

Conversión de texto a número y número a texto.

Código para escribir en consola	Resultado al ejecutar	Comentarios
<code>let num = 99; num.toString()</code>	'99'	Una forma de convertir un número a texto
<code>let num = 99; String(num)</code>	'99'	Otra forma de convertir un número en texto es utilizar un typecast que es una conversión explícita
<code>(99.5).toString()</code>	'99.5'	Conversión a texto de un literal numérico
<code>Number.parseInt("99.5")</code>	99	
<code>Number.parseInt("99.5 eur")</code>	99	Solo tiene en cuenta la parte del comienzo de la cadena que se corresponde con un int
<code>Number.parseInt("eur 99.5")</code>	NaN	No hay ningún int al comienzo ...
<code>Number.parseFloat("99.5")</code>	99.5	Convertir a número decimal. Debemos utilizar parseFloat si queremos mantener los decimales.
<code>Number.parseFloat("99.5 euracos")</code>	99.5	Solo tiene en cuenta la parte del comienzo de la cadena que se corresponde con un número decimal
<code>Number.parseFloat("eur 99.5")</code>	NaN	No hay ningún número decimal al comienzo ...
<code>Number("99.5")</code>	99.5	Mediante typecast (forzando <u>explícitamente</u> conversión de tipos)
<code>(4-"2")</code>	2	JS realiza una conversión <u>implícita</u> (convirtiendo el "2" a 2) para intentar realizar la operación
<code>Number("99.5 eur")</code>	NaN	
<code>(255).toString(2)</code>	'1111111'	Convertir el número 255 en base 10 a base 2 o binario (toString admite un parámetro radix que es la base: binario, octal, ...)
<code>(255).toString(16)</code>	'ff'	Convertir el número 255 a hexadecimal
<code>Number.parseInt("11101", 2)</code>	29	Convertir texto a número binario (parseInt también admite un parámetro radix que es la base: binario, octal ...)
<code>Number.parseInt("311101", 2)</code>	NaN	Número imposible en binario
<code>Number.parseInt("11103", 2)</code>	14	
<code>0b111011</code>	59	Número binario
<code>0xFF</code>	256	Número hexadecimal
<code>0o77</code>	63	Número octal