

# Bootcamp de Full Stack

Bienvenidos a la clase N°31

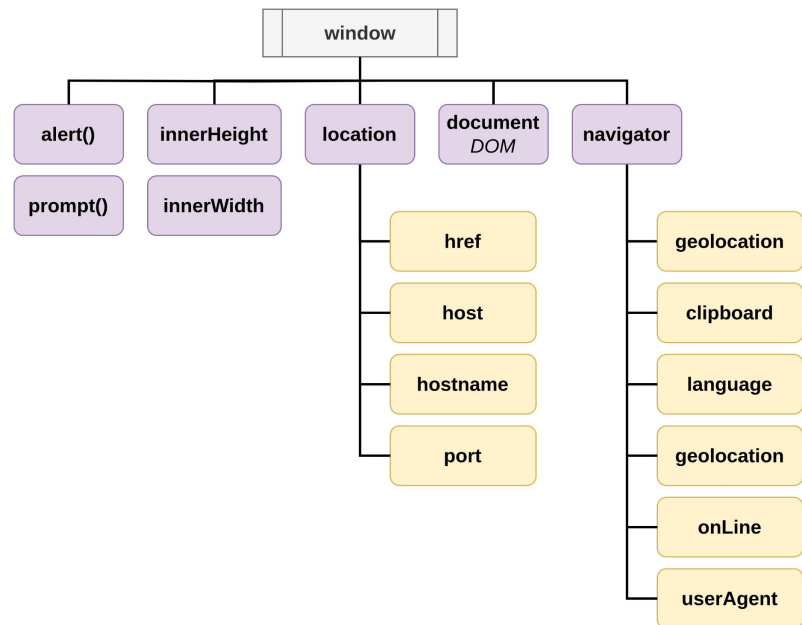
- BOM
- Window API
- Location API
- Navigator API
- History API
- Proyecto Integrador F1 (i4)

# Fundamentos

## BOM

El BOM (Browser Object Model) es una convención específica que se refiere a todos los objetos expuestos por el navegador web. Es decir, es una estructura de datos (M) de tipo Object (O) que representa al navegador en el que se está ejecutando el código de JavaScript (B). Se accede a través del objeto **window**.

BOM



# JavaScript

## Windows API

La **Window API** es la interfaz principal del navegador en JavaScript. Representa la ventana del navegador (o pestaña) donde se carga una página web y actúa como el objeto global en el entorno del cliente.

### Métodos:

- `window.alert()` Muestra un mensaje emergente.
- `window.confirm()` Muestra un diálogo con Aceptar / Cancelar. Devuelve true o false.
- `window.prompt()` Muestra un diálogo con campo de texto. Devuelve el texto o null.

### Propiedades:

- `window.innerWidth` Ancho actual de la ventana visible.
- `window.innerHeight` Alto actual de la ventana visible.

# JavaScript

## Location API

La **Location API** es una interfaz del navegador accesible a través de `window.location` que permite obtener información sobre la URL actual de la página y controlar la navegación (redirigir, recargar, etc.).

### Métodos:

- `location.reload()` Recarga la página.
- `location.assign(url)` Redirige a otra URL.

### Propiedades:

- `location.href` URL completa (también se usa para asignar un nuevo valor para redirigir).
- `location.hostname` Nombre del dominio.
- `location.pathname` Ruta del archivo actual.
- `location.search` Parámetros `?clave=valor`



# JavaScript

## Navigator API

La **Navigator API** es una interfaz del navegador que proporciona información sobre el entorno en el que se está ejecutando la página web, como el navegador, sistema operativo, estado de la conexión, idioma preferido y otras características del dispositivo o software.

### Métodos:

- `navigator.geolocation.getCurrentPosition()`      Obtiene la ubicación actual (requiere permiso).

### Propiedades:

- |                                    |  |
|------------------------------------|--|
| • <code>navigator.userAgent</code> | Información sobre navegador y sistema operativo. |
| • <code>navigator.language</code>  | Idioma preferido del navegador.                  |
| • <code>navigator.platform</code>  | Plataforma del sistema operativo.                |
| • <code>navigator.vendor</code>    | Nombre del proveedor del navegador.              |
| • <code>navigator.onLine</code>    | Indica si hay conexión a Internet (true/false).  |

# JavaScript

## History API

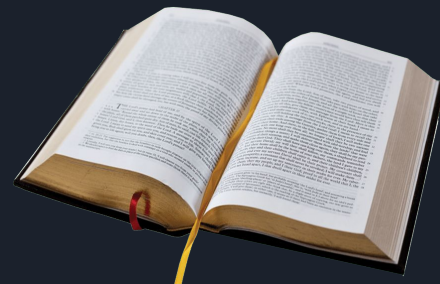
La **History API** es una interfaz del navegador que te permite interactuar con el historial de navegación del usuario, es decir, ver, modificar o navegar entre las páginas visitadas, sin recargar la página.

### Métodos:

- `history.back()` Ir a la página anterior.
- `history.forward()` Ir a la siguiente página.
- `history.go(n)` Navega a una posición relativa (**n** puede ser un valor positivo o negativo).

### Propiedades:

- `history.length` Número de entradas en el historial.



# BREAK

Descansemos 10 minutos



# Proyecto Integrador F1

## Iteración N°4



Para llevar a cabo el proyecto integrador F1, accede al documento de la consigna a través del siguiente enlace:

<https://docs.google.com/document/d/181methzV3GUL-YCBNp-qE94qE8-eEiA9Q169gyzUvSs/edit?usp=sharing>



# CIERRE DE CLASE

Continuaremos en la próxima clase

