

Objetos predefinidos en JavaScript:

**Window, navigator,
location, screen y history**



Índice

Window	3
Console	4
Ajustando el nivel de verbose	4
console.table	5
Navigator	5
Location	6
Screen	7
History	8

Window

El objeto window es el que sirve para controlar la ventana del navegador.

Es el objeto principal en la jerarquía y contiene las propiedades y métodos para controlar la ventana del navegador. De él dependen todos los demás objetos de la jerarquía.

En un navegador con pestañas, cada pestaña contiene su propio window object. Esto significa que el objeto window no se comparte entre diferentes pestañas de la misma ventana del navegador.

Propiedades:

Window.open(); -> El método open() crea una nueva ventana secundaria del navegador, similar a seleccionar Nueva ventana desde el menú archivo. El parámetro url especifica la URL a ser solicitada y cargada en la nueva ventana. Si url es una cadena de caracteres vacía, entonces se creará una nueva ventana en blanco (URL about:blank).

```
window.open();  
window.open([url]);
```

Window.close(); -> Cuando este método es llamado, la ventana referenciada es cerrada. Este método solo se permite ser llamado por ventanas que fueron abiertas por un script utilizando el método window.open(). Si la ventana no fue abierta por un script, el siguiente error aparece en la consola de JavaScript: Scripts may not close windows that were not opened by script.

```
window.close();
```

Window.alert(); -> Muestra un diálogo de alerta con contenido opcional especificado y un botón OK (Aceptar).

```
window.alert(message);
```

Window.confirm(); -> Muestra una ventana de diálogo con un mensaje opcional y dos botones, Aceptar y Cancelar.

```
window.confirm();
```

Window.getComputedStyle(); -> Devuelve el estilo computado del elemento. Los estilos computados representan los valores finales computed de las propiedades CSS del elemento.

```
var element = document.getElementById("elemId");  
var style = document.defaultView.getComputedStyle(element, pseudoElt);
```

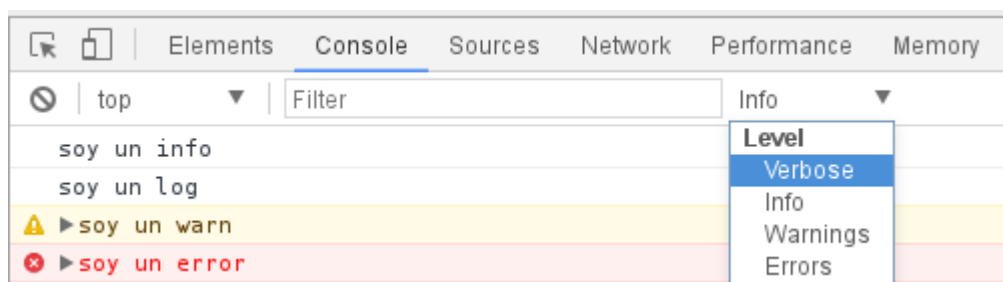
- Element es un element.
- PseudoElt es una cadena que especifica el pseudo elemento a tratar. Debería ser una cadena vacía para la mayoría de los elementos element.
- Style es un objeto del tipo CSSStyleDeclaration.

Console

Ajustando el nivel de verbose

Un ejemplo práctico:

```
console.info("soy un info");  
console.log("soy un log");  
console.warn("soy un warn");  
console.error("soy un error");
```



console.table

Un ejemplo práctico:

```
const pescados =[
{
  Nombre: "Congrio común",
  Orden: "Anguilliformes",
  Suborden: "Congroidei"
},
{
  Nombre: "Merluza europea",
  Orden: "Gadiformes",
  Suborden: "Paracanthopterygii"
},
{
  Nombre: "Txitxarro",
  Orden: "Perciformes",
  Suborden: "Percoidei"
}
];
```

```
> console.table(pescados);
```

VM5553:1

(index)	Nombre	Orden	Suborden
0	"Congrio común"	"Anguilliformes"	"Congroidei"
1	"Merluza europea"	"Gadiformes"	"Paracanthopterygii"
2	"Txitxarro"	"Perciformes"	"Percoidei"

► Array(3)

Navigator

Window.navigator es una referencia al objeto que contiene las propiedades del navegador y los métodos asignados al mismo.

Dentro de navigator podemos destacar userAgent, que contiene una cadena con información referente al navegador.

Un ejemplo de código para averiguar el navegador actual sería este:

```
let sBrowser, sUsrAg = navigator.userAgent;

if(sUsrAg.indexOf("Chrome") > -1) {
    sBrowser = "Google Chrome";
} else if (sUsrAg.indexOf("Safari") > -1) {
    sBrowser = "Apple Safari";
} else if (sUsrAg.indexOf("Opera") > -1) {
    sBrowser = "Opera";
} else if (sUsrAg.indexOf("Firefox") > -1) {
    sBrowser = "Mozilla Firefox";
} else if (sUsrAg.indexOf("MSIE") > -1) {
    sBrowser = "Microsoft Internet Explorer";
}

alert("Usted está utilizando: " + sBrowser);
```

navigator.appName; Siempre devuelve "Netscape". Es así por temas de compatibilidad.

Lo mismo pasa con **navigator.appCodeName** (Siempre devuelve "Mozilla" por compatibilidad) y **navigator.product** (Siempre devuelve "Gecko")

Navigator.connection -> Proporciona un objeto **NetworkInformation** que contiene información sobre la conexión de red de un dispositivo.

Navigator.geolocation -> Devuelve un objeto **Geolocation** que permite acceder a la ubicación del dispositivo

Navigator.online -> Devuelve un valor booleano que indica si el navegador está funcionando en línea.

Location

La propiedad de sólo lectura **Window.location** retorna un objeto **Location** con información acerca de la ubicación actual del documento.

Ejemplos:

location.reload(true); Recarga la página actual

location = "<https://moodle.iesgrancapitan.org/my/>" o **location.href** = "<https://moodle.iesgrancapitan.org/my/>"; Cambia la url actual y recarga

También se puede hacer con:

window.location.assign("https://moodle.iesgrancapitan.org/my/")

location.pathname; Devuelve el path

location.protocol; Devuelve el protocolo

location.href; Obtiene o establece la url completa.

location.origin; Obtiene o establece el protocolo, host y puerto de una url.

Screen

La propiedad Window.screen devuelve una referencia al objeto de pantalla asociado con la ventana. El objeto screen, que implementa la interfaz screen, es un objeto especial para inspeccionar las propiedades de la pantalla en la que se representa la ventana actual.

Propiedades:

- **screen.width** La propiedad screen.width devuelve el ancho de la pantalla del visitante en píxeles.
- **screen.height** La propiedad screen.height devuelve la altura de la pantalla del visitante en píxeles.
- **screen.availWidth** La propiedad screen.availWidth devuelve el ancho de la pantalla del visitante, en píxeles, menos las características de la interfaz como la barra de tareas de Windows.
- **screen.availHeight** La propiedad screen.availHeight devuelve la altura de la pantalla del visitante, en píxeles, menos las características de la interfaz como la barra de tareas de Windows.
- **screen.colorDepth** La propiedad screen.colorDepth devuelve el número de bits utilizados para mostrar un color.
- **screen.pixelDepth** La propiedad screen.pixelDepth devuelve la profundidad de píxeles de la pantalla.

History

Regresa una referencia al objeto History, quien provee una interfaz para manipular el historial de *sesión del navegador*.

Propiedades:

history.back(); Mismo efecto que clicar en volver hacia la página anterior en las flechas de navegación.

history.forward(); Mismo efecto que clicar hacia delante en las flechas de navegación.

current	URL de la entrada actual de HISTORY
length	número de entradas en el historial
next	URL de la entrada siguiente (con respecto a la entrada actual)
previous	URL de la entrada anterior (con respecto a la entrada actual)
 go	 Carga una URL de la lista. Por ejemplo, para volver a cargar la página una página que se visitó anteriormente: window.history.go (-1);