

Es una **función** nativa de Javascript que convierte un string en un número, y puede devolver tanto **enteros** como **decimales**.

```
Number('32'); // devuelve 32  
Number('1.5'); // devuelve 1.5  
Number('-3.24'); // devuelve -3.24
```

¿CÓMO FUNCIONA `Number()`?

La función recibe un string y, en caso de tener **caracteres no numéricos**, devuelve NaN. Caso contrario, devuelve el **número**.

```
Number('34.2'); // devuelve 34.2
Number(' 5 '); // devuelve 5
Number('100 años'); // devuelve NaN
Number('4.5_G'); // devuelve NaN
```



parseInt()

Es una **función** nativa de Javascript que recibe como parámetro la cadena de caracteres que queremos parsear y **devuelve un entero**.

```
parseInt('12'); // devuelve 12
parseInt(' 3 '); // devuelve 3
parseInt('18.9'); // devuelve 18
parseInt('-3'); // devuelve -3
parseInt('n405'); // devuelve NaN
```

¿CÓMO FUNCIONA `parseInt()`?

La función va a extraer la parte numérica del string que recibe hasta que llegue a un dato no numérico.

{ }

```
parseInt('34-F-45'); //devuelve 34  
parseInt('1ºJuego'); //devuelve 1
```

Si el primer caracter que intenta extraer **no es numérico**, se corta la ejecución y devuelve **NaN** (Not a Number).

{ }

```
parseInt('lista21'); //devuelve NaN  
parseInt('*42.3'); //devuelve NaN
```

parseFloat()

Es una **función** nativa de Javascript que recibe como parámetro la cadena de caracteres que queremos parsear y **devuelve un número con decimales** (si el string original los contenía).

```
parseFloat('12.6'); // devuelve 12.6
parseFloat(' 6.7 '); // devuelve 6.7
parseFloat('11.9e'); // devuelve 11.9
parseFloat('-4.32'); // devuelve -4.32
parseFloat('9,57'); // devuelve 9
// porque los decimales en JS se escriben con .
```

¿CÓMO FUNCIONA `parseFloat()`?

La función va a extraer la parte numérica, incluyendo los decimales, del string que recibe hasta llegar a un dato no numérico.

{ }

```
parseFloat('56.7-F-25.8'); // devuelve 56.7  
parseFloat('78.74%'); // devuelve 78.74
```

Si el primer caracter que intenta extraer **no es numérico**, se corta la ejecución y devuelve **NaN** (Not a Number).

{ }

```
parseFloat('resultado: 79.3%'); // devuelve NaN  
parseFloat('*42.3'); // devuelve NaN
```

“

Como Javascript está escrito en inglés estamos obligados a escribir los números también en inglés.

Eso quiere decir que la separación de decimales la escribimos con el punto.



```
var num1 = 10,4; // ❌ error  
var num2 = 10.4; // ✅ sin error
```

String()

Es una **función** nativa de Javascript convierte el valor que le pasemo en un **string**.

```
String(3005); // devuelve '3005'  
String(true); // devuelve 'true'  
String(['Juan', 'María']); // devuelve 'Juan, María'
```

Si bien String() puede trabajar con arrays, **no puede convertir objetos**.

```
String({nombre: 'Rosita'});  
// devuelve "[object Object]"
```

OPERANDO CON String()

{}

```
var numeroA = 5;  
var numeroB = 12;  
numeroA + numeroB; // suma los números y devuelve 17  
String(numeroA) + String(numeroB);  
// convierte a strings, concatena y devuelve '512'
```

{}

```
var llueve = true;  
String(llueve); // devuelve 'true'
```