# Date

Permite trabajar con fechas y horas.

## El constructor

El constructor Date:

new Date()

new Date(milisegundos)

new Date(cadenaFecha)

new Date(año\_num, mes\_num, dia\_num [, hor\_num, min\_num, seg\_num, mils\_num])

### Parámetros

**Milisegundos (**Valor entero que representa el número de milisegundos desde las 00:00:00 UTC del 1 de enero de 1970.)

**cadenaFecha (**Valor de tipo cadena que representa una fecha. La cadena debería estar en un formato reconocido por el método [Date.parse()](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/Date/parse" \o "Transforma una cadena con la representación de una fecha y hora, y devuelve el número de milisegundos desde las 00:00:00 del 1 de enero de 1970, hora local.).)

**año\_num, mes\_num, dia\_num (**Valores enteros con las representaciones de las partes de una fecha. Como valor entero, el mes se representa de 0 a 11, con 0=enero and 11=diciembre.)

**hor\_num, min\_num, seg\_num, mils\_num (**Valores enteros que representan las partes de una hora completa.)

## Descripción

Si no proporciona argumentos, el constructor crea un objeto **Date** con la hora y fecha de hoy según la hora local.

Si proporciona algunos argumentos, debe proporcionar al menos 2 argumentos. Los argumentos vacíos se establecen a 0 (ó 1 si falta el día).

La fecha se mide en milisegundos desde la media noche exacta del 01 de enero de 1970 en formato UTC. Un día contiene 86.400.000 milisegundos. El rango del objeto Date va desde -100,000,000 días hasta 100,000,000 días respecto del 01 de enero de 1970 UTC.

El objeto Date proporciona un comportamiento uniforme entre plataformas.

El objeto Date soporta métodos UTC (universales), además de métodos horarios locales. UTC, también conocido como Greenwich Mean Time (GMT), se refiere a la hora según el Estándar Horario Mundial (World Time Standard). La hora local es la hora establecida por el ordenador donde se ejecuta JavaScript.

Por compatibilidad con los cálculos del nuevo milenio (en otras palabras, para tener en cuenta el efecto 2000), debería especificar siempre el año completo; por ejemplo, utilice 1998, y no 98.

Para ayudarle a especificar el año completo, JavaScript incluye los métodos:

* [Date.prototype.getFullYear()](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/Date/getFullYear" \o "The documentation about this has not yet been written; please consider contributing!),
* [Date.prototype.setFullYear()](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/Date/setFullYear" \o "The documentation about this has not yet been written; please consider contributing!),
* [Date.prototype.getUTCFullYear()](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/Date/getUTCFullYear" \o "The documentation about this has not yet been written; please consider contributing!), y
* [Date.prototype.setUTCFullYear()](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/Date/setUTCFullYear).

El siguiente ejemplo devuelve el tiempo transcurrido entre horaA y horaB en milisegundos.

horaA = new Date();

// Sentencias que realizan alguna acción.

horaB = new Date();

diferenciaHoras = horaB - horaA;

## Propiedades

For proprietades available on*Date*instances, see*[Proprietades of Date instances](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/Date/prototype" \l "Proprietades)*.

[**Date.prototype**](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/Date/prototype)

Permite añadir propiedades a un objeto Date.

**Properties inherited from**[**Function**](https://developer.mozilla.org/es/docs/Web/JavaScript/Reference/Global_Objects/Function)**:**

[arity](https://developer.mozilla.org/es/docs/Web/JavaScript/Reference/Global_Objects/Function/Arity), [caller](https://developer.mozilla.org/es/docs/Web/JavaScript/Reference/Global_Objects/Function/Caller), [constructor](https://developer.mozilla.org/es/docs/Web/JavaScript/Reference/Global_Objects/Function/Constructor), [length](https://developer.mozilla.org/es/docs/Web/JavaScript/Reference/Global_Objects/Function/Length), [name](https://developer.mozilla.org/es/docs/Web/JavaScript/Reference/Global_Objects/Function/Name)

## Métodos

For m.c3.a9todos available on*Date*instances, see[*M.c3.a9todos of Date instances*](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/Date/prototype#M.C3.A9todos).

[**Date.now()**](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/Date/now)

Devuelve el valor numérico correspondiente a la hora actual.

[**Date.parse()**](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/Date/parse)

Transforma una cadena que representa una fecha, y devuelve el número de milisegundos desde el 1 de enero de 1970, hora local 00:00:00.

[**Date.UTC()**](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/Date/UTC)

Acepta los mismos parámetros que la forma más larga del constructor, y devuelve el número de milisegundos en un objeto Date desde el 1 de enero de 1970, hora universal 00:00:00.

## Instancias de  Date

### Métodos

* [**Date.prototype.getDate()**](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/Date/getDate)
  + Devuelve el día del mes de la fecha especificada según la hora local.
* [**Date.prototype.getDay()**](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/Date/getDay)
  + Devuelve el día de la semana de la fecha especificada según la hora local.
* [**Date.prototype.getFullYear()**](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/Date/getFullYear)
  + Devuelve el año de la fecha especificada según la hora local.
* [**Date.prototype.getHours()**](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/Date/getHours)
  + Devuelve la hora de la fecha especificada según la hora local.
* [**Date.prototype.getMilliseconds()**](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/Date/getMilliseconds)
  + Devuelve los milisegundos de la fecha especificada según la hora local.
* [**Date.prototype.getMinutes()**](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/Date/getMinutes)
  + Devuelve los minutos de la fecha especificada según la hora local.
* [**Date.prototype.getMonth()**](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/Date/getMonth)
  + Devuelve el mes de la fecha especificada según la hora local.
* [**Date.prototype.getSeconds()**](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/Date/getSeconds)
  + Devuelve los segundos de la fecha especificada según la hora local.
* [**Date.prototype.getTime()**](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/Date/getTime)
  + Devuelve el valor numérico correspondiente a la hora especificada según la hora universal.
* [**Date.prototype.getTimezoneOffset()**](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/Date/getTimezoneOffset)
  + Devuelve la diferencia horaria en minutos para la zona geográfica actual.
* [**Date.prototype.getUTCDate()**](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/Date/getUTCDate)
  + Devuelve el día del mes de la fecha especificada según la hora universal.
* [**Date.prototype.getUTCDay()**](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/Date/getUTCDay)
  + Devuelve el día de la semana de la fecha especificada según la hora universal.
* [**Date.prototype.getUTCFullYear()**](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/Date/getUTCFullYear)
  + Devuelve el día el año de la fecha especificada según la hora universal.
* [**Date.prototype.getUTCHours()**](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/Date/getUTCHours)
  + Devuelve las horas de la fecha especificada según la hora universal.
* [**Date.prototype.getUTCMilliseconds()**](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/Date/getUTCMilliseconds)
  + Devuelve los milisegundos de la fecha especificada según la hora universal.
* [**Date.prototype.getUTCMinutes()**](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/Date/getUTCMinutes)
  + Devuelve los minutos de la fecha especificada según la hora universal.
* [**Date.prototype.getUTCMonth()**](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/Date/getUTCMonth)
  + Devuelve el mes de la fecha especificada según la hora universal.
* [**Date.prototype.getUTCSeconds()**](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/Date/getUTCSeconds)
  + Devuelve los segundos de la fecha especificada según la hora universal.
* [**Date.prototype.getYear()**](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/Date/getYear)
  + Devuelve el año de la fecha especificada según la hora local. Use [Date.prototype.getFullYear()](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/Date/getFullYear" \o "The documentation about this has not yet been written; please consider contributing!) a cambio.
* [**Date.prototype.setDate()**](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/Date/setDate)
  + Establece el día del mes de la fecha especificada según la hora local.
* [**Date.prototype.setFullYear()**](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/Date/setFullYear)
  + Establece el año completo de la fecha especificada según la hora local.
* [**Date.prototype.setHours()**](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/Date/setHours)
  + Establece las horas de la fecha especificada según la hora local.
* [**Date.prototype.setMilliseconds()**](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/Date/setMilliseconds)
  + Establece los milisegundos de la fecha especificada según la hora local.
* [**Date.prototype.setMinutes()**](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/Date/setMinutes)
  + Establece los minutos de la fecha especificada según la hora local.
* [**Date.prototype.setMonth()**](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/Date/setMonth)
  + Establece el mes de la fecha especificada según la hora local.
* [**Date.prototype.setSeconds()**](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/Date/setSeconds)
  + Establece los segundos de la fecha especificada según la hora local.
* [**Date.prototype.setTime()**](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/Date/setTime)
  + Establece el valor del objeto Date según la hora local.
* [**Date.prototype.setUTCDate()**](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/Date/setUTCDate)
  + Establece el día del mes de la fecha especificada según la hora universal.
* [**Date.prototype.setUTCFullYear()**](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/Date/setUTCFullYear)
  + Establece el año completo de la fecha especificada según la hora universal.
* [**Date.prototype.setUTCHours()**](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/Date/setUTCHours)
  + Establece la hora de la fecha especificada según la hora universal.
* [**Date.prototype.setUTCMilliseconds()**](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/Date/setUTCMilliseconds)
  + Establece los milisegundos de la fecha especificada según la hora universal.
* [**Date.prototype.setUTCMinutes()**](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/Date/setUTCMinutes)
  + Establece los minutos de la fecha especificada según la hora universal.
* [**Date.prototype.setUTCMonth()**](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/Date/setUTCMonth)
  + Establece el mes de la fecha especificada según la hora universal.
* [**Date.prototype.setUTCSeconds()**](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/Date/setUTCSeconds)
  + Establece los segundos de la fecha especificada según la hora universal.
* [**Date.prototype.setYear ()**](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/Date/setYear)
  + Establece el año de la fecha especificada según la hora local. Use [Date.prototype.setFullYear()](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/Date/setFullYear" \o "The documentation about this has not yet been written; please consider contributing!) a cambio.
* [**Date.prototype.toGMTString()**](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/Date/toGMTString)
  + Convierte una fecha en una cadena, usando las convenciones de Internet GMT. Utilice [Date.prototype.toUTCString()](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/Date/toUTCString) a cambio.
* [**Date.prototype.toLocaleString()**](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/Date/toLocaleString)
  + Convierte una fecha en una cadena, usando las reglas de la localización actual. Sobreescribe el método [Object.toLocaleString()](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/Object/toLocaleString" \o "The documentation about this has not yet been written; please consider contributing!).
* [**Date.prototype.toLocaleDateString()**](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/Date/toLocaleDateString)
  + Devuelve la porción fecha (sin la hora) de una fecha como una cadena, usando las reglas de la localización actual.
* [**Date.prototype.toLocaleTimeString()**](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/Date/toLocaleTimeString)
  + Devuelve la porción hora (sin la fecha) de una fecha como una cadena, siguiendo las reglas de la localización actual.
* [**Date.prototype.toSource()**](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/Date/toSource)
  + Devuelve un literal que representa al objeto Date especificado; puede usar este valor para crear un nuevo objeto. Sobreescribe el método [Object.toSource()](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/Object/toSource" \o "The documentation about this has not yet been written; please consider contributing!).
* [**Date.prototype.toString()**](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/Date/toString)
  + Devuelve una cadena representando el objeto Date especificado. Sobreescribe el método [Object.toString()](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/Object/toString).
* [**Date.prototype.toUTCString()**](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/Date/toUTCString)
  + Convierte una fecha en una cadena, usando las reglas horarias universales.
* [**Date.prototype.valueOf()**](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/Date/valueOf)
  + Devuelve el valor primitivo de un objeto Date. Sobreescribe el método [Object.valueOf()](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/Object/valueOf" \o "The documentation about this has not yet been written; please consider contributing!).

### Ejemplo:

### Calcular el tiempo transcurrido

Los siguientes ejemplos muestran como determinar el tiempo transcurrido entre dos fechas:

// usando métodos estáticos

var inicio = Date.now();

// el evento cuyo tiempo ha transcurrido aquí:

hacerAlgoPorAlgunTiempo();

var fin = Date.now();

var transcurso = fin - inicio; // tiempo en milisegundos

// si tiene objetos Date

var inicio = new Date();

// el evento cuyo tiempo ha transcurrido aquí:

hacerAlgoPorAlgunTiempo();

var fin = new Date();

var transcurso = fin.getTime() - inicio.getTime(); // tiempo en milisegundos

# Number

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

## Creado por

El constructor Number:

new Number(valor)

### Parámetros

**Valor** (El valor numérico de un objeto que está siendo creado.)

## Descripción

La utilización primaria del objeto Number es:

* 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](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/NaN" \o "Un valor representando un Not-A-Number (No es Un Número).)).
* Crear objetos numéricos que también puedan soportar añadirles propiedades. Más claro, muy rara vez necesitará crea un objeto Number.
* Las propiedades de Number son propiedades de la clase en sí, no de objetos individuales Number.

**JavaScript 1.2**: Number(x) actualmente produce NaN además de un error si x es una cadena que no contiene un literal numérico correctamente formateado. Por ejemplo, el siguiente código despliega NaN:

x=Number("tres");

document.write(x + "<BR>");

Puede convertir cualquier objeto a un number utilizando funciones numéricas de alto nivel del objetoNumber.

## Propiedades

[Number.MAX\_VALUE](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/Number/MAX_VALUE): El número más grande representable.

[Number.MIN\_VALUE](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/Number/MIN_VALUE): El número más pequeño representable.

[Number.NaN](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/Number/NaN): Valor especial "no es número" NaN.

[Number.NEGATIVE\_INFINITY](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/Number/NEGATIVE_INFINITY): Valor especial para representar infinitos negativos; retorno de un desborde de pila overflow.

[Number.POSITIVE\_INFINITY](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/Number/POSITIVE_INFINITY): Valor especial para representar infinitos positivos; retorno de un desborde de pila overflow.

[Number.prototype](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/Number/prototype): Permite la adición de propiedades a un objeto Number.

## Métodos

[Number.prototype.toExponential](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/Number/toExponential): Retorna una cadena representando el número en notación exponencial.

[Number.prototype.toFixed](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/Number/toFixed): Retorna una cadena representando el número en una notación de punto fijo.

[Number.prototype.toLocaleString](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/Number/toLocaleString): Retorna una cadena humanamente legible representando el número utilizando el ámbito local del entorno. Sobre escribe el método [Object.toLocaleString()](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/Object/toLocaleString" \o "The documentation about this has not yet been written; please consider contributing!).

[Number.prototype.toPrecision](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/Number/toPrecision): Retorna una cadena representando el número en una notación de precisión de punto fijo.

[Number.prototype.toSource](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/Number/toSource): Retorna un objeto literal representando el objeto Number especificado; puede utilizar este valor para crear un nuevo objeto. sobre escribe el método [Object.toSource()](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/Object/toSource" \o "The documentation about this has not yet been written; please consider contributing!).

[Number.prototype.toString](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/Number/toString): Retorna una cadena representando el objeto especificado. Sobre escribe el método [Object.toString()](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/Object/toString" \o "Devuelve una cadena que representa al objeto.).

[Number.prototype.valueOf](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/Number/valueOf): Retorna el valor primitivo de un objeto especificado. Sobre escribe el método [Object.valueOf()](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/Object/valueOf" \o "El método valueOf() retorna el valor primitivo del objeto especificado.).

Adicionalmente, este objeto hereda los métodos [watch()](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/Object/watch" \o "Controla si se asigna un valor a una propiedad y ejecuta una función cuando esto ocurre.) y [unwatch()](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/Object/unwatch" \o "Elimina un control establecido con el método watch() sobre una propiedad de un objeto.) de [Object](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/Object" \o "Crea un contenedor de objeto).

## Ejemplos

### Ejemplo: Utilizando el objeto Number para asignar valores a variables numéricas

El siguiente ejemplo utiliza las propiedades del objeto Number para asignar valores a varias variables numéricas:

MásgrandeNum = Number.MAX\_VALUE;

MáspequeNum = Number.MIN\_VALUE;

infinitoNum = Number.POSITIVE\_INFINITY;

notInfinitoNum = Number.NEGATIVE\_INFINITY;

noesNum = Number.NaN;

### Ejemplo:

### Utilizando el objeto Number para modificar todos los objetos Number

El siguiente ejemplo crea un objeto Number, miNúmero, luego añade la propiedad descripción a todos los objetos Number. Luego el valor es asignado a la propiedad descripción del objeto miNúmero.

miNúmero = new Number(65);

Number.prototype.descripción = null;

miNúmero.descripción = "velocidad del viento";

**Math**

**Math** es un objeto incorporado por javascript que posee propiedades y métodos creados por constantes y funciones matemáticas. No es un objeto de función.

## Descripción

A diferencia de los demás objetos globales, el objeto **Math** no se puede editar. Todas las propiedades y métodos de **Math son** estáticos. Usted se puede referir a la constante pi como ***Math.PI*** y puede llamar a la función seno como ***Math.sin(x),*** donde ***x*** es el argumento del método. Las constantes se definen con la precisión completa de los números reales en JavaScript.

## Propiedades

[Math.E](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/Math/E)

Constante de Euler, la base de los logaritmos naturales, aproximadamente 2.718

[Math.LN2](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/Math/LN2)

Logaritmo natural de 2, aproximadamente 0.693

[Math.LN10](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/Math/LN10)

Logaritmo natural de 10, aproximadamente 2.303

[Math.LOG2E](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/Math/LOG2E)

Logaritmo de E con base 2, aproximadamente 1.443

[Math.LOG10E](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/Math/LOG10E)

Logaritmo de E con base 10, aproximadamente 0.434

[Math.PI](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/Math/PI)

Ratio of the circumference of a circle to its diameter, approximately 3.14159.

[Math.SQRT1\_2](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/Math/SQRT1_2)

Raíz cuadrada de 1/2; Equivalentemente, 1 sobre la raíz cuadrada de 2, aproximadamente 0.707.

[Math.SQRT2](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/Math/SQRT2)

Raíz cuadrada de 2, aproximadamente 1.414.

## Métodos

Tenga en cuenta que las funciones trigonométricas (sin(), cos(), tan(), asin(), acos(), atan(), atan2()) devuelven ángulos en radianes. Para convertir radianes a grados, divida por (Math.PI / 180), y multiplique por esto para convertir a la inversa.

Tenga en cuenta que muchas de las funciones matemáticas tienen una precisión que es dependiente de la implementación. Esto significa que los diferentes navegadores pueden dar un resultado diferente, e incluso el mismo motor de JS en un sistema operativo o arquitectura diferente puede dar resultados diferentes.

[Math.abs(x)](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Global_Objects/Math/abs)

Devuelve el valor absoluto de un número.

[Math.acos(x)](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Global_Objects/Math/acos)

Devuelve el arco coseno de un número.

[Math.acosh(x)](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Global_Objects/Math/acosh)

Devuelve el arco coseno hiperbólico de un número.

[Math.asin(x)](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Global_Objects/Math/asin)

Devuelve el arco seno de un número.

[Math.asinh(x)](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Global_Objects/Math/asinh)

Devuelve el arco seno hiperbólico de un número.

[Math.atan(x)](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Global_Objects/Math/atan)

Devuelve el arco tangente de un número.

[Math.atanh(x)](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Global_Objects/Math/atanh)

Devuelve el arco tangente hiperbólico de un número.

[Math.atan2(y, x)](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Global_Objects/Math/atan2)

Devuelve el arco tangente del cociente de sus argumentos.

[Math.cbrt(x)](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Global_Objects/Math/cbrt)

Devuelve la raíz cúbica de un número.

[Math.ceil(x)](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Global_Objects/Math/ceil)

Devuelve el entero más pequeño mayor o igual que un número.

[Math.clz32(x)](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Global_Objects/Math/clz32)

Devuelve el número de ceros iniciales de un entero de 32 bits.

[Math.cos(x)](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Global_Objects/Math/cos)

Devuelve el coseno de un número.

[Math.cosh(x)](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Global_Objects/Math/cosh)

Devuelve el coseno hiperbólico de un número.

[Math.exp(x)](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Global_Objects/Math/exp)

Devuelve ex, donde x es el argumento, y e es la constante de Euler (2.718...), la base de los logaritmos naturales.

[Math.expm1(x)](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Global_Objects/Math/expm1)

Las devoluciones restando 1 exp (x)

[Math.floor(x)](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Global_Objects/Math/floor)

Devuelve el mayor entero menor que o igual a un número.

[Math.fround(x)](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Global_Objects/Math/fround)

Devuelve la representación flotante de precisión simple más cercana de un número.

[Math.hypot([x[, y[, …]]])](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Global_Objects/Math/hypot)

Devuelve la raíz cuadrada de la suma de los cuadrados de sus argumentos.

[Math.imul(x, y)](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Global_Objects/Math/imul)

Devuelve el resultado de una multiplicación de enteros de 32 bits.

[Math.log(x)](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Global_Objects/Math/log)

Devuelve el logaritmo natural (log, también ln) de un número.

[Math.log1p(x)](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Global_Objects/Math/log1p)

Devuelve el logaritmo natural de x + 1 (loge, también ln) de un número.

[Math.log10(x)](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Global_Objects/Math/log10)

Devuelve el logaritmo en base 10 de x.

[Math.log2(x)](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Global_Objects/Math/log2)

Devuelve el logaritmo en base 2 de x.

[Math.max([x[, y[, …]]])](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Global_Objects/Math/max)

Devuelve el mayor de cero o más números.

[Math.min([x[, y[, …]]])](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Global_Objects/Math/min)

Devuelve el más pequeño de cero o más números.

[Math.pow(x, y)](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Global_Objects/Math/pow)

Las devoluciones de base a la potencia de exponente, que es, baseexponent.

[Math.random()](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Global_Objects/Math/random)

Devuelve un número pseudo-aleatorio entre 0 y 1.

[Math.round(x)](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Global_Objects/Math/round)

Devuelve el valor de un número redondeado al número entero más cercano.

[Math.sign(x)](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Global_Objects/Math/sign)

Devuelve el signo de la x, que indica si x es positivo, negativo o cero.

[Math.sin(x)](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Global_Objects/Math/sin)

Devuelve el seno de un número.

[Math.sinh(x)](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Global_Objects/Math/sinh)

Devuelve el seno hiperbólico de un número.

[Math.sqrt(x)](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Global_Objects/Math/sqrt)

Devuelve la raíz cuadrada positiva de un número.

[Math.tan(x)](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Global_Objects/Math/tan)

Devuelve la tangente de un número.

[Math.tanh(x)](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Global_Objects/Math/tanh)

Devuelve la tangente hiperbólica de un número.

[Math.trunc(x)](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Global_Objects/Math/trunc)

Devuelve la parte entera del número x, la eliminación de los dígitos fraccionarios.