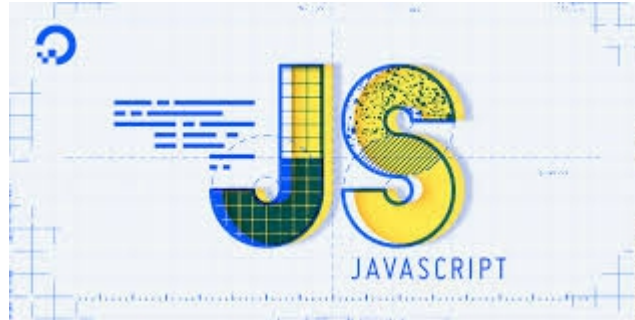


PROYECTO DE LENGUAJE DE MARCAS  
3-TRIMESTRE

Date and Math object methods:



JOSE MARIA BELLIDO VERA  
1-ASIR  
IES TRIANA  
CURSO 2018/2019

## nueva fecha ()

`new Date()` crea un nuevo objeto de fecha con la fecha y hora actual :

```
var d = new Date();
```

y le das salida por un texto te devuelve la fecha actual y la hora actual.

## nueva fecha (año, mes, ...)

`new Date(year, month, ...)` crea un nuevo objeto de fecha con una fecha y hora especificadas.

Los 7 números especifican año, mes, día, hora, minuto, segundo y milisegundos (en ese orden):

```
var d = new Date(2018, 11, 24, 10, 33, 30, 0);
```

## nueva fecha ( dateString )

`new Date(dateString)` crea un nuevo objeto de fecha a partir de una cadena de fecha :

```
var d = new Date("October 13, 2014 11:13:00");
```

## nueva fecha ( milisegundos )

`new Date(milliseconds)` crea un nuevo objeto de fecha como tiempo cero más milisegundos :

```
var d = new Date(0);
```

El 1 de enero de 1970 más 100 000 000 000 milisegundos es aproximadamente el 3 de marzo de 1973:

```
var d = new Date(100000000000);
```

El 1 de enero de 1970 menos 100 000 000 000 milisegundos es aproximadamente el 31 de octubre de 1966:

```
var d = new Date(-1000000000000);
```

## Fechas ISO de JavaScript

ISO 8601 es el estándar internacional para la representación de fechas y horas.

La sintaxis ISO 8601 (YYYY-MM-DD) también es el formato de fecha de JavaScript preferido:

## Ejemplo (fecha completa)

```
var d = new Date("2015-03-25");
```

La fecha calculada será relativa a su zona horaria.

Dependiendo de su zona horaria, el resultado anterior variará entre el 24 de marzo y el 25 de marzo.

---

## Fechas ISO (Año y Mes)

Las fechas ISO se pueden escribir sin especificar el día (YYYY-MM):

```
var d = new Date("2015-03");
```

Las zonas horarias variarán el resultado anterior entre el 28 de febrero y el 1 de marzo.

---

## Fechas ISO (solo año)

Las fechas ISO se pueden escribir sin mes y día (YYYY):

```
var d = new Date("2015");
```

Las zonas horarias variarán el resultado anterior entre el 31 de diciembre de 2014 y el 1 de enero de 2015.

---

# Fechas ISO (Fecha-Hora)

Las fechas ISO se pueden escribir con horas, minutos y segundos agregados (YYYY-MM-DDTHH: MM: SSZ):

```
var d = new Date("2015-03-25T12:00:00Z");
```

La fecha y la hora se separan con una T mayúscula.

La hora UTC se define con una letra mayúscula Z.

Si desea modificar el tiempo relativo a UTC, elimine la Z y agregue + HH: MM o -HH: MM en su lugar:

```
var d = new Date("2015-03-25T12:00:00-06:30");
```

UTC (Tiempo Universal Coordinado) es lo mismo que GMT (Tiempo Medio de Greenwich)

# JavaScript Math Object:

## Math.round ()

Math.round(x) devuelve el valor de x redondeado a su entero más cercano:

```
Math.round(4.7);    // returns 5  
Math.round(4.4);    // returns 4
```

---

## Math.pow ()

Math.pow(x, y) devuelve el valor de x al poder de y:

```
Math.pow(8, 2);      // returns 64
```

---

## Math.sqrt ()

`Math.sqrt(x)` devuelve la raíz cuadrada de x:

```
Math.sqrt(64);    // returns 8
```

---

---

## Math.abs ()

`Math.abs(x)` devuelve el valor absoluto (positivo) de x:

```
Math.abs(-4.7);   // returns 4.7
```

---

---

## Math.ceil ()

`Math.ceil(x)` devuelve el valor de x redondeado hasta a su número entero más próximo:

```
Math.ceil(4.4);   // returns 5
```

---

---

## Math.floor ()

`Math.floor(x)` devuelve el valor de x redondeado hacia abajo a su número entero más próximo:

```
Math.floor(4.7);  // returns 4
```

---

---

## Math.sin ()

`Math.sin(x)` devuelve el seno (un valor entre -1 y 1) del ángulo `x` (dado en radianes).

Si desea utilizar grados en lugar de radianes, debe convertir los grados en radianes:

Ángulo en radianes = ángulo en grados x  $\text{PI} / 180$ .

```
Math.sin(90 * Math.PI / 180);    // returns 1 (the sine of 90 degrees)
```

---

## Math.cos ()

`Math.cos(x)` devuelve el coseno (un valor entre -1 y 1) del ángulo `x` (dado en radianes).

Si desea utilizar grados en lugar de radianes, debe convertir los grados en radianes:

Ángulo en radianes = ángulo en grados x  $\text{PI} / 180$ .

```
Math.cos(0 * Math.PI / 180);    // returns 1 (the cos of 0 degrees)
```

---

## Math.min () y Math.max ()

`Math.min()` y `Math.max()` se puede usar para encontrar el valor más bajo o más alto en una lista de argumentos:

```
Math.min(0, 150, 30, 20, -8, -200); // returns -200
```

```
Math.max(0, 150, 30, 20, -8, -200); // returns 150
```

---

# Math.random ()

`Math.random()` devuelve un número aleatorio entre 0 (incluido) y 1 (exclusivo):

```
Math.random();    // returns a random number
```

Aprenderás más sobre `Math.random()` el próximo capítulo de este tutorial.

---

## Propiedades matemáticas (constantes)

JavaScript proporciona 8 constantes matemáticas a las que se puede acceder con el objeto `Math`:

```
Math.E          // returns Euler's number
Math.PI         // returns PI
Math.SQRT2      // returns the square root of 2
Math.SQRT1_2    // returns the square root of 1/2
Math.LN2        // returns the natural logarithm of 2
Math.LN10       // returns the natural logarithm of 10
Math.LOG2E      // returns base 2 logarithm of E
Math.LOG10E     // returns base 10 logarithm of E
```

---

# Constructor de matematicas

A diferencia de otros objetos globales, el objeto Math no tiene constructor. Los métodos y propiedades son estáticos.

Todos los métodos y propiedades (constantes) pueden usarse sin crear primero un objeto matemático.

---

## Métodos de objeto matemático

Method	Description
<code>abs(x)</code>	Returns the absolute value of x
<code>acos(x)</code>	Returns the arccosine of x, in radians
<code>asin(x)</code>	Returns the arcsine of x, in radians
<code>atan(x)</code>	Returns the arctangent of x as a numeric value between $-\pi/2$ and $\pi/2$ radians
<code>atan2(y, x)</code>	Returns the arctangent of the quotient of its arguments
<code>ceil(x)</code>	Returns the value of x rounded up to its nearest integer
<code>cos(x)</code>	Returns the cosine of x (x is in radians)
<code>exp(x)</code>	Returns the value of $E^x$
<code>floor(x)</code>	Returns the value of x rounded down to its nearest integer
<code>log(x)</code>	Returns the natural logarithm (base E) of x
<code>max(x, y, z, ..., n)</code>	Returns the number with the highest value
<code>min(x, y, z, ..., n)</code>	Returns the number with the lowest value
<code>pow(x, y)</code>	Returns the value of x to the power of y
<code>random()</code>	Returns a random number between 0 and 1
<code>round(x)</code>	Returns the value of x rounded to its nearest integer
<code>sin(x)</code>	Returns the sine of x (x is in radians)
<code>sqrt(x)</code>	Returns the square root of x
<code>tan(x)</code>	Returns the tangent of an angle



