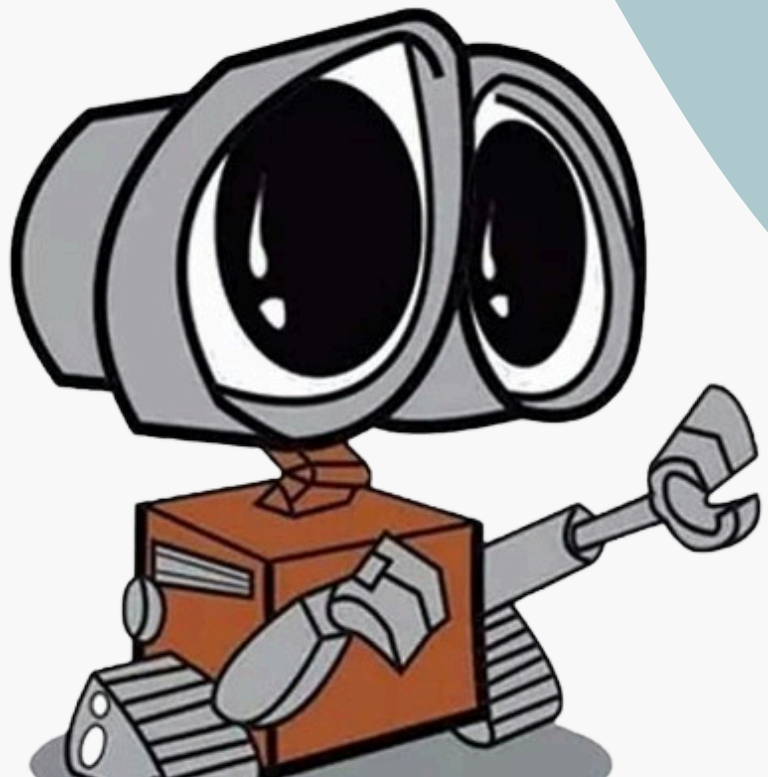


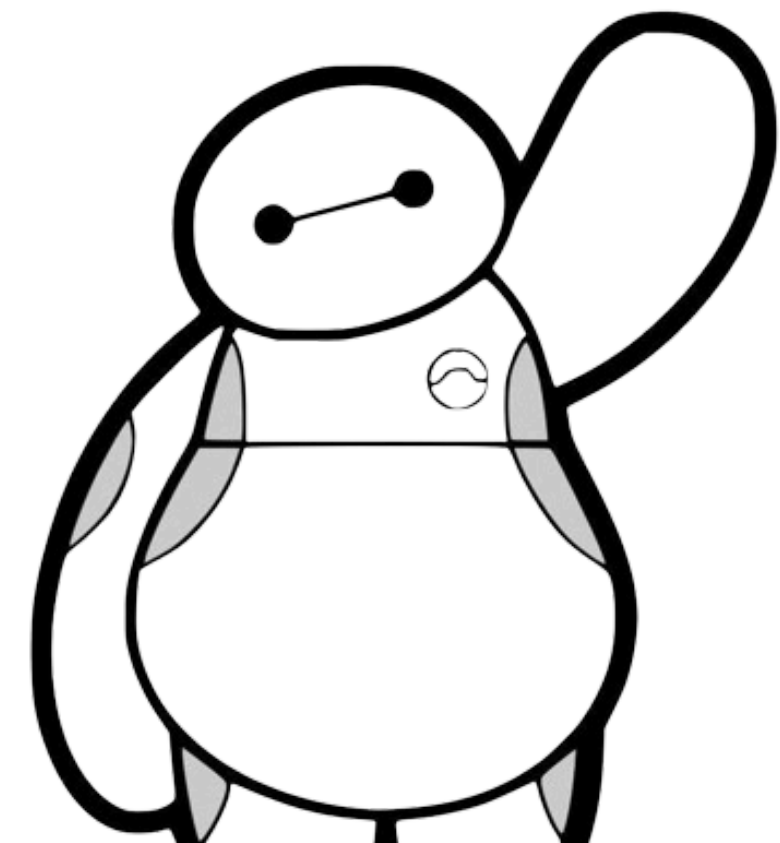
# OBJETO WINDOW

Katherine Vásquez  
Ana Maria Sarria



# Contenido

1. Concepto
2. Propiedades
3. Eventos
4. Metodos
5. Ejemplos



# Qué es el objeto window?

El objeto window representa la ventana del navegador que contiene una página web. Funciona como un puente de comunicación entre el código JavaScript y la ventana del navegador, permitiendo al desarrollador controlar y modificar diversos aspectos de la experiencia del usuario, como:


- Carga de contenido
- interacción con elementos
- Navegacion
- Comunicación con el servidor
- Informacion de usuario





# características

El objeto window es un componente central del modelo de objetos del navegador en JavaScript. Algunas de sus características más importantes son:

- Representación de la ventana del navegador
  - Globalidad
  - Propiedades de ventana
  - Métodos y eventos
  - Manipulación del DOM
  - Gestión de ventanas y marcos
- 

# PROPIEDADES

## 1. `window.document`

Representa el documento cargado en la ventana y proporciona métodos para interactuar con el DOM (Document Object Model).

## 2. `window.location`

Proporciona información sobre la URL de la página actual y permite cambiarla para redirigir a otra página.

## 3. `window.navigator`

Contiene información sobre el navegador del usuario, como el nombre, la versión y las características del navegador.

## 4. `innerWidth` y `innerHeight`

Representan el ancho y alto del área de contenido de la ventana del navegador, excluyendo barras de desplazamiento, bordes y barras de herramientas.

## 5. `window.screen`

Proporciona información sobre la pantalla del usuario, como su tamaño y resolución.

## load

Este evento se dispara cuando se completa la carga de todos los recursos de la página web, como imágenes, scripts y estilos

## unload

Se activa cuando el usuario abandona la página o cierra la ventana del navegador.

Puedes usar este evento para realizar limpieza de recursos o realizar acciones antes de que el usuario deje la página

# EVENTOS

## `window.alert()`

Muestra un cuadro de diálogo con un mensaje y un botón "Aceptar"

## `window.prompt()`

Muestra un cuadro de diálogo con un mensaje, un campo de entrada para que el usuario ingrese texto y botones "Aceptar" y "Cancelar". Devuelve el texto ingresado por el usuario o null si hace clic en "Cancelar".

# METODOS

# Ejemplo de Propiedad

La propiedad localStorage es un objeto de almacenamiento local que proporciona el navegador para que puedas almacenar datos de manera persistente en el dispositivo del usuario. Funciona como una especie de base de datos de clave-valor donde puedes guardar información relevante para tu aplicación web, como preferencias del usuario, configuraciones o datos de sesión.

```
// Propiedad localStorage para almacenar datos localmente en el navegador
// Guardar datos en localStorage
window.localStorage.setItem('usuario', 'Juan');

// Obtener datos de localStorage
const usuario = window.localStorage.getItem('usuario');
console.log(usuario); // Imprime: Juan
```



# Ejemplo de Metodo

El método `open()` es utilizado para abrir una nueva ventana del navegador o una nueva pestaña, dependiendo de la configuración del navegador y los permisos del usuario. Es útil cuando quieres abrir contenido externo o generar una ventana emergente con información adicional.

```
// Método open() para abrir una nueva ventana del navegador
function abrirVentana() {
    window.open('https://www.ejemplo.com', '_blank', 'width=600,height=400');
}
```

# Ejemplo de Evento

En el ejemplo Objeto de evento (DOMContentLoaded), se añade un evento DOMContentLoaded que ejecutará una función anónima para imprimir un mensaje en la consola cuando se complete la carga del DOM.

```
// Evento DOMContentLoaded que se dispara cuando se ha cargado
window.addEventListener('DOMContentLoaded', (event) => {
  console.log('DOM completamente cargado y analizado');
});
```

**!Gracias  
por su  
atención!**

