

# Objeto Date

- El objeto `Date` se utiliza para trabajar con fechas y horas. Permite realizar **controles relacionados con el tiempo en las aplicaciones web**.
- Los objetos `Date` se crean con `new Date()` .  
**Por defecto JavaScript crea un objeto `Date` con la hora local actual del sistema.**
- Las fechas en JavaScript **se miden como el n.º de milisegundos transcurridos desde la época UNIX** (1 de enero de 1970 00:00:00 UTC)
- **El tiempo universal coordinado o UTC** es el principal referencia de tiempo por el cual el mundo regula los relojes y el tiempo. Es muy similar a la hora Greenwich. El tiempo local será el tiempo UTC con un desfase + o – de múltiplos de 1 hora dependiendo del huso horario local.
- **Standard ISO 8601**, es un standard que indica una forma de cómo expresar una fecha y una hora:  
Siempre tiene 24 o 27 caracteres de longitud y se refiere a hora UTC (Z):  
`YYYY-MM-DDTHH:mm:ss.sssZ` o  
`±YYYYYY-MM-DDTHH:mm:ss.sssZ` // para indicar años fuera del rango 0000 y 9999 y antes y después del año 0
- El año debe expresarse con 4 dígitos. El mes entre 0 y 11, y el día entre 1 y 31.

# Objeto Date

- Hay 4 formas de instanciar (crear un objeto de tipo Date):
- **var d = new Date ();**  
//Por defecto crea un objeto Date tomando la hora local actual del sistema.  
//esta hora no se actualiza en con el transcurso del tiempo
- **var d= new Date (milisegundosDesdeEpocaUNIX) ;**  
var dHoy=new Date()  
var dAyer=new Date(dHoy.getTime()-24\*60\*60\*1000)  
//dHoy.getTime()→ devuelve la fecha en msg desde 1/1/1970.
- **var d = new Date (cadena\_de\_Fecha\_ISO) ;**  
//Existen otros formatos no ISO, pero no son standard por lo que los resultados dependen de cada navegador. **¡¡¡UTILIZAD formato ISO!!!**  
"YYYY", "YYYY-MM", "YYYY-MM-DD",  
"YYYY-MM-DDTHH:MM:SSZ" //Fecha UTC  
"YYYY-MM-DDTHH:MM:SS+/-HH:00" //Fecha con timezone
- **var d = new Date (año, mes, día, h, min, seg, msg) ;**  
// año con 4 dígitos, mes comienza en 0, Enero es 0, Febrero 1, etc.  
// Los parámetros son tomados como de una fecha local  
// También es válido: new Date(año,mes)new Date(año,mes,día)  
// new Date(año, mes, día, hora)new Date(año,mes,día, hora, min)  
// new Date(año,mes,día,hora,min,seg)

# Objeto Date

- Cuenta con una serie de métodos divididos en tres subconjuntos:
  - **Métodos de lectura.** Empiezan por el prefijo `get`.  
[https://www.w3schools.com/js/js\\_date\\_methods.asp](https://www.w3schools.com/js/js_date_methods.asp)

```
var d=new Date()
```

```
d.getDate()           // d.getUTCDate()
d.getDay()             // d.getUTCDay()
d.getFullYear()        // d.getUTCFullYear()
d.getHours()           // d.getUTCHours()
d.getMilliseconds()    // d.getUTCMilliseconds()
d.getMinutes()         // d.getUTCMinutes()
d.getMonth()           // d.getUTCMonth()
d.getTime()            d.getTimezoneOffset()
```

`d.getYear()` **.-> deprecated** (devuelve la diferencia entre `getFullYear` y 1900)

# Objeto Date

- Cuenta con una serie de métodos divididos en tres subconjuntos:

- **Métodos de escritura.** Empiezan por el prefijo `set`.

[https://www.w3schools.com/js/js\\_date\\_methods\\_set.asp](https://www.w3schools.com/js/js_date_methods_set.asp)

```
var d=new Date()
```

```
d.setDate()           // d.setUTCDate()
d.setFullYear()        // d.setUTCFullYear()
d.setHours()           // d.setUTCHours()
d.setMilliseconds()    // d.setUTCMilliseconds()
d.setMinutes()         // d.setUTCMinutes()
d.setMonth()           // d.setUTCMonth()
```

```
d.setTime()
```

```
d.setYear()-> deprecated
```

# Objeto Date

- Métodos para mostrar la fecha y hora

```
var d=new Date()

d.toString()           // Thu Oct 13 2022
d.toDateString()       // Thu Oct 13 2022
d.toGMTString()         // Thu, 13 Oct 2022 10:59:11 GMT
d.toUTCString()         // Thu, 13 Oct 2022 10:59:11 GMT
d.toISOString()         // 2022-10-13T10:59:11.875Z
d.toLocaleString()      // 13/10/2022, 12:59:11
d.toLocaleDateString()  // 13/10/2022
d.toLocaleTimeString()  // 12:59:11
d.toString()            // Thu Oct 13 2022 12:59:11 GMT+0200 (hora de verano de Europa central)
```

# Objeto Date

Métodos del objeto Date	
Método	Descripción
<code>getDate()</code>	Devuelve el día del mes (de 1-31).
<code>getDay()</code>	Devuelve el día de la semana (de 0-6).
<code>getFullYear()</code>	Devuelve el año (4 dígitos).
<code>getHours()</code>	Devuelve la hora (de 0-23).
<code>getMilliseconds()</code>	Devuelve los milisegundos (de 0-999).
<code>getMinutes()</code>	Devuelve los minutos (de 0-59).
<code>getMonth()</code>	Devuelve el mes (de 0-11).
<code>getSeconds()</code>	Devuelve los segundos (de 0-59).
<code>getTime()</code>	Devuelve los milisegundos desde media noche del 1 de Enero de 1970.
<code>getTimezoneOffset()</code>	Devuelve la diferencia de tiempo entre GMT y la hora local, en minutos.
<code>getUTCDate()</code>	Devuelve el día del mes en base a la hora UTC (de 1-31).
<code>getUTCDay()</code>	Devuelve el día de la semana en base a la hora UTC (de 0-6).
<code>getUTCFullYear()</code>	Devuelve el año en base a la hora UTC (4 dígitos).
<code>setDate()</code>	Ajusta el día del mes del objeto (de 1-31).
<code>setFullYear()</code>	Ajusta el año del objeto (4 dígitos).
<code>setHours()</code>	Ajusta la hora del objeto (de 0-23).

# Ejemplos de uso: Objeto Date

- - CUIDADO - - : cambio de fecha y hora

- Cuando establecemos un día, hora, año, mes, etc lo haremos utilizando una expresión (la local o la UTC), y según esa expresión solo se cambia el número en cuestión implicado y no afecta al resto de datos de fecha y hora. Si cambiamos la fecha, cambiará solo la fecha, el resto de datos permanecerán igual, si cambiamos las horas cambiarán solo las horas el resto.
- Ahora bien, si al realizar ese cambio estamos implícitamente pasando de un huso horario a otro, como puede ocurrir en España al tener 2 husos horarios (el horario de verano y el horario de invierno), el método no tendrá en cuenta el posible cambio de huso horario por lo que se además cambiará la hora indirectamente, y posiblemente otros datos.

```
d
► Date Sat Oct 28 2023 12:20:47 GMT+0200 (hora de verano de Europa central)

d.setMonth(10)
1701170447366

d
► Date Tue Nov 28 2023 12:20:47 GMT+0100 (hora estándar de Europa central)
```

- Al cambiar el mes al de noviembre estamos pasando a otro huso horario, pero esto no lo refleja el método `setMonth()` en la hora, por lo que en realidad estaremos cambiando también la hora. Ya que:
  - Inicialmente d es: 10:20 del 28 de octubre según UTC.
  - Después d es: 11:20 del 28 de noviembre según UTC.

# Objeto Date

## Ejemplos de uso:

- - CUIDADO - - : cambio de fecha y hora

- Si hubiésemos utilizado los métodos `setUTC*` esto no habría ocurrido, ya que no habría habido cambio de huso horario entre la variable original y la resultante tras la modificación:

```
d
► Date Sat Oct 28 2023 12:20:47 GMT+0200 (hora de verano de Europa central)
d.setUTCMonth(10)
1701166847366
d
► Date Tue Nov 28 2023 11:20:47 GMT+0100 (hora estándar de Europa central)
```

- `d` originalmente es 28 de octubre a las 10:20 según UTC y después del cambio es 28 de noviembre a las 10:20 según UTC.



# Ejemplos de uso: Objeto Date

- CUIDADO: cuando incrementamos o decrementamos un número de horas, minutos, días, meses etc, ya que puede haber un cambio de huso horario.

```
d=new Date(2023,9,29,16,30)
```

```
► Date Sun Oct 29 2023 16:30:00 GMT+0100 (hora estándar de Europa central)
```

```
d.setHours(d.getHours()-24)
```

```
1698503400000
```

```
d
```

```
► Date Sat Oct 28 2023 16:30:00 GMT+0200 (hora de verano de Europa central)
```

Observa que se pretende descontar 24 horas pero realmente se han descontado 25 horas. Por lo que, en este caso y para ser estrictos mejor utilizar los métodos UTC, ya que no ocurre esto porque no hay cambios de husos en el mismo año:

```
>> d=new Date(2023,9,29,16,30)
```

```
← ► Date Sun Oct 29 2023 16:30:00 GMT+0100 (hora estándar de Europa central)
```

```
>> d.setUTCHours(d.getUTCHours()-24)
```

```
← 1698507000000
```

```
>> d
```

```
← ► Date Sat Oct 28 2023 17:30:00 GMT+0200 (hora de verano de Europa central)
```