



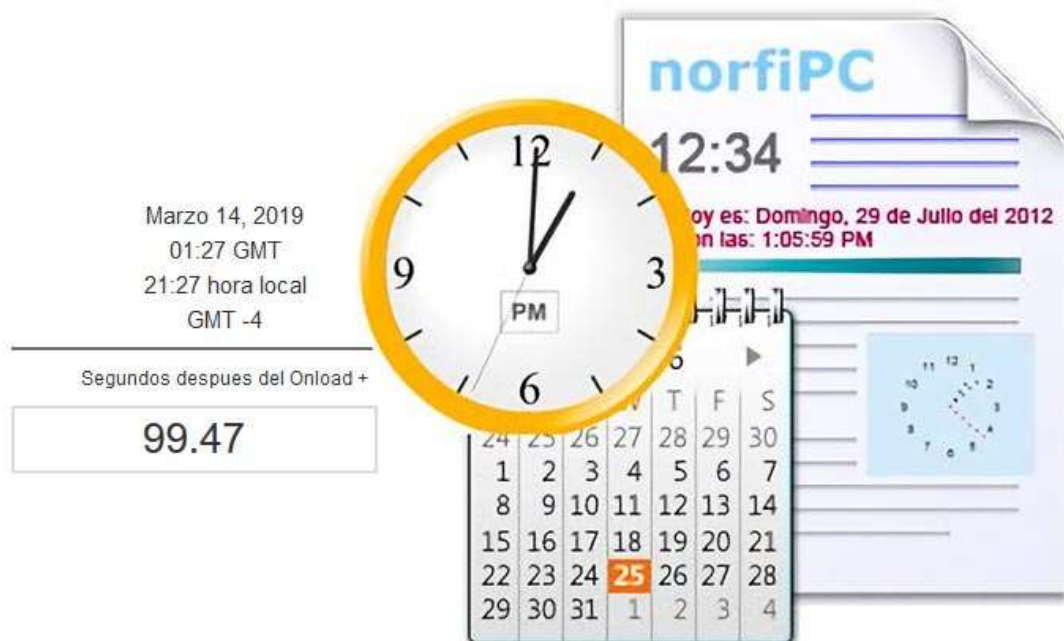
Docs_

Mostrar fecha y hora



Como mostrar la fecha y hora en las páginas web usando Javascript

Códigos para insertar en una página y así mostrar la hora, el día, el día de la semana, el mes, el año, además relojes y calendarios que embellecen tu página y la hacen más útil y funcional para los usuarios. Ejemplos prácticos para copiar y pegar.



Javascript facilita agregar diferentes funcionalidades en las páginas web. Al ser un lenguaje que se ejecuta del lado del cliente o navegador, permite darle dinamismo a las páginas, es decir hacer que cambie su contenido sin tener que recargarla más del servidor.

Una de estas funcionalidades es usar Javascript para mostrar la fecha y hora en las páginas de una infinidad de formas.

Diversos programas de edición web y CMS brindan la posibilidad de crear páginas que incluyen los scripts necesarios para mostrar la fecha y hora, así como calendarios, almanaques, etc.

No obstante si eres emprendedor o deseas un resultado diferente, puedes conocer cómo hacerlo manualmente.

Puedes hacerlo de formas diferentes, directamente copiando y pegando los códigos que aparecen en esta página, modificarlos previamente a tu gusto o hasta crearlos usando las funciones que se muestran más adelante.

No es nada difícil, ni se necesita tener conocimientos avanzados de Javascript. Para crearlos usando las funciones solo es necesario algunos conocimientos elementales, para modificarlos solo un poco de razonamiento lógico y para usarlos copiando y pegando, solo saber el lugar donde se desean mostrar.

En todos los casos se puede usar CSS para darle un estilo único.

Funciones de JavaScript para mostrar la fecha y hora

A continuación, algunas de las funciones de Javascript que se pueden emplear para mostrar la fecha y hora, cada una devuelve parámetros diferentes pero se pueden combinar para crear el resultado que se desee. El objeto *Date* es usado para ejecutar mediante las diferentes funciones cualquiera de los parámetros necesarios.

Las funciones más utilizadas son las siguientes:

<code>getDate()</code>	Devuelve el día del mes (entre el 1 y el 31)
<code>getDay()</code>	Devuelve el día de la semana (entre el 0 y el 6)
<code>getMonth()</code>	Devuelve el mes (entre el 0 y el 11)
<code>getFullYear()</code>	Devuelve el año (en formato de 4 dígitos)
<code>getHours()</code>	Devuelve la hora (entre el 0 y el 24)
<code>getMinutes()</code>	Devuelve los minutos (desde 0 a 59)
<code>getSeconds()</code>	Devuelve los segundos (desde 0 a 59)
<code>getTime()</code>	Devuelve el número de milisegundos desde el 1ro de Enero de 1970
<code>getTimezoneOffset()</code>	Devuelve la diferencia de horario en minutos entre la hora local y GMT (Meridiano de Greenwich)
<code>getUTCHours()</code>	Devuelve la hora de acuerdo a UTC (Tiempo Universal Coordinado)

¿Como usar las funciones de Javascript?

Para crear una función es necesario primero establecer una variable con *new Date()* para definir la fecha y hora actual, a continuación se agrega la función a ejecutar en cada caso.

En caso de usar solamente *Date* se mostrará el resultado de la variable en bruto, es decir la secuencia del día de la semana, el mes, el día, el año, la hora GMT y el huso horario, todo en inglés.

Pruébalo en el siguiente botón:

Esto no resulta de utilidad, es necesario agregar las funciones por separado y combinarlas para lograr un resultado que valga la pena.

Ejemplos prácticos del uso de funciones para obtener la fecha y hora

Algunos ejemplos escribiendo con *document.write* el resultado de la función en la página.

De la siguiente forma con *getDate()* se escribe el día del mes:

```
<script type="text/javascript">

var d = new Date();

document.write(d.getDate());

</script>
```

Para escribir la hora utiliza *getHours()*:

```
<script type="text/javascript">

var d = new Date();

document.write(d.getHours());

</script>
```

Así sucesivamente se puede usar con cada función.

Para usar varias encadenadas hazlo de la siguiente forma:

```
<script type="text/javascript">

var d = new Date();

document.write('Fecha: '+d.getDate(), '<br>Dia de la semana: '+d.getDay(), '<br>Mes (
0 al 11): '+d.getMonth(), '<br>Año: '+d.getFullYear(), '<br>Hora: '+d.getHours(), '<br
>Hora UTC: '+d.getUTCHours(), '<br>Minutos: '+d.getMinutes(), '<br>Segundos: '+d.getSe
conds());

</script>
```

Al usarlo se mostrará:

Fecha: 15

Dia de la semana: 0

Mes (0 al 11): 4

Año: 2022

Hora: 11

Hora UTC: 9

Minutos: 41

Segundos: 16

Para mostrar la diferencia en minutos con la zona horaria de Greenwich utiliza *getTimezoneOffset()*:

```
<script type="text/javascript">

var d = new Date();

document.write('<br>Diferencia de GTM en minutos: '+d.getTimezoneOffset());

</script>
```

En todos los casos la función *new Date()* como primer paso determina la hora y fecha actual de acuerdo a lo que indica Windows, así como la zona horaria. Todos los parámetros de Javascript son globales, o sea estándar para todos los equipos e idiomas, por lo que algunos de ellos necesitan ser convertidos antes de mostrarlos, por ejemplo:

- La función *getDay()* devuelve el día de la semana, pero en un formato entre el 0 y el 6, donde el 0 corresponde al domingo.
- La función *getMonth()* devuelve el mes comprendido entre el 0 y el 11, donde 0 corresponde a Enero.

Para mostrar el resultado de esas y otras funciones correctamente, se pueden emplear algunos trucos.

Trucos para convertir el resultado de funciones

- En el siguiente ejemplo se traducen los días de la semana a los nombres en español creando un array.

```
<script type="text/javascript">

var d=new Date();

var dia=new Array(7);

dia[0]="Domingo";

dia[1]="Lunes";

dia[2]="Martes";

dia[3]="Miercoles";

dia[4]="Jueves";

dia[5]="Viernes";

dia[6]="Sabado";

document.write("Hoy es: " + dia[d.getDay()]);

</script>
```

Al usarlo se mostrará:

Hoy es: Domingo

- En el caso del mes, en que la función *getMonth()* siempre devuelve el resultado con un atraso de una cifra, se puede utilizar:

```
<script type="text/javascript">

var hoy = new Date();

var m = hoy.getMonth() + 1;

var mes = (m < 10) ? '0' + m : m;

document.write('Este mes es el número: '+mes);

</script>
```

- Si deseas que se muestre el nombre del mes entonces utiliza:

```
<script type="text/javascript">

var mm=new Date();

var m2 = mm.getMonth() + 1;

var mesok = (m2 < 10) ? '0' + m2 : m2;

var mesok=new Array(12);

mesok[0]="Enero";

mesok[1]="Febrero";

mesok[2]="Marzo";

mesok[3]="Abril";

mesok[4]="Mayo";
```

```
mesok[5]="Junio";  
  
mesok[6]="Julio";  
  
mesok[7]="Agosto";  
  
mesok[8]="Septiembre";  
  
mesok[9]="Octubre";  
  
mesok[10]="Noviembre";  
  
mesok[11]="Diciembre";  
  
document.write("Este mes es: " + mesok[mm.getMonth()]);  
  
</script>
```

Un ejemplo interesante del uso de JavaScript para usarlo en tu blog o sitio web.

El tiempo que llevas ya en esta página:

Lee en otro artículo como usarlo en una página: [Mostrar en una página el tiempo conectado a internet con JavaScript](#)

Ejemplos prácticos de códigos para insertar en las paginas web

Los siguientes cuadros de texto contienen códigos de varios ejemplos prácticos diferentes para usar en las páginas web, da un clic encima del que te interese, cópialo y pégalo en el lugar del código HTML donde deseas que se muestre.

Se ha tratado de incluir solo una selección de los más sencillos, para evitar que influyan en el rendimiento de la página.

Códigos para mostrar la fecha de tres formas diferentes

```
<div style="float:right;">

<script type="text/javascript">

//<![CDATA[

var today = new Date();

var m = today.getMonth() + 1;

var mes = (m < 10) ? '0' + m : m;

    document.write('Fecha: '+today.getDate(), '/' +mes, '/' +today.getYear());

//]]>    </script></div>
```

```
<script type="text/javascript">

//<![CDATA[

var date = new Date();

var d = date.getDate();

var day = (d < 10) ? '0' + d : d;

var m = date.getMonth() + 1;

var month = (m < 10) ? '0' + m : m;
```

```

var yy = date.getFullYear();

var year = (yy < 1000) ? yy + 1900 : yy;

document.write(day + "/" + month + "/" + year);

//]]>

</script>

```

```
<script type="text/javascript">
```

```

//<![CDATA[
function makeArray() {
for (i = 0; i<makeArray.arguments.length; i++)
this[i + 1] = makeArray.arguments[i];}
var months = new makeArray('Enero', 'Febrero', 'Marzo', 'Abril', 'Mayo',
'Junio', 'Julio', 'Agosto', 'Septiembre', 'Octubre', 'Noviembre', 'Diciembre');
var date = new Date();
var day = date.getDate();
var month = date.getMonth() + 1;
var yy = date.getFullYear();
var year = (yy < 1000) ? yy + 1900 : yy;
document.write("Hoy es " + day + " de " + months[month] + " del " + year);
//]]>

</script>

```

Codigo para insertar un pequeño reloj digital:

```

<script type="text/javascript">

function startTime(){

```

```

today=new Date();

h=today.getHours();

m=today.getMinutes();

s=today.getSeconds();

m=checkTime(m);

s=checkTime(s);

document.getElementById('reloj').innerHTML=h+":"+m+":"+s;

t=setTimeout('startTime()',500);}

function checkTime(i)

{if (i<10) {i="0" + i;}return i;}

window.onload=function(){startTime();}

</script>

<div id="reloj" style="font-size:20px;"></div>

```

Código para mostrar la fecha y la hora en la barra de estado del navegador Internet Explorer, al cargar la página

```

<script type="text/javascript">

function sbClock() {

    var DateString=(new Date()).toString();

```

```
self.status=DateString.substring(0,3+DateString.lastIndexOf(':'));

setTimeout("sbClock()",200);}

window.onload=function(){sbClock();}

</script>
```

Consejos al insertar los scripts en el código HTML de la página

✓ Para agregar texto en el script de manera que se muestre en el resultado, enciérralo entre comillas simples, por ejemplo: 'texto'

✓ Si el código usado contiene alguno de los siguientes caracteres: <, >, ", es necesario encerrarlo de la siguiente forma para que valide:

```
<script type="text/javascript">  
  
//<br/><br/>CODIGO AQUI<br/><br/>//]]&gt;<br/><br/>&lt;/script&gt;</pre></div><div data-bbox="84 378 915 480" data-label="Text"><p>La validación es sumamente importante para las páginas, es una prueba para comprobar que el código es correcto y la página se puede mostrar perfectamente en todos los navegadores. Una página que no valide no aparece en los resultados de las búsquedas de Google.</p><hr/></div><div data-bbox="84 552 894 653" data-label="Text"><p>✓ En cualquiera de los ejemplos se puede modificar fácilmente el estilo con el que se muestran, es decir el tamaño del texto, el color, etc. para eso solo encierra el script dentro de un contenedor DIV que posea una etiqueta de estilo con los parámetros necesarios, por ejemplo:</p></div><div data-bbox="84 717 481 797" data-label="Text"><pre>&lt;div style="color:blue;font-size:18px;"&gt;<br/><br/>SCRIPT AQUI<br/><br/>&lt;/div&gt;</pre></div><div data-bbox="84 934 246 952" data-label="Page-Footer"><hr/><p>@mihifidem 01/2024</p></div>
```


✓ Cuando se empleen diferentes scripts en la página que requieran del evento *onload* para iniciarse, deben agruparse para evitar cualquier conflicto.

Hazlo de la siguiente forma:

```
<script type="text/javascript">

window.onload=function() {

funcion1();

funcion2();

funcion3();

}

</script>
```

Onload. Es un evento de Javascript, de efecto similar a un botón o un vínculo, ejecuta la función que se le indique, pero solo después de que la pagina haya concluido de cargar por completo. Resulta de mucha utilidad en las páginas que necesiten de cargar muchos elementos, ya sean scripts, imágenes, etc. Usándolo en esos casos se logra que se represente el contenido de la página con más rapidez en el navegador del usuario. Para usarlo solo es necesario crear la función necesaria e incluirla como se ha indica.