

OBJETO DATE

La fecha se almacena como los milisegundos que han pasado desde una fecha base que es el **1 de enero de 1970 a las 00:00:00**. Aún así, el rango de fechas que puede manejar es de -100000000 (-1 00 millones) hasta +100000000 (1 00 millones) de días relativos al de enero de 1970 UTC (Universal Coordinated Time).

Rango de valores para los campos de una fecha:

- Milisegundos 0-999
- Segundos y minutos 0-59
- Horas 0-23
- Días de la semana 0(domingo) – 6 (sábado)
- Día del mes 1- 31
- Mes 0 (enero). – 11 (diciembre)
- Año Desde 1970

Probando el constructor

- Vacío: Si no se pasa este parámetro, entonces se crea una instancia con la fecha y hora actuales.

```
new Date()
```

- Una cantidad de milisegundos: Se crea una fecha sumándolos a la fecha base.

```
New Date(86400000)
```

- “Mes en letras día, año horas:minutos:segundos”: El mes debe estar escrito en inglés. Si se omite la parte correspondiente a las horas, minutos o segundos, tomarán el valor 00.

```
new Date("August 15, 2008")  
new Date("August 15,2008 13:15:34")
```

- “Año, mes, día”: Estos valores deben ser numéricos enteros.

```
new Date(2008, 7, 15)
```

- “Año, mes, día, hora, minuto, segundos”: Exactamente igual que el anterior pero ampliado para incluir la hora.

```
new Date(2008, 7, 15, 13, 15, 34)
```

MÉTODOS DE LECTURA (GET...)

- **getTime ()** : Devuelve el número de milisegundos que han pasado desde la media noche del 1 de Enero de 1970.
- **getMilliseconds(), getSeconds(), getMinutes () y getHours ()** : Devuelven el campo correspondiente a los milisegundos, segundos, minutos u hora, respectivamente, de la instancia actual.
- **getDate () , getDay () , getMonth ()** : Nos permiten obtener el día del mes, el día de la semana y el mes de la instancia actual, representados siempre con un número entero.
- **getYear () , getFullYear ()** : Devuelven el año de la instancia actual. ~~getYear~~ siempre lo hará como un número de cuatro cifras.
- **getUTCMilliseconds(), getUTCSeconds(), getUTCMinutes(), getUTCHours(), getUTCDate(), getUTCDay(), getUTCMonth(), getUTCFullYear()**: Funcionan exactamente igual que sus análogos pero devolviendo el campo de acuerdo al UTC.
- **getTimezoneOffset()**: Devuelve la cantidad de minutos de diferencia que existen entre la hora UTC y la local. Esto nos será útil para averiguar la zona horaria en la que nos encontramos (pasando esos minutos a horas). Si sale un valor negativo, -60 por ejemplo, estaremos en la zona horaria GMT + 1. Sin embargo, el mismo valor pero positivo (60), indicaría que estamos en GMT-1.
- **now()**: Devuelve el número de milisegundos que han pasado desde la media noche del 1 de Enero de 1970.

Probando getDate(), getMonth(), getFullYear().

```
<SCRIPT language=javascript>
var miFecha=new Date()
document.write("La fecha del día de hoy: ");
document.write(miFecha.getDate()+"-"+(miFecha.getMonth()+1)+"-"+miFecha.get
FullYear())
</SCRIPT>
```

[Probando y arreglando getDay\(\)](#)

```
<SCRIPT language=javascript>

var miFecha=new Date()
document.write("miFecha.getDay()="+miFecha.getDay()+"<BR>")
var tbdays=new Array("Domingo", "Lunes", "Martes", "Miércoles", "Jueves",
"Viernes", "Sábado");
document.write("Hoy es "+tbdays[miFecha.getDay()])
</SCRIPT>
```

[Probando y arreglando getHours\(\), getMinutes\(\), getSeconds\(\), getMilliseconds\(\)](#)

```
<SCRIPT language=javascript>
var miFecha=new Date()
document.write("Hora : ");
document.write(miFecha.getHours()+":"+miFecha.getMinutes()+
":"+miFecha.getSeconds()+":"+miFecha.getMilliseconds())
document.write("<BR>");
// formateo de la hora
var h=miFecha.getHours();
if (h<10) {h = "0" + h}
var m=miFecha.getMinutes();
if (m<10) {m = "0" + m}
var s=miFecha.getSeconds();
if (s<10) {s = "0" + s}
var ms=miFecha.getMilliseconds();
document.write("Hora con un formato personalizado : ");
document.write(h+":"+m+":"+s+":"+ms);
</SCRIPT>
```

[Probando getTime\(\)](#)

```
<SCRIPT language=javascript>
var miFecha=new Date();
//cantidad de milisegundos desde el 1/1/1970
document.write("Milisegundos desde el 1/1/1970 hasta ahora " +
miFecha.getTime());
</SCRIPT>
```

/*CÓDIGO RESUMEN DE METODOS DE LECTURA*/

```
<HTML>
<SCRIPT>
var miFecha=new Date()
document.write("La fecha de hoy es " + miFecha + "</br>")
document.write("Dia de la semana " + miFecha.getDay() + "</br>"); // Dia de la semana
//Para ser correctos
var tbdays=new Array
("Domingo","Lunes","Martes","Miercoles","Jueves","Viernes","Sabado")
document.write("Dia de la semana " + tbdays[miFecha.getDay()] + "</br>"); // Dia de la
semana
document.write("Dia del mes " + miFecha.getDate() + "</br>"); //Dia del mes
document.write("Mes " + miFecha.getMonth()+ "</br>"); //Mes
//Para ser correctos
document.write("Mes correcto " + (miFecha.getMonth()+1)+ "</br>"); //Mes
var tbmonths=new Array
("Enero","Febrero","Marzo","Abril","Mayo","Junio","Julio","Agosto","Septiembre","Octub
re","Noviembre","Diciembre")
document.write("Mes correcto " + tbmonths[(miFecha.getMonth())]+ "</br>"); //Mes
document.write("Año " + miFecha.getYear() + "</br>"); //Año
//Para ser correctos
document.write("Año " + miFecha.getFullYear() + "</br>"); //Año

var h=miFecha.getHours();
var m=miFecha.getMinutes();
var s=miFecha.getSeconds();
var ms=miFecha.getMilliseconds();
document.write("Hora" +h + ":" + m + ":" + s + ":" + ms + "</br>"); //hora
//Para ser correctos
if (h<10) {h="0"+ h}
if (m<10) {m="0"+ m}
if (s<10) {s="0"+ s}
document.write("Hora" +h + ":" + m + ":" + s + ":" + ms + "</br>"); //hora

document.write("Milisegundos desde 1970 - " + miFecha.getTime() + "</br></br>")
document.write("Minutos de diferencia entre hora local y la zona horaria de referencia
- " + miFecha.getTimezoneOffset()+ "</br>")
document.write("Dia de la semana UTC " + miFecha.getUTCDay() + "</br>"); // Dia de la
semana
document.write("Dia del mes UTC " + miFecha.getUTCDate() + "</br>"); //Dia del mes
document.write("Mes UTC " + (miFecha.getUTCMonth() +1)+ "</br>"); //Mes
document.write("Año UTC " + miFecha.getUTCFullYear() + "</br>"); //Año
document.write("Hora UTC" + miFecha.getUTCHours() + ":" + miFecha.getUTCMinutes() +
":" + miFecha.getUTCSeconds() + ":" + miFecha.getUTCMilliseconds() + "</br>"); //hora
</SCRIPT>
</HTML>
```

METODOS DE ESCRITURA (SET...)

- **setTime (milisegundos):** Este método crea una fecha añadiendo o restando los milisegundos indicados a la fecha base
- **setMilliseconds (mseg), setSeconds (seg, mseg), setMinutes(min, seg, mseg), setHours(hora,min, seg, mseg):** Fija los campos correspondientes a los milisegundos (mseg), segundos (seg), minutos (min) y hora. En todos estos métodos el primer parámetro es obligatorio y el resto son opcionales.
- **setDate(día), setMonth(mes, día), ~~setYear(año)~~, setFullYear(año, mes, día) :** Fija los campos que corresponden al día del mes, el mes y el año. Al igual que los métodos anteriores, sólo es obligatorio el primer parámetro.
- **setUTCMilliseconds(mseg), setUTCSeconds(seg, mseg), setUTCMinutes(min, seg, mseg), setUTCHours(hora, min, seg, mseg),setUTCDate día, setUTCMonth(mes, día), setUTCFullYear(año, mes, día):** Realizan lo mismo que sus equivalentes pero ajustadas al UTC.

Probando setDate()

```
<SCRIPT language=javascript>
var miFecha =new Date();
document.write("Fecha      antes      de      la      modificación:      "+
miFecha.getDate()+"/"+(miFecha.getMonth()+1)+"/"+miFecha.getFullYear()+"<br>")
miFecha.setDate(8) //Asigna el numero del dia de la fecha
document.write("La fecha después de la modificación : ");
document.write(miFecha.getDate()+"/"+(miFecha.getMonth()+1)+"/"+miFecha.ge
tFullYear());
</SCRIPT>
```

Probando setTime()

```
<SCRIPT language=javascript>
var miFecha=new Date();
// cantidad de milisegundos por día
var msDia = 86400000;
document.write("Hoy es :")
document.write(miFecha.getDate()+"/"+(miFecha.getMonth()+1)+"/"+miFecha.ge
tFullYear()+"<br>")
// sumamos 90 días a la fecha actual
```

```
//Obtenemos los milisegundos desde 1970 y le sumamos los milisegundos de 90 días
//Esto lo asignamos a miFecha
miFecha.setTime(miFecha.getTime() + 90 * msDia);
// Cambia completamente la fecha
document.write("Dentro de 90 días será :")
document.write(miFecha.getDate()+"/"+(miFecha.getMonth()+1)+"/"+miFecha.getFullYear());
</SCRIPT>
```

[Probando setFullYear\(\), setDate\(\), setMonth\(\)](#)

```
<SCRIPT language=javascript>
var miFecha=new Date();
document.write(miFecha.getDate()+"/"+(miFecha.getMonth()+1)+"/"+miFecha.getFullYear()+"<br>")

// Asignación de la fecha
miFecha.setFullYear(2011);
miFecha.setDate(8);
miFecha.setMonth(6);

document.write("La nueva fecha es :");
document.write(miFecha.getDate()+"/"+(miFecha.getMonth()+1)+"/"+miFecha.getFullYear());
</SCRIPT>
```

[Establecer una fecha, a partir de otra](#)

```
var myDate=new Date();
myDate.setDate(myDate.getDate()+5);
```

[Comparar dos fechas](#)

```
var x=new Date();
x.setFullYear(2100,0,14);
var today = new Date();

if (x>today) {
    alert("Today is before 14th January 2100");
}
else{
    alert("Today is after 14th January 2100");
}
```

MÉTODOS DE TRANSFORMACIÓN

- **parse (fecha_cadena) :** Recibe una fecha expresada como una cadena en inglés y devuelve el número de milisegundos que han pasado desde la fecha base. Si no es posible interpretar la cadena como una fecha se devuelve NaN. Este método sólo puede ser llamado desde el objeto Date y no desde sus instancias.
- **UTC (año, mes, día, hora, minuto, segundo, milisegundo):** Toma la fecha indicada y halla el número de milisegundos que han pasado desde la fecha base de acuerdo al UTC. Los tres primeros parámetros son obligatorios para usar este método (año, mes, día) y el resto opcionales. También es un método exclusivo del objeto Date.
- **toDateString(), toLocaleDateString():** Devuelven la fecha sin información sobre la hora, y en forma de texto. Además, toLocaleDateString ajusta los nombres de los campos a la configuración de idioma de la máquina local sobre la que se ejecuta el código. ·
- **toTimeString () , toLocaleTimeString () :** Similar a los anteriores pero devuelven únicamente la parte a la hora
- **toGMTString(),toUTCString(),toLocaleString():** Expresan la fecha y la hora en base al UTC y configuración local respectivamente.
- **toString():** Convierte un objeto Date a string.
- **toISOString():** Devuelve una fecha como un string utilizando el estándar ISO.
- **toJSON():** devuelve una fecha como un string formateado a JSON date.

```
<html>
<script>
document.write("parse   " + Date.parse("April 20,2012") + "<br>")

var later = new Date("August 13,1977");

document.write("to-DateString   " + later.toString() + "<br>")
document.write("to-localDateString   " + later.toLocaleDateString() +
"<br>")

document.write("toTimeString   " + later.toString() + "<br>")
document.write("toLocaleTimeString   " + later.toLocaleTimeString() +
"<br>")

document.write("toGMTString   " + later.toGMTString() + "<br>")
document.write("toUTCString   " + later.toUTCString() + "<br>")
document.write("toString   " + later.toString() + "<br>")
document.write("toLocaleString   " + later.toLocaleString() + "<br>")
document.write("toISOString   " + later.toISOString() + "<br>")
</script>
</html>
```