

Conversiones de Tipos

La mayoría de las veces, los operadores y funciones convierten automáticamente los valores que se les pasan al tipo correcto. Esto es llamado "conversión de tipo".

Por ejemplo, `alert` convierte automáticamente cualquier valor a string para mostrarlo. Las operaciones matemáticas convierten los valores a números.

También hay casos donde necesitamos convertir de manera explícita un valor al tipo esperado.

Aún no hablamos de objetos

En este capítulo no hablamos de objetos. Por ahora, solamente veremos los valores primitivos.

Más adelante, después de haberlos tratado, veremos en el capítulo [Conversión de objeto a valor primitivo](#) cómo funciona la conversión de objetos.

ToString

La conversión a string ocurre cuando necesitamos la representación en forma de texto de un valor.

Por ejemplo, `alert(value)` lo hace para mostrar el valor como texto.

También podemos llamar a la función `String(value)` para convertir un valor a string:

```
let value = true;  
alert(typeof value); // boolean
```

```
value = String(value); // ahora value es el string "true"  
alert(typeof value); // string
```

La conversión a string es bastante obvia. El boolean `false` se convierte en `"false"`, `null` en `"null"`, etc.

ToNumber

La conversión numérica ocurre automáticamente en funciones matemáticas y expresiones.

Por ejemplo, cuando se dividen valores no numéricos usando /:

`alert("6" / "2");` // 3, los strings son convertidos a números

Podemos usar la función `Number(value)` para convertir de forma explícita un valor a un número:

```
let str = "123";  
alert(typeof str); // string  
let num = Number(str); // se convierte en 123  
alert(typeof num); // number
```

La conversión explícita es requerida usualmente cuando leemos un valor desde una fuente basada en texto, como lo son los campos de texto en los formularios, pero que esperamos que contengan un valor numérico.

Si el string no es un número válido, el resultado de la conversión será NaN. Por ejemplo:

```
let age = Number("un texto arbitrario en vez de un número");  
alert(age); // NaN, conversión fallida
```

Reglas de conversión numérica:

Valor	Se convierte en...
undefined	NaN
null	0
true and false	1 y 0
string	Se eliminan los espacios (incluye espacios, tabs \t, saltos de línea \n, etc.) al inicio y final del

Valor

Se convierte en...

texto. Si el string resultante es vacío, el resultado es 0, en caso contrario el número es "leído" del string. Un error devuelve NaN.

Ejemplos:

```
alert( Number(" 123 ") ); // 123
alert( Number("123z") );   // NaN (error al leer un número en "z")
alert( Number(true) );     // 1
alert( Number(false) );   // 0
```

Ten en cuenta que null y undefined se comportan de distinta manera aquí: null se convierte en 0 mientras que undefined se convierte en NaN.

Adición '+' concatena strings

Casi todas las operaciones matemáticas convierten valores a números. Una excepción notable es la suma +. Si uno de los valores sumados es un string, el otro valor es convertido a string.

Luego, los concatena (une):

```
alert( 1 + '2' ); // '12' (string a la derecha)
alert( '1' + 2 ); // '12' (string a la izquierda)
```

Esto ocurre solo si al menos uno de los argumentos es un string, en caso contrario los valores son convertidos a número.

ToBoolean

La conversión a boolean es la más simple.

Ocurre en operaciones lógicas (más adelante veremos test condicionales y otras cosas similares), pero también puede realizarse de forma explícita llamando a la función Boolean(value).

Las reglas de conversión:

- Los valores que son intuitivamente “vacíos”, como 0, "", null, undefined, y NaN, se convierten en false.
- Otros valores se convierten en true.

Por ejemplo:

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

Ten en cuenta: el string con un cero "0" es true

Algunos lenguajes (como PHP) tratan "0" como false. Pero en JavaScript, un string no vacío es siempre true.

```
alert( Boolean("0") ); // true
alert( Boolean(" ") ); // sólo espacios, también true (cualquier string no vacío es true)
```