



Funciones Math en JavaScript explicadas

Math


`Math` es uno de los objetos globales u objeto incorporado estándar de JavaScript y puede ser utilizado en cualquier lugar donde puedas usar JavaScript. Contiene constantes útiles como π y el número de Euler, además de funciones como `floor()`, `round()` y `ceil()`.

En este artículo, veremos ejemplos de muchas de esas funciones. Pero primero, aprendamos más sobre el objeto `Math`.

Ejemplo

El siguiente ejemplo muestra como usar el objeto `Math` para escribir una función que calcula el área de un círculo:

```
function calcularAreaDeCirculo(radio) {  
  return Math.PI * Math.pow(radio, 2);  
}  
  
calcularAreaDeCirculo(1); // 3.141592653589793
```



`Math.max()` es una función que devuelve el valor mayor de una lista de valores numéricos pasados como parámetros. Si se pasa un valor no numérico como parámetro, `Math.max()` devolverá `NaN`.

Un arreglo de valores numéricos puede ser pasado como un solo parámetro a `Math.max()` usando `spread (...)` o `apply`. Cualquiera de estos métodos puede, sin embargo, fallar cuando la cantidad de números en el arreglo es muy grande.

Sintaxis

```
Math.max(valor1, valor2, valor3, ...);
```

Parámetros

Números, o una cantidad limitada de números en un arreglo.

Valor de retorno

El mayor valor numérico entre los números dados, o `NaN` si alguno de los valores no es numérico.

Ejemplos

Números como parámetros

```
Math.max(4, 13, 27, 0, -5); // devuelve 27
```

Parámetro inválido



Arreglo como parámetro, usando Spread (...)

```
let numeros = [4, 13, 27, 0, -5];  
  
Math.max(...numeros); // devuelve 27
```

Arreglo como parámetro, usando .apply()

```
let numeros = [4, 13, 27, 0, -5];  
  
Math.max.apply(null, numeros); // devuelve 27
```

Math Min

La función `Math.min()` devuelve el menor valor de cero o más números.

Le puedes pasar cualquier cantidad de argumentos.

```
Math.min(7, 2, 9, -6);  
// devuelve -6
```

Math PI

aproximadamente 3.14149 y se representa comúnmente con la letra griega π .

Ejemplo

```
Math.PI \\ 3.141592653589793
```

Más información:

[MDN](#)

Math Pow

`Math.pow()` devuelve el valor de un número elevado a otro número.

Sintaxis

`Math.pow(base, exponente)` , donde `base` es el número base y `exponente` es el número al que se eleva la `base` .

`pow()` es un método estático de `Math` , por lo tanto, siempre es llamado como `Math.pow()` en vez de un método en otro objeto.

Ejemplos

```
Math.pow(5, 2); // 25
Math.pow(7, 4); // 2401
Math.pow(9, 0.5); // 3
Math.pow(-8, 2); // 64
Math.pow(-4, 3); // -64
```



La función `Math.sqrt()` devuelve la raíz cuadrada de un número.

Si se ingresa un número negativo, te devuelve `NaN`.

`sqrt()` es un método estático de `Math`, por lo tanto, siempre es llamado como `Math.sqrt()` en vez de un método en otro objeto.

Sintaxis

`Math.sqrt(x)`, donde `x` es un número.

Ejemplos

```
Math.sqrt(25); // 5
Math.sqrt(169); // 13
Math.sqrt(3); // 1.732050807568
Math.sqrt(1); // 1
Math.sqrt(-5); // NaN
```

Math Trunc

`Math.trunc()` es un método del objeto estándar `Math` que devuelve sólo la parte entera de un número dado, simplemente quitando las unidades fraccionarias. Esto resulta en hacer un redondeo de la parte decimal a 0. Cualquier valor ingresado que no sea un número devolverá como resultado `NaN`.

Cuidado: este método es una característica de ECMAScript 2015 (ES6) y no tiene soporte en los navegadores antiguos.

```
Math.trunc(-0.9) // -0
Math.trunc(-1.5) // -1
Math.trunc('foo') // NaN
```

Math Ceil

`Math.ceil()` es un método del objeto estándar `Math` que redondea un número dado hacia el siguiente número entero. Toma nota que para los números negativos, esto significa que el número será redondeado "hacia 0" en vez de el número de mayor valor absoluto (ve los ejemplos a continuación).

Ejemplos

```
Math.ceil(0.1) // 1
Math.ceil(1.3) // 2
Math.ceil(-0.9) // -0
Math.ceil(-1.5) // -1
```

Math Floor

`Math.floor()` es un método del objeto estándar `Math` que redondea un número dado hacia el número entero anterior. Toma nota que para los números negativos esto significa que el número será redondeado "lejos de 0" en vez de el número de menor valor absoluto debido a que `Math.floor()` devuelve el número entero que sea menor o igual al número dado.

Ejemplos

```
Math.floor(-0.9) // -1
Math.floor(-1.3) // -2
```

Aplicando math.floor: Cómo crear una Máquina Tragamonedas de JavaScript

Para este ejercicio, tenemos que generar tres números al azar usando una fórmula específica y no la fórmula general. `Math.floor`

```
(Math.random() * (3 - 1 + 1)) + 1;
```

```
puestoUno = Math.floor(Math.random() * (3 - 1 + 1)) + 1;
puestoDos = Math.floor(Math.random() * (3 - 1 + 1)) + 1;
puestoTres = Math.floor(Math.random() * (3 - 1 + 1)) + 1;
```

Otro ejemplo: Encontrando el residuo

```
5 % 2 = 1 porque
Math.floor(5 / 2) = 2 (Cociente)
2 * 2 = 4
5 - 4 = 1 (Residuo)
```

Uso

En matemáticas, se puede verificar si un número es par o impar viendo el residuo de dividir dicho número entre 2.

```
17 % 2 = 1 (17 es impar)
48 % 2 = 0 (48 es par)
```

Nota No confundas que el operador *Resto* % no funciona bien con números negativos.

