

U2. Objetos

Ejercicios: Objetos [Date](#), [Array](#) y [Métodos Window](#) [Temporizadores](#)

1. Crea un objeto “fecha” empleando el objeto Date. A partir de ese objeto:
 - a. Muestra por consola el día (del 1 al 31)
 - b. Muestra por consola el mes (del 1 al 12)
 - c. Muestra por consola el año
 - d. Muestra el día de la semana (Lunes, Martes, ...)
 - e. Muestra la fecha completa con formato dd/mm/aaaa
 - f. Muestra las horas (de 0 a 23)
 - g. Muestra la fecha completa con formato dd/mm/aaaa HH:MM:SS
2. Crea un script que:
 - a. Solicite al usuario la fecha de las próximas vacaciones con un formato string tipo “aaaa-mm-dd” o “aaaa/mm/dd”.
 - b. Convierte la fecha anterior en un objeto Date y calcula cuántos milisegundos quedan hasta esa fecha.
 - c. Calcule cuántos días quedan hasta esa fecha
3. Realiza los pasos que se indican a continuación:
 - a. Crea un array con 4 elementos
 - b. Concatena 2 elementos más al final y 2 al principio
 - c. Muestra por consola el resultado después de aplicar cada uno utilizando toString()
 - d. Elimina las posiciones de la 3 a la 5 (incluidas)
 - e. Inserta el número 10 y el 25 entre el penúltimo y el último
 - f. Muestra el array del paso anterior, pero con los elementos separados por "==" utilizando join().
 - g. Copia el array anterior en otra variable e invierte el orden de los elementos con esa segunda variable.
 - h. Muestra los dos arrays (directamente, sin join() ni toString()). ¿Puedes explicar qué ha pasado con los dos arrays y por qué?
 - i. Consulta en qué índice del array está el 25 y en qué índice está un elemento que no esté en el array.

Método Every y Some

El **método every** devolverá un boolean indicando **si todos los elementos del array cumplen cierta condición**. Esta función recibirá cualquier elemento, lo testeará, y devolverá cierto o falso dependiendo de si cumple la condición o no.

```
var a = [3, 21, 15, 61, 9, 54];
```

```
console.log(a.every(function(num) { // Comprueba si cada nº es menor a 100
```

```
return num < 100; })); // Imprime true
```

```
console.log(a.every(function(num) { // Comprueba si cada nº es par
return num % 2 == 0; })); // Imprime false
```

Por otro lado, el **método some** es similar a every, pero devuelve cierto en el momento en el que **uno de los elementos del array cumple la condición**.

```
var a = [3, 21, 15, 61, 9, 54];
```

```
console.log(a.some(function(num) { // Comprueba si algún elemento del array es par
return num % 2 == 0; })); // Imprime true
```

4. A partir del ejercicio anterior, utilizando every, some y reduce, muestra por consola los resultados que se indican a continuación:

- Si todos los elementos del array son de tipo Number
- Si alguno de los elementos del array es menor de 20
- Utiliza el [método reduce\(\)](#) para obtener la suma del array siguiente:

```
var a = [4, 21, 33, 12, 9, 54];
```

Método Sort

Podemos **ordenar** los elementos de un array usando el método sort.

```
var a = ["Peter", "Anne", "Thomas", "Jen", "Rob", "Alison"];
```

```
a.sort(); // Ordena el array original
```

¿Qué ocurre si intentamos ordenar elementos que no son string?. Por defecto, lo ordenará por su valor como string (teniendo en cuenta que si son objetos, se intentará llamar al método toString() para ordenarlo). Para ello, tendremos que pasar una función (de ordenación), que comparará 2 valores del array y devolverá un valor numérico indicando cual es menor (negativo si el primero es menor, 0 si son iguales y positivo si el primero es mayor).

```
var a = [20, 6, 100, 51, 28, 9];
```

```
a.sort(); // Ordena el array original
```

```
console.log(a); // Imprime [100, 20, 28, 51, 6, 9]
```

```
a.sort(function(n1, n2) {
```

```
    return n1 - n2;
```

```
});

console.log(a); // Imprime [6, 9, 20, 28, 51, 100]

Veamos un ejemplo con objetos, en este caso son personas y las ordenaremos según la edad.

function Persona(nombre, edad) { // Constructor de la clase persona this.nombre = name; this.edad = edad;

this.toString = function() { // Método toString() return this.nombre + " (" + this.edad + ")";

}}

var personas = [];

personas[0] = new Persona("Thomas", 24);

personas[1] = new Persona("Mary", 15);

personas[2] = new Persona("John", 51);

personas[3] = new Persona("Philippa", 9);

personas.sort(function(p1, p2) {

    return p1.edad - p2.edad; // Comparamos la edad

});

console.log(personas.toString()); // Imprime:"Filippa (9),Mary (15),Thomas (24),John (51)"
```

5. Copia el Array de objetos Estudiante que creaste en U2Ejercicios1, 6d y utiliza sort para ordenar a los estudiantes en función de la nota.

Métodos Map y Filter

Para **modificar todos los elementos de un array**, el **método map** recibe una función que transforma cada elemento y lo devuelve. Este método devolverá al final un nuevo array del mismo tamaño conteniendo todos los elementos resultantes.

```
var a = [4, 21, 33, 12, 9, 54];
```

```
console.log(a.map(function(num) {
```

```
return num*2; })); // Imprime [8, 42, 66, 24, 18, 108]
```

Para **filtrar los elementos de un array**, y obtener como resultado un array que contenga sólo los elementos que cumplan cierta condición, usamos el **método filter**.

```
var a = [4, 21, 33, 12, 9, 54];

console.log(a.filter(function(num) {

return num % 2 == 0; // Si devuelve true, el elemento se queda en el array devuelto })); //
Imprime [4, 12, 54]
```

6. Crea un array con diferentes tipos de valores (números, strings, booleanos).
 - a. A partir de dicho array, utilizando map(), genera otro array con dichos valores convertidos todos a número.
 - b. Utiliza filter para no incluir en el array final los elementos que no hayan podido ser convertidos (NaN).
 - c. Imprime dicho array resultante por consola.

Método forEach

Podemos **iterar por los elementos de un array** usando el **método forEach**. De forma opcional, podemos llevar un seguimiento del índice al que está accediendo en cada momento, e incluso recibir el array como tercer parámetro.

```
var a = [3, 21, 15, 61, 9, 54];

a.forEach(function(num, indice, array) { // índice y array son parámetros opcionales

    console.log("Índice " + indice + " en [" + array + "] es " + num);

}); // Imprime -> Índice 0 en [3,21,15,61,9,54] es 3, Índice 1 en [3,21,15,61,9,54] es 21, ...
```

7. Utiliza el método forEach para:
 - a. Multiplicar por 2 cada elemento del array a que has empleado en el ejercicio 4c (var a = [4, 21, 33, 12, 9, 54];). Muestra el array por pantalla.
 - b. Imprimir por pantalla el array resultante del ejercicio 6 (indica el índice y los atributos del objeto correspondiente al elemento en el que estamos).
8. Realiza una web que muestre un botón “Temporizador”. Cuando pulsemos ese botón debe activarse un temporizador que muestre un aviso que diga “Ya han pasado 5 segundos” cuando haya transcurrido ese tiempo.
9. Crea una web que muestre la hora por consola formateada como HH:MM:SS. El refresco de esa información debe ser cada 5 segundos.
10. Incorpora a la web anterior 2 botones:
 - a. El primero debe activar que se muestre la hora por consola.
 - b. El segundo debe parar que se muestre la hora por consola.