

# Clase Date

Sobre la clase Date recae todo el trabajo con fechas y horas en Javascript, como obtener una fecha, el día la hora actuales y otras cosas. Para trabajar con fechas necesitamos instanciar un objeto de la clase Date y con él ya podemos realizar las operaciones que necesitemos.

Un objeto de la clase Date se puede crear de dos maneras distintas. Por un lado podemos crear el objeto con el día y hora actuales y por otro podemos crearlo con un día y hora distintos a los actuales. Esto depende de los parámetros que pasemos al construir los objetos.

Para crear un objeto fecha con el día y hora actuales colocamos los paréntesis vacíos al llamar al constructor de la clase Date.

```
let miFecha = new Date()
```

Para crear un objeto fecha con un día y hora distintos de los actuales tenemos que indicar entre paréntesis el momento con que inicializar el objeto. Hay varias maneras de expresar un día y hora válidas, por eso podemos construir una fecha guiándonos por varios esquemas. Estos son dos de ellos, suficientes para crear todo tipo de fechas y horas.

```
let miFecha = new  
Date(año,mes,dia,hora,minutos,segundos)  
let miFecha2 = new Date(año,mes,dia)
```

Los valores que debe recibir el constructor son siempre numéricos. Un detalle: el mes comienza por 0, es decir, enero

es el mes 0. Si no indicamos la hora, el objeto fecha se crea con hora 00:00:00.

Los objetos de la clase Date no tienen propiedades pero si un montón de métodos, vamos a verlos ahora.

`getDate()`

Devuelve el día del mes, un valor que puede ir desde 1 a 31. Depende del mes que sea, claro.

`getDay()`

Devuelve el día de la semana. Por ejemplo, si la fecha corresponde con el martes te devolvería 2.

`getHours()`

Retorna el valor de la hora.

`getMinutes()`

Devuelve los minutos.

`getMonth()`

Devuelve el mes, comenzando por cero (atención a este valor porque el mes de enero correspondería con el valor 0, o diciembre con el 11).

`getSeconds()`

Devuelve los segundos.

`getTime()`

Devuelve los milisegundos transcurridos entre el día 1 de enero de 1970 y la fecha correspondiente al objeto al que se le pasa el mensaje.

`getFullYear()`

Retorna el año, al que se le ha restado 1900. Por ejemplo, para el 1995 retorna 95, para el 2005 retorna 105. Este método está obsoleto en Netscape a partir de la versión 1.3 de Javascript y ahora se utiliza `getFullYear()`.

`getFullYear()`

Retorna el año con todos los dígitos. Usar este método para estar seguros de que funcionará todo bien en fechas posteriores al año 2000.

`setDate()`

Actualiza el día del mes.

`setHours()`

Actualiza la hora.

`setMinutes()`

Cambia los minutos.

`setMonth()`

Cambia el mes (atención al mes que empieza por 0).

`setSeconds()`

Cambia los segundos.

`setTime()`

Actualiza la fecha completa. Recibe un número de milisegundos desde el 1 de enero de 1970.

`setYear()`

Cambia el año recibe un número, al que le suma 1900 antes de colocarlo como año de la fecha. Por ejemplo, si recibe 95 colocará el año 1995. Este método está obsoleto a partir de

Javascript 1.3 en Netscape. Ahora se utiliza `setFullYear()`, indicando el año con todos los dígitos.

`setFullYear()`

Cambia el año de la fecha al número que recibe por parámetro. El número se indica completo ej: 2005 o 1995. Utilizar este método para estar seguros que todo funciona para fechas posteriores a 2000.

Como habréis podido apreciar, hay algún método obsoleto por cuestiones relativas al año 2000: `setYear()` y `getYear()`. Estos métodos se comportan bien en Internet Explorer y no nos dará ningún problema el utilizarlos. Sin embargo, no funcionarán correctamente en Netscape, por lo que es interesante que utilicemos en su lugar los métodos `getFullYear()` y `setFullYear()`, que funcionan bien en Netscape e Internet Explorer.

## Ejemplos de trabajo con fechas en Javascript

Visto lo anterior, ahora podemos poner los conocimientos en la práctica en un ejemplo completo de trabajo con objetos de la clase `Date`.

En este ejemplo vamos a crear dos fechas, una con el instante actual y otra con fecha del pasado. Luego las imprimiremos las dos y extraeremos su año para imprimirlo también. Luego actualizaremos el año de una de las fechas y la volveremos a escribir con un formato más legible.

```
//en estas líneas creamos las fechas
let miFechaActual = new Date()
let miFechaPasada = new Date(1998,4,23)
```

```
//en estas líneas imprimimos las fechas.  
document.write (miFechaActual)  
document.write ("<br>")  
document.write (miFechaPasada)  
  
//extraemos el año de las dos fechas  
let anoActual = miFechaActual.getFullYear()  
let anoPasado = miFechaPasada.getFullYear()  
  
//Escribimos en año en la página  
document.write("<br>El año actual es: " +  
anoActual)  
document.write("<br>El año pasado es: " +  
anoPasado)  
  
//cambiamos el año en la fecha actual  
miFechaActual.setFullYear(2005)  
  
//extraemos el día mes y año  
let dia = miFechaActual.getDate()  
let mes = parseInt(miFechaActual.getMonth()) + 1  
let ano = miFechaActual.getFullYear()  
  
//escribimos la fecha en un formato legible  
document.write ("<br>")  
document.write (dia + "/" + mes + "/" + ano)
```

Hay que destacar un detalle antes de acabar y es que el número del mes puede empieza desde 0. Por esta razón sumamos uno al mes que devuelve el método `getMonth`.