

SOC

Servei d'Ocupació
de Catalunya



Generalitat
de Catalunya



Unió Europea
Fons social europeu
L'FSE inverteix en el teu futur



MÓDULO 1. MF0951_2 INTEGRAR COMPONENTES SOFTWARE EN PÁGINAS WEB

UNIDAD FORMATIVA 1. UF1305 INTEGRAR COMPONENTES SOFTWARE EN PÁGINAS WEB.





3.6

Fecha y Hora - Math - Random

_Creación new Date()

_Visualización de fechas

_Formatos

_Acceso a los componentes de la fecha

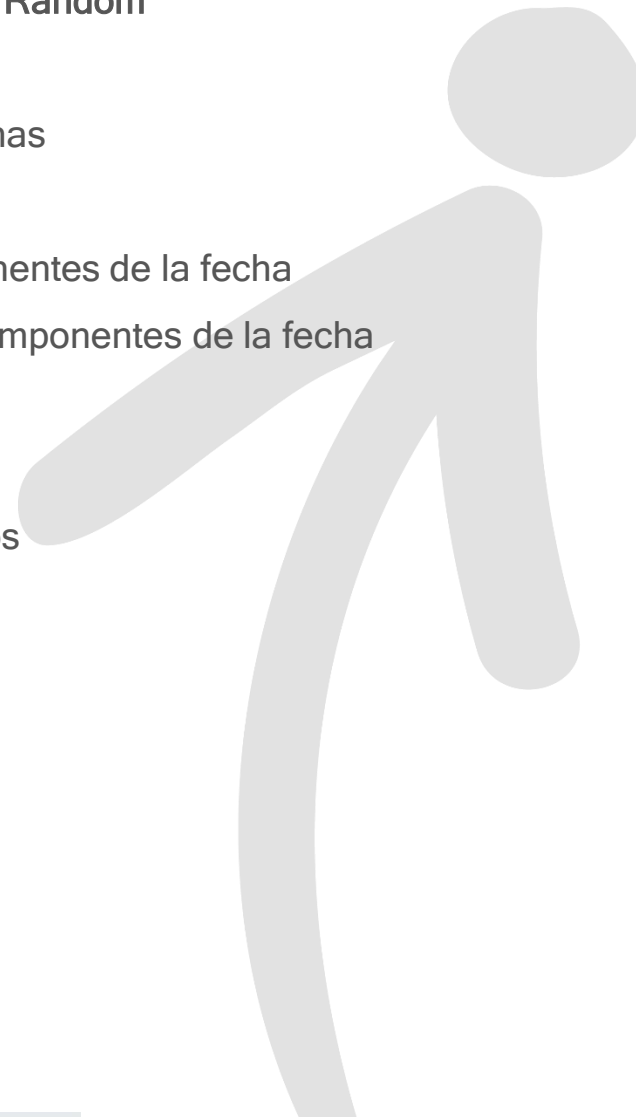
_Estableciendo los componentes de la fecha

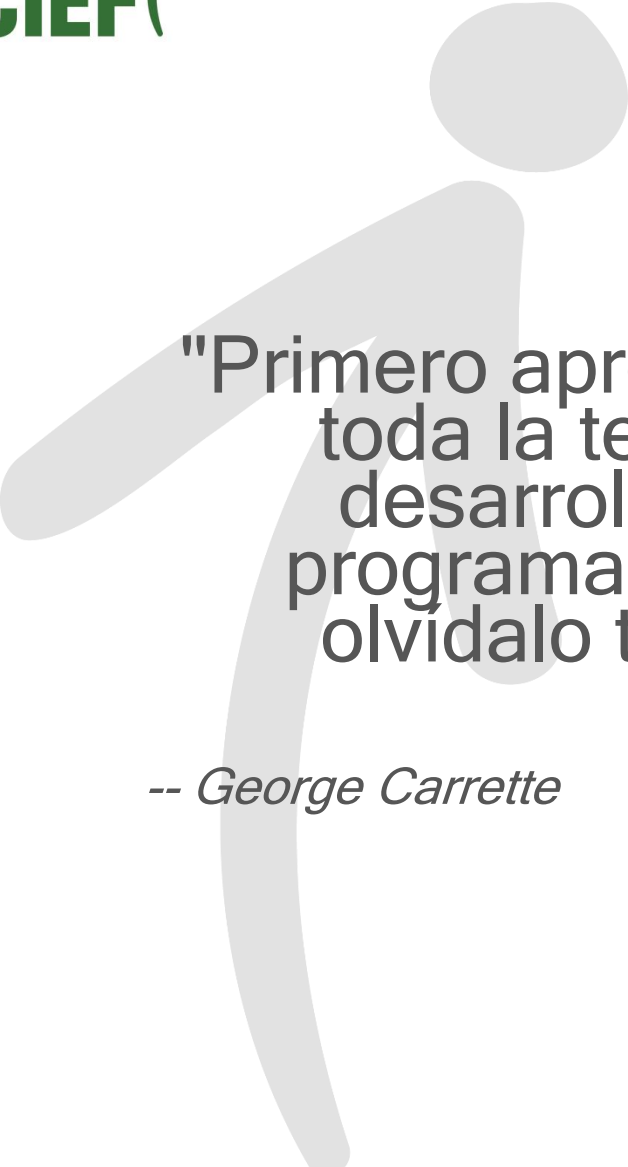
_Comparar fechas

Math

_Métodos matemáticos

_Random



A large, light grey silhouette of a person is positioned in the background, behind the main text. The silhouette is simple, with a circular head and a body that tapers towards the bottom, giving it a modern, abstract feel.

"Primero aprende informática y toda la teoría. Después desarrolla un estilo de programación. Entonces, olvídale todo y hackea"

-- *George Carrette*



Javascript.I



3.6.

Fechas horas



3.6 Fechas y horas_Creación new Date()

Date. Este objeto almacena la fecha, la hora, y brinda métodos para administrarla.

Para crear un nuevo objeto Date se lo instancia con new Date() junto con uno de los siguientes argumentos:

new Date()

Sin argumentos - crea un objeto Date para la fecha y la hora actuales:

```
let now = new Date();  
alert( now ); // muestra en pantalla la fecha y la hora actuales
```

3.6 Fechas y horas_Creación new Date()

`new Date(year, month, ...)`

crea un nuevo objeto de fecha con una **fecha y hora especificadas** .

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

```
<!DOCTYPE html>
<html>
<body>

<h2>JavaScript new Date()</h2>

<p>Using new Date(7 numbers), creates a new date object with the specified date and
time:</p>

<p id="demo"></p>

<script>
const d = new Date(2018, 11, 24, 10, 33, 30, 0);
document.getElementById("demo").innerHTML = d;
</script>

</body>
</html>
```

Nota: JavaScript cuenta los meses
del 0 al 11 :
enero = 0 .
diciembre = 11

```
const d = new Date(2018, 15, 24, 10, 33, 30);
```

```
const d = new Date(2019, 3, 24, 10, 33, 30);
```

JavaScript new Date()

Using new Date(7 numbers), creates a new date object with the specified date and time:

Mon Dec 24 2018 10:33:30 GMT+0100 (hora estándar de Europa central)

3.6 Fechas y horas_Creación new Date()



`new Date(year, month, ...)`

crea un nuevo objeto de fecha con una **fecha y hora especificadas** .

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

```
<!DOCTYPE html>
<html>
<body>

<h2>JavaScript new Date()</h2>

<p>Using new Date(7 numbers), creates a new date object with the specified date and
time:</p>

<p id="demo"></p>

<script>
const d = new Date(2018, 11, 24, 10, 33, 30, 0);
document.getElementById("demo").innerHTML = d;
</script>

</body>
</html>
```

JavaScript new Date()

Using new Date(7 numbers), creates a new date object with the specified date and time:

Mon Dec 24 2018 10:33:30 GMT+0100 (hora estándar de Europa central)



3.6 Fechas y horas_Creación new Date()

`new Date(year, month, ...)`

6 números especifican año, mes, día, hora, minuto, segundo:

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

5 números especifican año, mes, día, hora y minuto:

```
const d = new Date(2018, 11, 24, 10, 33);
```

4 números especifican año, mes, día y hora:

```
const d = new Date(2018, 11, 24, 10);
```

3 números especifican año, mes y día:

```
const d = new Date(2018, 11, 24);
```

2 números especifican año y mes:

```
const d = new Date(2018, 11);
```

No puede omitir el mes. Si proporciona solo un parámetro, se tratará como milisegundos.

3.6 Fechas y horas_Creación new Date()

nueva fecha (*cadena de fecha*)

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

```
<!DOCTYPE html>
<html>
<body>

<h2>JavaScript new Date()</h2>
<p>A Date object can be created with a specified date and
time:</p>

<p id="demo"></p>

<script>
const d = new Date("October 13, 2014 11:13:00");
document.getElementById("demo").innerHTML = d;
</script>

</body>
</html>
```

JavaScript new Date()

A Date object can be created with a specified date and time:

Mon Oct 13 2014 11:13:00 GMT+0200 (hora de verano de Europa central)

3.6 Fechas y horas_Visualización

JavaScript generará (por defecto) las fechas en formato de cadena de texto completo:

```
Sun Mar 27 2022 12:29:33 GMT+0200 (hora de verano de Europa central)
```

Cuando muestra un objeto de fecha en HTML, se convierte automáticamente en una cadena, con el `toString()` método.

```
<p id="demo"></p>

<script>
const d = new Date();
document.getElementById("demo").innerHTML = d.toString();
</script>
```

El `toUTCString()` método convierte una fecha en una cadena UTC (un estándar de visualización de fechas).

```
const d = new Date();
d.toUTCString();
```

El `toDateString()` método convierte una fecha a un formato más legible:

```
<script>
const d = new Date();
document.getElementById("demo").innerHTML = d.toDateString();
</script>
```

JavaScript `toDateString()`

The `toDateString()` method converts a date to a date string:

Sun Mar 27 2022

3.6 Fechas y horas_Formatos



En general, hay 3 tipos de formatos de entrada de fecha de JavaScript:

Fecha ISO	"2015-03-25" (El estándar internacional)
Cita corta	"25/03/2015"
Fecha larga	"25 de marzo de 2015" o "25 de marzo de 2015"

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:

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

```
<script>  
const d = new Date();  
document.getElementById("demo").innerHTML = d.toString();  
</script>
```





Method	Description
getFullYear()	Get the year as a four digit number (yyyy)
getMonth()	Get the month as a number (0-11)
getDate()	Get the day as a number (1-31)
getHours()	Get the hour (0-23)
getMinutes()	Get the minute (0-59)
getSeconds()	Get the second (0-59)
getMilliseconds()	Get the millisecond (0-999)
getTime()	Get the time (milliseconds since January 1, 1970)
getDay()	Get the weekday as a number (0-6)
Date.now()	Get the time. ECMAScript 5.

```

<!DOCTYPE html>
<html>
<body>

<h2>JavaScript getFullYear()</h2>
<p>The getFullYear() method returns the full year of a date:</p>

<p id="demo"></p>

<script>
const d = new Date();
document.getElementById("demo").innerHTML = d.getFullYear();
</script>

</body>
</html>

```

JavaScript getFullYear()

The `getFullYear()` method returns the full year of a date:

2022



Method	Description
getFullYear()	Get the year as a four digit number (yyyy)
getMonth()	Get the month as a number (0-11)
getDate()	Get the day as a number (1-31)
getHours()	Get the hour (0-23)
getMinutes()	Get the minute (0-59)
getSeconds()	Get the second (0-59)
getMilliseconds()	Get the millisecond (0-999)
getTime()	Get the time (milliseconds since January 1, 1970)
getDay()	Get the weekday as a number (0-6)
Date.now()	Get the time. ECMAScript 5.

JavaScript getMonth()

The `getMonth()` method returns the month as a number.

You can use an array to display the name of the month:

March

```
<!DOCTYPE html>
<html>
<body>

<h2>JavaScript getMonth()</h2>
<p>The getMonth() method returns the month as a number.</p>
<p>You can use an array to display the name of the month:</p>

<p id="demo"></p>

<script>
const months =
["January","February","March","April","May","June","July","August","Septem
ber","October","November","December"];

const d = new Date();
let month = months[d.getMonth()];
document.getElementById("demo").innerHTML = month;
</script>

</body>
</html>
```



setDate()	Set the day as a number (1-31)
setFullYear()	Set the year (optionally month and day)
setHours()	Set the hour (0-23)
setMilliseconds()	Set the milliseconds (0-999)
setMinutes()	Set the minutes (0-59)
setMonth()	Set the month (0-11)
setSeconds()	Set the seconds (0-59)

El `setFullYear()` método establece el año de un objeto de fecha. En este ejemplo a 2020: mes y día opcional.

JavaScript setFullYear()

The `setFullYear()` method can optionally set month and day.

Please note that month counts from 0. December is month 11:

Thu Dec 03 2020 12:51:49 GMT+0100 (hora estándar de Europa central)

```
<!DOCTYPE html>
<html>
<body>

<h2>JavaScript setFullYear()</h2>
<p>The setFullYear() method can optionally set month and day.</p>
<p>Please note that month counts from 0. December is month 11:</p>

<p id="demo"></p>

<script>
const d = new Date();
d.setFullYear(2020, 11, 3);
document.getElementById("demo").innerHTML = d;
</script>


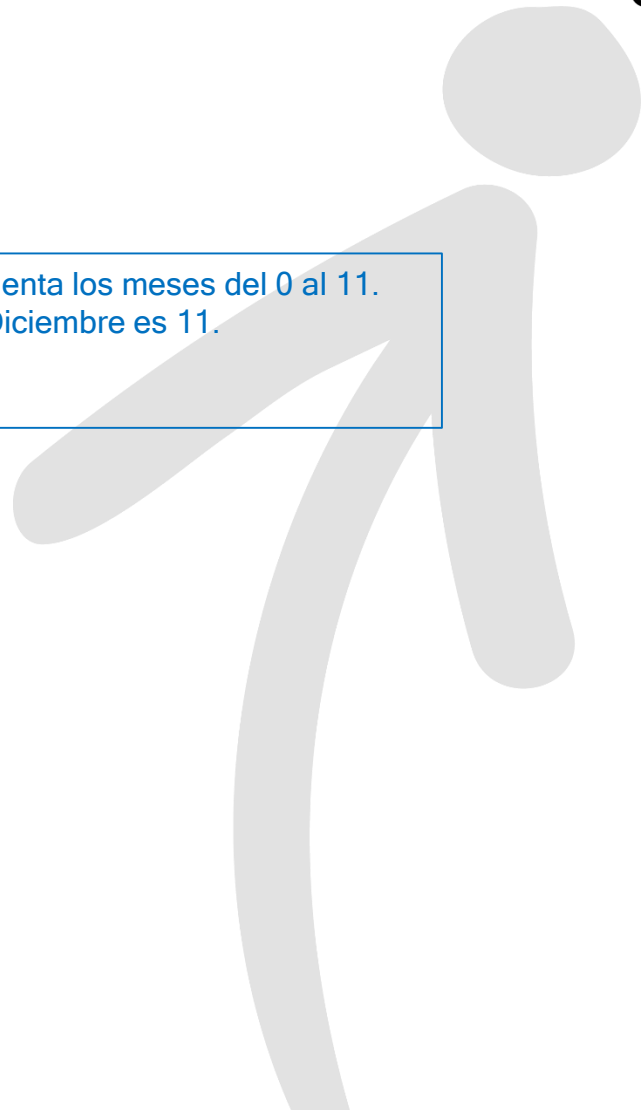
</body>
</html>
```

3.6 Fechas y horas_Comparar fechas

Las fechas se pueden comparar fácilmente.

El siguiente ejemplo compara la fecha de hoy con el 14 de enero de 2100:

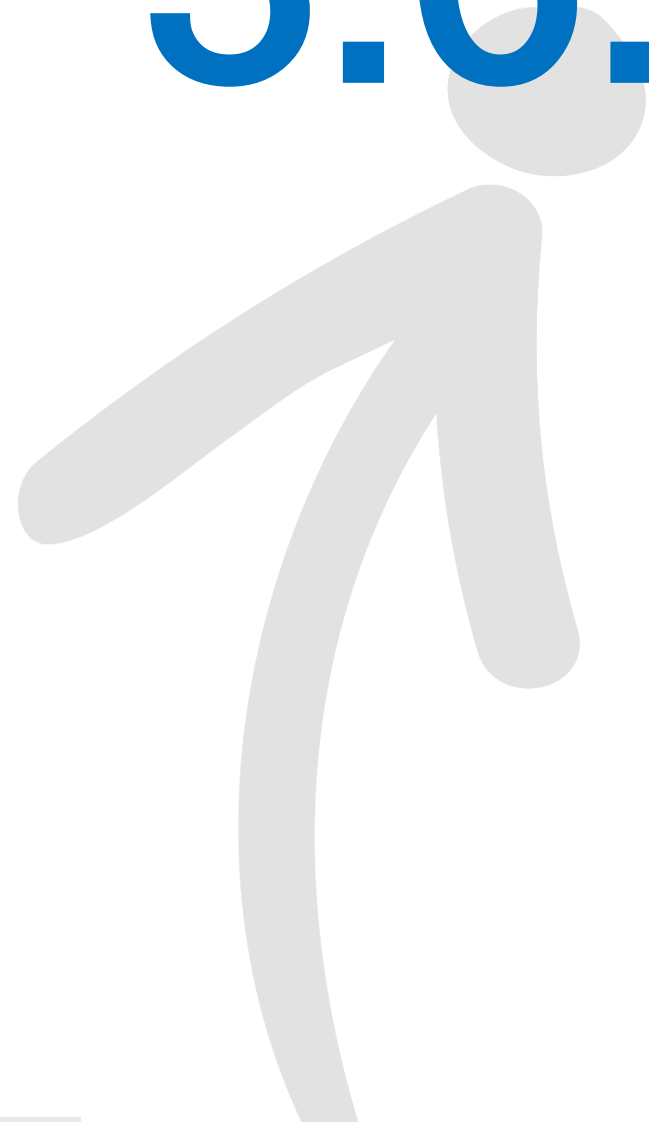
```
let text = "";  
const today = new Date();  
const someday = new Date();  
someday.setFullYear(2100, 0, 14);  
  
if (someday > today) {  
    text = "Today is before January 14, 2100.";  
} else {  
    text = "Today is after January 14, 2100.";  
}
```



JavaScript cuenta los meses del 0 al 11.
Enero es 0. Diciembre es 11.

3.6.

Math





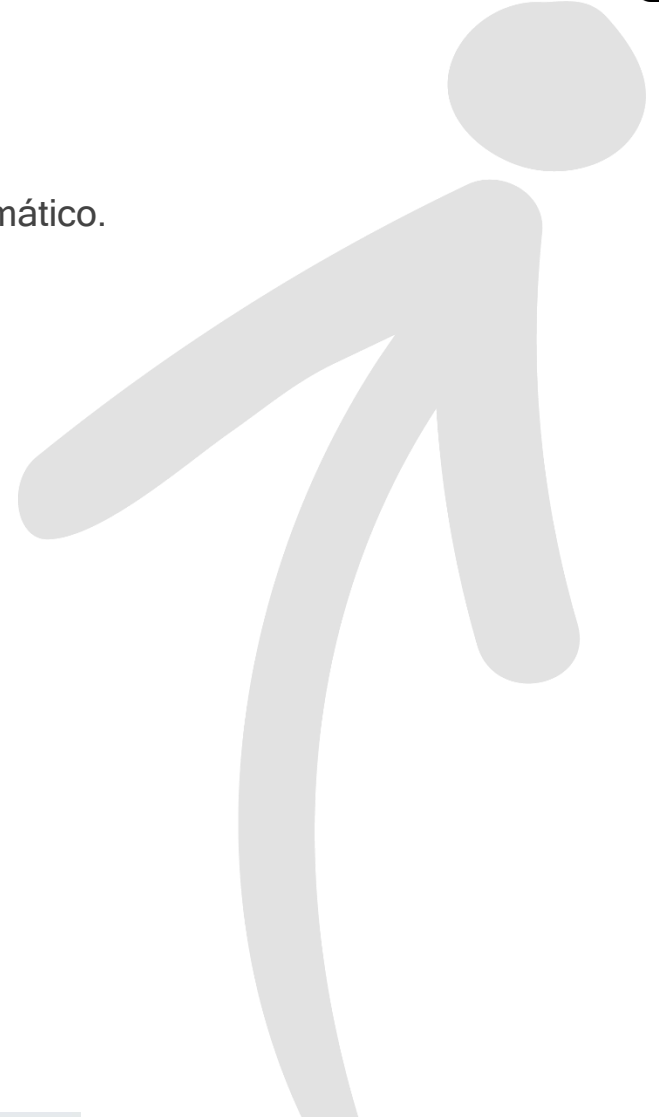
A diferencia de otros objetos, el objeto Math no tiene constructor.

El objeto Math es estático.

Todos los métodos y propiedades se pueden usar sin crear primero un objeto matemático.

Constantes

```
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
```



3.6 Math_métodos



Number to Integer

<code>Math.round(x)</code>	Returns x rounded to its nearest integer	
<code>Math.ceil(x)</code>	Returns x rounded up to its nearest integer	4,4 (5)
<code>Math.floor(x)</code>	Returns x rounded down to its nearest integer	4,9 (4)
<code>Math.trunc(x)</code>	Returns the integer part of x (new in ES6)	4,7 (4)

Math.round() devuelve el entero más cercano:

```
Math.round(4.6);//5
Math.round(4.5);//5
Math.round(4.4);//4
```

Math.sign(x) devuelve si x es negativo, nulo o positivo //-1,0,1

Math.pow(x, y) devuelve el valor de x a la potencia de y

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

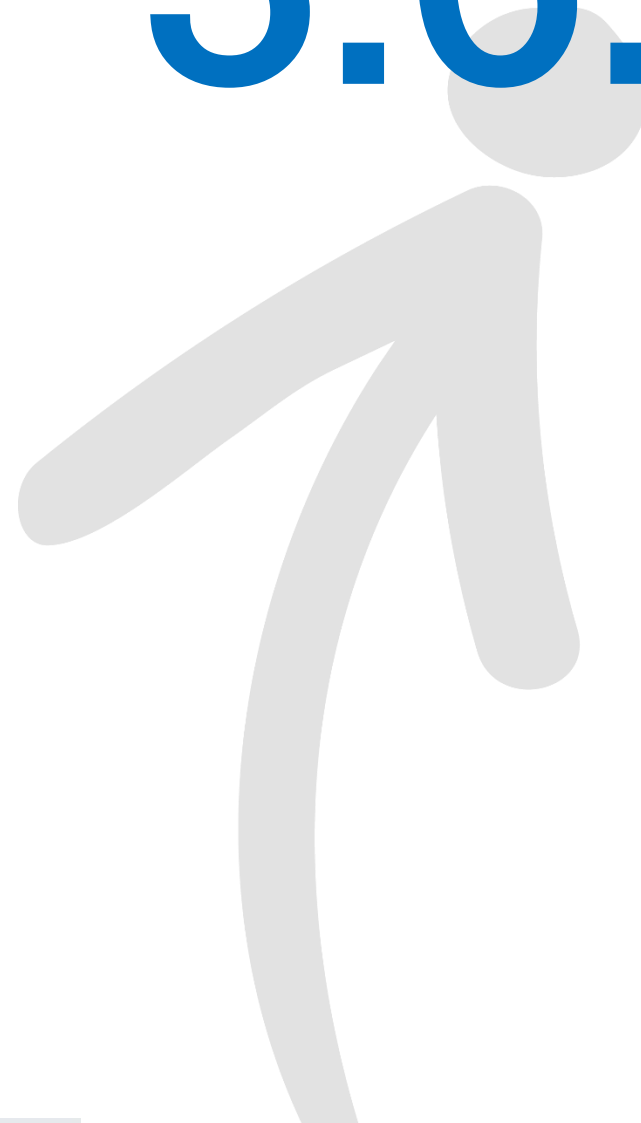
Math.abs(x) devuelve el valor absoluto (positivo) de x:

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);
```

3.6.

Random



3.6 Math_Random



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

```
<!DOCTYPE html>
<html>
<body>

<h2>JavaScript Math.random()</h2>

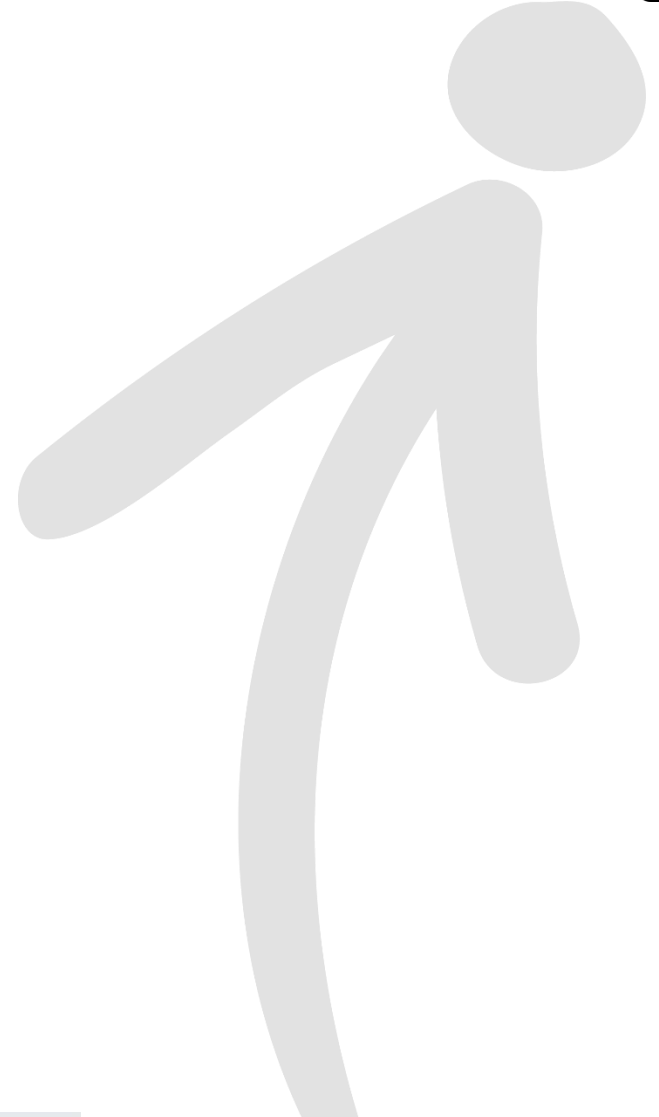
<p>Math.random() returns a random number between 0 and 1:</p>

<p id="demo"></p>

<p>Tip: Click on "Run" several times.</p>

<script>
document.getElementById("demo").innerHTML = Math.random();
</script>

</body>
</html>
```



3.6 Random_Aleatorios enteros

Math.random() used with Math.floor() puede usarse para devolver enteros aleatorios

```
// Returns a random integer from 0 to 9:  
Math.floor(Math.random() * 10);  
  
// Returns a random integer from 0 to 10:  
Math.floor(Math.random() * 11);  
  
// Returns a random integer from 0 to 99:  
Math.floor(Math.random() * 100);  
  
// Returns a random integer from 0 to 100:  
Math.floor(Math.random() * 101);  
  
// Returns a random integer from 1 to 10:  
Math.floor(Math.random() * 10) + 1;  
  
// Returns a random integer from 1 to 100:  
Math.floor(Math.random() * 100) + 1;
```

```
function getRndInteger(min, max) {  
    return Math.floor(Math.random() * (max - min) ) +  
    min;  
}  
  
function getRndInteger(min, max) {  
    return Math.floor(Math.random() * (max - min) ) +  
    min;  
}
```

excluidos

incluidos





Barcelona

Francesc Tàrraga 14
08027 Barcelona
93 351 78 00

Madrid

Campanar 12
28028 Madrid
91 502 13 40

Reus

Alcalde Joan Bertran 34-38
43202 Reus
977 31 24 36

info@grupcief.com

www.grupcief.com

