

Cliente

Nombre y apellidos: Evelia Gil Paredes

Curso: 2025/2026



El primer paso es crear el html en el que saldrá la información. Este html ha de estar enlazado con el «script», que se encargará de toda la parte lógica de la página web.

```
crómetro > index.html > ...
1  <!DOCTYPE html>
2  <html lang="en">
3    <head>
4      <meta charset="UTF-8" />
5      <meta name="viewport" content="width=device-width, initial-scale=1.0" />
6      <title>Cronómetro</title>
7      <script src="script.js" defer></script>
8      <link rel="stylesheet" href="styles.css" />
9    </head>
10   <body></body>
11 </html>
```

A continuación, obtengo el «cuerpo» (*body*) del html, para poder añadir luego todos los elementos.

Añado un «main» con una «section» y un «input» de tipo date, para poder escoger cualquier otra fecha en el sitio web.

El inputDate tendrá un *id*, el tipo «date» y un valor máximo a seleccionar (ya que no quiero calcular el tiempo posterior a esa fecha).

También creo un título y un subtítulo.

En el elemento «bodyElement» añado el «mainElement» y al «mainElement» añado el «sectionElement».

```
crómetro > JS script.js > ...
1  const bodyElement = document.body;
2
3  const mainElement = document.createElement("main");
4  bodyElement.appendChild(mainElement);
5
6  const sectionElement = document.createElement("section");
7  mainElement.appendChild(sectionElement);
8
9  const inputDate = document.createElement("input");
10 inputDate.setAttribute("id", "inputDate");
11 inputDate.setAttribute("type", "date");
12 inputDate.setAttribute("max", "2025-12-14");
13
14 const title = document.createElement("h1");
15 title.textContent = "Cronómetro";
16
17 const subtitle = document.createElement("h2");
18 subtitle.textContent = `Tiempo desde hoy hasta mi cumpleaños`;
19
```

Cronómetro

Tiempo desde hoy hasta mi cumpleaños

Después, creo una tabla y dos arrays, por un lado, el array de los títulos de la tabla y, por otro lado, un array con los id de las columnas que utilizaré después para mostrar los valores del cronómetro.

A continuación, creo dos filas y a cada fila le otorgo columnas.

Si $i = 0 \rightarrow$ es la primera fila, creo 5 columnas *th* (encabezados) y a cada columna le doy un nombre, obtenido del array de «tableTitle».

Si no es la primera fila, le añado 5 columnas y cada una tendrá un *id*, escogido del array «tableElements».

A cada fila le añado las columnas y a la tabla le añado las filas.

Luego creo un elemento h3 que usaré después.

```
const table = document.createElement("table");
const tableTitles = ["Meses", "Días", "Horas", "Minutos", "Segundos"];
const tableElements = ["months", "days", "hours", "minutes", "seconds"];

for (let i = 0; i < 2; i++) {
  const file = document.createElement("tr");
  if (i == 0) {
    for (let j = 0; j < 5; j++) {
      const column = document.createElement("th");
      column.textContent = `${tableTitles[j]}`;
      file.appendChild(column);
    }
  } else {
    for (let j = 0; j < 5; j++) {
      const column = document.createElement("td");
      column.setAttribute("id", `${tableElements[j]}Column`);
      column.textContent = "0";
      file.appendChild(column);
    }
  }
  table.appendChild(file);
}

const daysLeft = document.createElement("h3");
```

Tras crear los elementos ya mencionados, creo otro h3 en el que saldrá la fecha límite.

Esta fecha límite (mi cumpleaños) se obtendrá a partir de la variable «birthdayDate», una variable de tipo `date` cuyo valor es mi cumpleaños.

Y en este elemento `h3`, aparecerá la fecha de mi cumpleaños pasada a *string* con la función «`toLocaleDateString`».

Finalmente, en cuanto a elementos, añado a la sección todos los títulos, subtítulos, la tabla, el `inputDate` y cualquier otro elemento `html` creado con anterioridad.

```
const birthdayDate = new Date("2025-12-13");
birthdayDate.setHours(0, 0, 0);
const limitDate = document.createElement("h3");
limitDate.textContent = `${birthdayDate.toLocaleDateString()}`;
sectionElement.appendChild(limitDate);

sectionElement.appendChild(title);
sectionElement.appendChild(subtitle);
sectionElement.appendChild(limitDate);
sectionElement.appendChild(table);
sectionElement.appendChild(daysLeft);
sectionElement.appendChild(inputDate);
```

Llegados a este punto del programa, creo la fecha actual a la que llamo «`dateNow`», una variable de tipo fecha (*date*).

Y la variable copia de la fecha actual «`currentDate`», que servirá para modificarla.

```
57
58 let dateNow = new Date();
59 let currentDate = new Date();
```

Uso los valores de la variable «`dateNow`» para establecer la fecha mínima a seleccionar en el «`inputDate`», para no poder seleccionar una fecha más allá de la fecha actual.

Para ello, guardo en variables el valor de los días, meses y años en «`minDay`», «`minMonth`», «`minYear`» y todos ellos los guardo en una sola variable llamada «`minDate`», que se utiliza para establecer el atributo «`min`» del `inputDate`.

El «`minDay`» y el «`minMonth`» se deben convertir a `string` y añadirles un 0 siempre que sean resultados de una sola cifra con la función «`padStart(número de valores, elemento con el que se rellena el número de valores hasta tener el establecido en el parámetro anterior)`». Además, al mes le tengo que añadir 1 ya que los meses empiezan desde el 0 hasta el 11, pero en el `input` se establecen los años desde el 1 hasta el 12.

```

61 let minDay = String(dateNow.getDate()).padStart(2, "0");
62 let minMonth = String(dateNow.getMonth() + 1).padStart(2, "0");
63 let minYear = dateNow.getFullYear();
64 let minDate = `${minYear}-${minMonth}-${minDay}`;
65 inputDate.setAttribute("min", minDate);
66
67 const timeToChange = 1000;
68 counter();
69

```

Ahora explicaré la función encargada de la diferencia entre las fechas.

NOTA: siempre se debe poner la fecha superior – la fecha inferior, en este caso con las variables del min y del max del «inputDate» se cumple, sino tendríamos que intercambiar los valores.

Elementos:

Meses: (años de la fecha de mi cumpleaños – años actuales) * 12 meses + meses de mi cumpleaños – meses actuales. Función que sirve para obtener los meses sin tener que calcular los años (los cuales no me hacen falta en esta práctica).

Días: días de la fecha de mi cumpleaños – días actuales.

Horas: horas de la fecha de mi cumpleaños – horas actuales.

Minutos: minutos de la fecha de mi cumpleaños – minutos actuales.

Segundos: segundos de la fecha de mi cumpleaños – segundos actuales.

Luego, se necesitan realizar los ajustes para la fecha negativa, para ello se parte de los segundos hasta los meses.

Ajuste de segundos, minutos, horas. Si el resultado de la resta anterior es menor que 0, es decir, negativo, se resta 1 al valor superior (si es segundos, se resta a los minutos) y se toma ese tiempo prestado sumándoselo a la variable que se está ajustando.

Ej: si sale que son 5 segundos hasta mi cumpleaños y estamos en el segundo 30, el resultado de la resta es de -25 segundos.

Desde los 30 segundos actuales hasta el minuto son 30 segundos + los 5 segundos que se necesitan hasta los segundos de mi cumpleaños, eso hacen un total de 35 segundos.

Como el ajuste se resta un minuto, ya que no es un minuto completo lo que se necesita y se le suman 60 segundos (del minuto eliminado) a los actuales. Resultado: 60 – 25 = 35 minutos.

Esa misma lógica se sigue con los minutos y las horas.

```

function getDifferences(){
  let month =
    (birthdayDate.getFullYear() - currentDate.getFullYear()) * 12 +
    (birthdayDate.getMonth() - currentDate.getMonth());
  let day = birthdayDate.getDate() - currentDate.getDate();
  let hour = birthdayDate.getHours() - currentDate.getHours();
  let minute = birthdayDate.getMinutes() - currentDate.getMinutes();
  let second = birthdayDate.getSeconds() - currentDate.getSeconds();

  if (second < 0){
    minute--;
    second += 60;
  }
  if (minute < 0){
    hour--;
    minute += 60;
  }
  if (hour < 0){
    day--;
    hour += 24;
  }
}

```

En cuanto al ajuste de días, se sigue la lógica: si el resultado es negativo, hay que tomar prestado un mes. sin embargo, se debe tener en cuenta que según el mes, hay unos días u otros.

Para tener en cuenta este hecho, se utiliza la variable «daysInPreviousMonth» (días del mes anterior), que es un variable de tipo fecha, en la que se establece el año del cumpleaños, el mes del cumpleaños y para obtener el número de días del mes, se usa el 0.

A la variable días («days») se le suman los días del mes anterior.

Finalmente, se retorna un objeto json con todos los valores: meses, días, horas, minutos y segundos.

```

    if (day < 0) {
        month--;
        const daysInPreviousMonth = new Date(
            birthdayDate.getFullYear(),
            birthdayDate.getMonth(),
            0
        ).getDate();
        day += daysInPreviousMonth;
    }
    if (month < 0) {
        month += 12;
    }
    return { month, day, hour, minute, second };
}

```

Se crea un intervalo, una función que llama a la función contador («counter») cada segundo (1000 milisegundos).

```

const intervalDate = setInterval(() => {
    counter();
}, timeToChange);

```

En cuanto al inputDate, le añado un evento «change», que detecta el cambio del input.

Obtengo el valor el input con e.target.value (valor del elemento emisor del evento), que se obtiene en string. Cambio la variable «currentDate» por la nueva fecha obtenida del valor del input y como siempre que se hace las horas, los minutos, los segundos y los milisegundos se ponen a 0, consigo esos valores de la fecha actual con la variable «now» y se los doy a «currentDate».

Luego llamo a la función «counter» para que se recalcule el cronómetro.

```

inputDate.addEventListener("change", (e) => {
  let dateString = e.target.value;
  currentDate = new Date(dateString);
  const now = new Date();
  currentDate.setHours(
    now.getHours(),
    now.getMinutes(),
    now.getSeconds(),
    now.getMilliseconds()
  );

  counter();
});

```

Creo una variable «timeToChange» que sirve para ir actualizando la fecha, variable que usaré más tarde.

Y llamo a la función «counter» para que desde la apertura de la página web obtengamos un resultado.

```

7  const timeToChange = 1000;
8  counter();

```


Función «counter».

Creo las variables «counters» para los meses, días, horas, minutos y segundos con valor a 0.

Luego limpio el contenido de la variable «daysLeft».

Obtengo todas las columnas de los meses, días, horas, minutos y segundos.

Si la fecha actual es menor que la fecha de mi cumpleaños:

Obtengo la diferencia entre fechas → objeto json de la función «getDifferences».

Luego actualizo los «counters» con sus valores, respectivamente, accediendo a la propiedad del objeto «differencesBetweenDates».

Si la diferencia es menor o igual que 0, limpio el intervalo creado anteriormente.

```
function counter() {  
  let counterMonths = 0;  
  let counterDays = 0;  
  let counterHours = 0;  
  let counterMinutes = 0;  
  let counterSeconds = 0;  
  daysLeft.textContent = ``;  
  
  const monthColumn = document.getElementById("monthsColumn");  
  const daysColumn = document.getElementById("daysColumn");  
  const hoursColumn = document.getElementById("hoursColumn");  
  const minutesColumn = document.getElementById("minutesColumn");  
  const secondsColumn = document.getElementById("secondsColumn");  
  
  if (currentDate <= birthdayDate) {  
    differenceBetweenDates = getDifferences();  
    counterMonths = differenceBetweenDates.month;  
    counterDays = differenceBetweenDates.day;  
    counterHours = differenceBetweenDates.hour;  
    counterMinutes = differenceBetweenDates.minute;  
    counterSeconds = differenceBetweenDates.second;  
    differenceDate = birthdayDate - currentDate;  
  
    if (differenceDate <= 0) {  
      clearInterval(intervalDate);  
    }  
  }  
}
```

En la segunda parte de la función, establezco los colores.

Si hay un mes o más, el color de la tabla es verde.

Si hay menos o igual a 0 meses y menos de 7 días, la tabla es rojo.

Si no, por defecto, es naranja.

Luego voy añadiendo los 1000 milisegundos a la fecha actual.

```
... if (counterMonths >= 1) {  
...   table.style.color = "#16a34a";  
... } else if (counterMonths <= 0 && counterDays < 7) {  
...   table.style.color = "#ef4444";  
... } else {  
...   table.style.color = "#dd8512ff";  
... }  
... currentDate = new Date(currentDate.getTime() + timeToChange);  
... }
```

Si la fecha cumpleaños es menor que la actual, por ejemplo, si selecciono el día 13 de diciembre de 2025, la tabla es roja y sale un mensaje de que no se puede.

```
... } else {  
...   table.style.color = "#ef4444";  
...   daysLeft.style.color = "#ef4444";  
...   daysLeft.textContent = `La fecha actual no puede ser mayor que la fecha máxima`;  
... }
```

Finalmente, en las columnas obtenidas anteriormente, escribo el valor de los contadores.

```
... monthColumn.textContent = `${counterMonths}`;  
... daysColumn.textContent = `${counterDays}`;  
... hoursColumn.textContent = `${counterHours}`;  
... minutesColumn.textContent = `${counterMinutes}`;  
... secondsColumn.textContent = `${counterSeconds}`;  
... }
```

Resultado:



Cronómetro

Tiempo desde hoy hasta mi cumpleaños

13/12/2025

Meses	Días	Horas	Minutos	Segundos
0	29	3	45	9

13/11/2025

Cronómetro

Tiempo desde hoy hasta mi cumpleaños

13/12/2025

Meses	Días	Horas	Minutos	Segundos
0	1	3	45	0

11/12/2025