



Carmelo José Jaén Díaz



# Objetos nativos. Math

- 
- C.F.G.S. DAW
  - 6 horas semanales
  - Mes aprox. de impartición: Nov
  - iPasen - [cjaedia071@g.educaand.es](mailto:cjaedia071@g.educaand.es)

# Índice



Objetivo

Glosario

Interacción persona - ordenador

Objetivos

Características. Usable.

Características. Visual.

Características. Educativo y actualizado.

# OBJETIVO

---



- Aprender cómo se puede manejar el tiempo en JavaScript.
- Saber cómo se programa la ejecución de funciones de forma puntual y periódica en el tiempo.
- Registrar en el front-end información de navegación, usuario, etc., mediante cookies o mediante el almacenamiento local que ofrece el nuevo estándar HTML5.
- Profundizar en el concepto de objeto.
- Conocer y manejar funciones relativas al lenguaje sobre arrays, strings, números.

# GLOSARIO

---



**Backtracking.** Estrategia utilizada en algoritmos que resuelven problemas que tienen ciertas restricciones. Este término fue creado por primera vez por el matemático D. H. Lehmer en la década de los cincuenta.

**BOM (Browser Object Model).** Convención específica implementada por los navegadores para que JavaScript pudiese hacer uso de sus métodos y propiedades de forma uniforme.

**Expresión regular.** Secuencia de caracteres que forman un patrón determinado, generalmente un patrón de búsqueda.

**NaN.** Propiedad que indica Not a Number (valor no numérico).

**Objeto window.** Aquel que soportan todos los navegadores y que representa la ventana del navegador. Se estudiará en profundidad en capítulos posteriores.

**URI (Uniform Resource Identifier).** Cadena de caracteres que identifica un recurso en una red de forma unívoca. Una URI puede ser una URL, una URN o ambas.

# GLOSARIO

---



URN. Localizador de recursos en la web que funciona de forma parecida a una URL, pero su principal diferencia es que no indica exactamente dónde se encuentra dicho objeto.

# INTRODUCCIÓN

---



Como os podréis imaginar, el objeto nativo `Math` tiene propiedades y métodos para constantes y funciones matemáticas.

Se diferencia de los demás objetos nativos en que no puede editarse, es decir, los métodos y propiedades del objeto nativo `Math` son estáticos.

A continuación, vamos a ver cuáles son las propiedades, entre las que nos encontramos el clásico *`Math.PI`*, que contiene el número Pi.

Y posteriormente, veremos la utilidad del método *`Math.random()`*, que genera números aleatorios; y *`Math.max()`* junto con *`Math.min()`* que seleccionan el mayor o menor de una lista de valores.

# INTRODUCCIÓN

---



¿Sabías que hasta la versión de `#Javascript #ES7` para crear potencias había que usar `Math.pow(base,exponente)` y ahora puedes usar directamente `base**exponente`? ¡Así es!

# CARACTERÍSTICAS

---



- Permite realizar operaciones matemáticas.
- No es un constructor: NO usamos `let x = new Math();`  
*//Sus propiedades y métodos deben ser llamados usando directamente el nombre del objeto.*
- Ej: `let pi = Math.PI; let raiz = Math.sqrt(36);`

Podemos encontrar todos los métodos de *Math* en [W3Schools.com>JavaScript>JS References](#).



# PROPIEDADES DE *Math*

---



Al ser propiedades no hace falta poner los paréntesis al final cuando se declaran.

!!!Recuerda que todas las propiedades deben escribirse en mayúscula!!!

Entre las distintas propiedades del objeto *Math* se encuentran:

- E, devuelve el número de Euler (approx. 2.718).
- LN2, devuelve el logaritmo neperiano de 2 (approx. 0.693)
- LN10, devuelve el logaritmo neperiano de 10 (approx. 2.302)
- LOG2E, devuelve el logaritmo en base-2 de E (approx. 1.442)
- LOG10E, devuelve el logaritmo en base-10 de E (approx. 0.434)
- PI, devuelve el número PI (approx. 3.14)
- SQRT1\_2, devuelve la raíz cuadrada de 1/2 (approx. 0.707)
- SQRT2, devuelve la raíz cuadrada de 2 (approx. 1.414)

```
let pi = Math.PI;  
alert ("El número PI vale:"  
+pi);
```

# MÉTODOS DE *Math*.

## *random*



Método: random()

**Finalidad:** devuelve un número aleatorio entre 0 y 1. Al multiplicarlo por el número deseado, se obtiene un número aleatorio entre 0 y ese número.

Ejemplo de uso:

```
alert ("Número aleatorio: "+Math.random());
```

# MÉTODOS DE *Math*.

## *max y min*



Método: max(<lista de valores>) y min (<lista de valores>)

Finalidad: devuelve el número mayor (o menor) de una lista de valores.

Ejemplo de uso:

```
alert ("Máximo de valores: "+Math.max(1,5,3,9,6)); // Devuelve 9  
alert ("Mínimo de valores: "+Math.min(1,5,3,9,6)); // Devuelve 1
```

# MÉTODOS DE *Math*.

## *round*



Método: round(<número>)

Finalidad: devuelve el número redondeado al alza o baja dependiendo de su valor.

Ejemplo de uso:

```
let a = 4.3; //Devuelve 4
let b = 4.7; //Devuelve 5
let c = 4.5; //Devuelve 5
alert ("Redondeo de "+c+" es: "+Math.round(c));
```

# MÉTODOS DE *Math*.

## *ceil*



Método: `ceil(<número>)`

Finalidad: devuelve el número redondeado siempre al alza.

Ejemplo de uso:

```
let a = 4.3; //Devuelve 5
alert ("Redondeo de "+a+" es: "+Math.ceil(a));
```

# MÉTODOS DE *Math*.

## *floor*



Método: floor(<número>)

Finalidad: devuelve el número redondeado siempre a la baja.

Ejemplo de uso:

```
let b = 4.7; //Devuelve 4
alert ("Redondeo de "+b+" es: "+Math.floor(b));
```

# MÉTODOS DE *Math*.

## Obtener un número aleatorio entre 0 y N



Haciendo uso del método *random* y de los métodos de redondeo, podemos

Ejemplo de uso:

```
let n = <número deseado>;  
let aleatorio = Math.floor(Math.random()*n+1);  
alert ("Aleatorio es: "+aleatorio);
```

# MÉTODOS DE *Math*.

## *pow*



Método: `pow(<base>,<exponente>)`

Finalidad: devuelve el valor de la potencia.

Ejemplo de uso:

```
alert ("La potencia de 3 al cuadrado es: "+Math.pow(3,2));
```



# MÉTODOS DE *Math*.

## *pow*



Método: `pow(<base>,<exponente>)`

Finalidad: devuelve el valor de la potencia.

Ejemplo de uso:

```
alert ("La potencia de 3 al cuadrado es: "+Math.pow(3,2));
```

# MÉTODOS DE *Math*.

## *sqrt*



Método: `sqrt(<numero>)`

Finalidad: devuelve la raíz cuadrada del número.

Ejemplo de uso:

```
alert("Raíz cuadrada de 36 es: "+Math.sqrt(36));
```

# MÉTODOS DE *Math*.

## *abs*



Método: `abs(<número>)`

Finalidad: devuelve el valor absoluto de un número.

Ejemplo de uso:

```
alert ("Valor absoluto de -5: "+Math.abs(-5));
```

# MÉTODOS DE *Math*.

## Métodos trigonométricos



Mediante los siguientes métodos de `Math`, es posible obtener el valor de:

- `sin(<número>)`, devuelve el seno de un número en radianes.
- `cos(<número>)`, devuelve el coseno de un número en radianes.
- `tan(<número>)`, devuelve la tangente de un número en radianes.
- `asin(<número>)`, devuelve el arcoseno de un número en radianes.
- `acos(<número>)`, devuelve el arcocoseno de un número en radianes.
- `atan(<número>)`, devuelve la arcotangente de un número en radianes.
- `atan2(<coordenada Y>, (<coordenada X>)`, devuelve el ángulo (en radianes) en el plano entre el eje positivo (x) y la línea desde el (0,0) y el punto (x, y), para `Math.atan2(y, x)`.

