

1. Objeto window

En este módulo vamos a aprender a integrar los conocimientos adquiridos hasta ahora: HTML, CSS y JS para crear documentos dinámicos y que reaccionen a eventos generados por los usuarios.

Cuando un documento web se muestra en el navegador este nos proporciona una serie de *interfaces* que representan la ventana del navegador, el documento, los datos guardados y muchas más cosas. Podremos interactuar con esos elementos usando JavaScript.

Lo primero que vamos a aprender es a manejar el objeto window y el DOM.

Window

El objeto **window** representa el tab del navegador que contiene un documento HTML. Nos proporciona una serie de propiedades y métodos que podemos ejecutar para realizar determinadas acciones.

A su vez, **window** es el contexto global de JavaScript que se ejecuta en el navegador por lo que todas sus propiedades y métodos están disponibles en cualquier lado donde se referencien. Esto implica que las propiedades y métodos del objeto **window** pueden ser usados directamente sin referenciar el objeto:

```
// Es lo mismo que "window.console.log('hello world');"
console.log('hello world');
```

Propiedades interesantes:

- **window.innerHeight**: nos muestra la altura del área de contenido de la ventana (en píxeles).
- **window.innerWidth**: nos muestra el ancho del área de contenido de la ventana (en píxeles) incluida la barra de scroll.

- **window.outerHeight:** nos muestra la altura completa de la ventana.
- **window.outerWidth:** nos muestra la anchura completa de la ventana
- **window.location:** nos muestra la URL del documento que contiene la ventana. Podremos modificar esa URL usando estos métodos (<https://developer.mozilla.org/en-US/docs/Web/API/Location>).
- **window.navigator:** proporciona información sobre el navegador que estamos usando. El objeto tiene una serie de propiedades y métodos para acceder tanto a información del navegador como a otra información relacionada, entre ellas: **window.navigator.userAgent** que nos da toda la información sobre el tipo de navegador.
- **window.scrollX** y **window.scrollY:** nos muestra la cantidad de scroll horizontal y vertical respectivamente (en píxeles).
- **window.console:** es una referencia al objeto **Console** que proporciona varios métodos para mostrar contenido en consola de javascript. Aquí (<https://developer.mozilla.org/en-US/docs/Web/API/Console>) están los métodos.

Métodos útiles:

- **window.alert():** permite crear mensajes de diálogo que bloquean el interface de usuario.
- **window.open():** permite abrir un nuevo tab de navegador.
- **window.close():** permite cerrar el tab actual si fue abierta por nosotros con **window.open()**.
- **window.confirm():** permite crear un mensaje de diálogo que se puede aceptar o cancelar. Devuelve true o false.
- **window.prompt():** permite crear un mensaje de diálogo que permite al usuario introducir un dato que devuelve la función en formato texto.
- **window.print():** abre la ventana de impresión que permite mandar a la impresora el documento HTML cargado en la ventana.

- **window.setTimeout()**: permite ejecutar una función después de un tiempo.
- **window.clearTimeout()**: permite cancelar el timeout creado por la anterior.

```
// Muestra la URL del tab en la consola.  
const showURL = () => console.log(window.location.href);  
  
// Ejecuta la funcion anterior después de 1 segundo (1000 milisegundos).  
const delayed = setTimeout(showURL, 1000);  
  
// Cancela la ejecución.  
window.clearTimeout(delayed);
```

- **window.setInterval()**: permite ejecutar una función cada X tiempo.
- **window.clearInterval()**: permite cancelar su ejecución.

```
let contador = 0;  
  
// Muestra un contador en la consola. Cada segundo aumenta la cuenta en 10 y  
se detiene cuando llega a 100.  
const showCount = () => {  
  console.log(contador);  
  if (contador >= 100) {  
    window.clearInterval(interval);  
  } else {  
    contador += 10;  
  }  
};  
  
//Ejecuta función anterior cada segundo.  
const interval = window.setInterval(showCount, 1000);
```

Completado. Continuamos.

