



# Objeto Number

Miguel Ángel López Moyano



# ÍNDICE

1. Utilidad
2. Propiedades
3. Métodos
4. Ejemplos
5. Bibliografía

# UTILIDAD

Permite trabajar con valores numéricos. El objeto Number es un envoltorio para valores primitivos numéricos.

Se crea con el constructor Number: `new Number(valor)`

Siendo valor el valor numérico del objeto que está siendo creado.

# UTILIDAD

Se usa principalmente para:

- Acceder sus propiedades constantes, las cuales representan los números representables más grande y más pequeño, infinitos positivos y negativos y el valor No es Un Número (NaN).
- Crear objetos numéricos que también puedan soportar añadirles propiedades.

# PROPIEDADES

- `Number.MAX_VALUE`: el número más grande que se puede representar.
- `Number.MIN_VALUE`: el número más pequeño que se puede representar.
- `Number.NaN`: valor especial "no es número" NaN.
- `Number.NEGATIVE_INFINITY`: valor especial para representar infinitos negativos. Devuelve un desbordamiento de pila overflow.
- `Number.POSITIVE_INFINITY`: valor especial para representar infinitos positivos. Devuelve un desbordamiento de pila overflow.

# PROPIEDADES

- `Number.prototype`: permite añadir propiedades a un objeto `Number`.
- `Number.EPSILON`: representa la diferencia entre el uno y el menor valor mayor que uno que se puede representar como un número.
- `Number.MAX_SAFE_INTEGER` representa el entero máximo que puede representarse en Javascript con seguridad ( $2^{53} - 1$ ).
- `Number.MIN_SAFE_INTEGER` representa el entero mínimo que puede representarse en Javascript con seguridad  $-(2^{53} - 1)$ .

# MÉTODOS

- `Number.prototype.toExponential`: devuelve una cadena representando el número en notación exponencial.
- `Number.prototype.toFixed`: devuelve una cadena representando el número en una notación de punto fijo.
- `Number.prototype.toLocaleString`: devuelve una cadena legible representando el número utilizando el ámbito local del entorno. Sobrescribe el método `Object.toLocaleString()`.
- `Number.prototype.toPrecision`: devuelve una cadena representando el número en una notación de precisión de punto fijo.

# MÉTODOS

- `Number.prototype.toSource`: devuelve un objeto literal representando el objeto `Number` especificado. Se puede utilizar este valor para crear un nuevo objeto. Sobrescribe el método `Object.toSource()`.
- `Number.prototype.toString`: devuelve una cadena representando el objeto especificado. Sobrescribe el método `Object.toString()`.
- `Number.prototype.valueOf`: devuelve el valor primitivo de un objeto especificado. Sobrescribe el método `Object.valueOf()`.
- `Number.isFinite`: determina si el valor pasado es un número finito.



# MÉTODOS

- `Number.isInteger`: determina si el valor pasado es de tipo entero.
- `Number.isNaN`: determina si el valor pasado es NaN.
- `Number.isSafeInteger`: determina si el valor proporcionado es un número entero seguro.
- `Number.parseFloat`: analiza un argumento de cadena y devuelve un número de coma flotante. Este método comporta de forma idéntica a la función global `parseFloat()` y es parte de ECMAScript 6 (su finalidad es la modularización de globales).
- `Number.parseInt`: analiza un argumento de cadena y devuelve un número entero de la raíz o base especificada.

# EJEMPLOS

```
var x=Number("tres");  
document.write(x + "<BR>");
```

```
//Esto imprimirá NaN, ya que "tres" no es un número.
```

# EJEMPLOS

```
var masGrandeNum = Number.MAX_VALUE;  
var maspequeNum = Number.MIN_VALUE;  
var infinitoNum = Number.POSITIVE_INFINITY;  
var notInfinitoNum = Number.NEGATIVE_INFINITY;  
var noesNum = Number.NaN;
```

# EJEMPLOS

```
var valor = Number(true) // valor = 1  
var valor = Number(false) // valor = 0
```

# EJEMPLOS

```
var numero = 1234.56789  
var num1 = numero.toFixed(2) // num1 = 1234.57  
var num2 = numero.toFixed(4) // num2 = 1234.5679  
var num3 = numero.toFixed() // num3 = 1235
```

# EJEMPLOS

```
var numeroEnBase10 = parseInt(numero,base)
```

```
var valor = parseInt("7b",16);
```

La variable valor será igual a 123, ya que el número 7b en sistema hexadecimal es igual a 123 en sistema decimal.

# BIBLIOGRAFÍA

- [https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos\\_globales/Number](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/Number)
- [http://aprende-web.net/javascript/js9\\_1.php](http://aprende-web.net/javascript/js9_1.php)

**¿DUDAS?**

