EDUCACIÓN IT

Bootcamp de Full Stack F2

Bienvenidos a la clase N°40

- POO - Clases - Herencia
- Patrón Singleton
- Iteración de Objetos
- Cookies
- LocalStorage
- SessionStorage
- Bonus Extra: Patrón Observer

IT

EDUCACIÓN IT

JavaScript

POO - Clases - Herencia

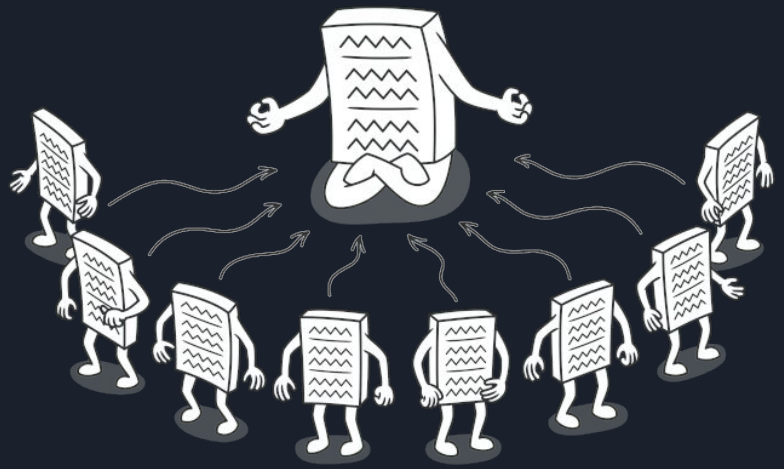
- **POO**: Programación orientada a objetos. este es un paradigma de programación que parte del concepto de "objetos" como base, los cuales contienen información en forma de propiedades y código en forma de métodos. El propósito de su creación fue acercar el manejo de las estructuras de un programa al manejo de las cosas en el mundo real, de ahí el nombre "objeto" como algo genérico, que puede representar cualquier cosa tangible. Las principales características de la POO es el encapsulamiento, herencia y polimorfismo.
- Una **clase** es un elemento de la POO que actúa como una plantilla o molde y va a definir las características y comportamientos de una entidad.
- La **herencia** es un mecanismo basado en clases, por medio del cual una clase se deriva de otra de manera que extiende su funcionalidad.

EDUCACIÓN IT

JavaScript

Patrón Singleton

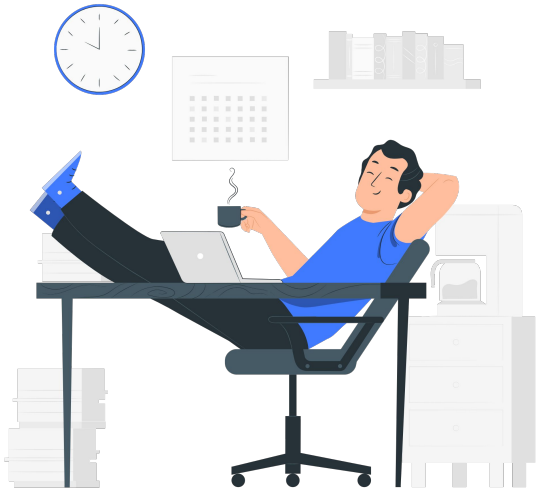
- **Singleton** es un patrón de diseño creacional que nos permite asegurarnos de que una clase tenga una única instancia, a la vez que proporciona un punto de acceso global a dicha instancia.



EDUCACIÓN IT

BREAK

Descansemos 15 minutos



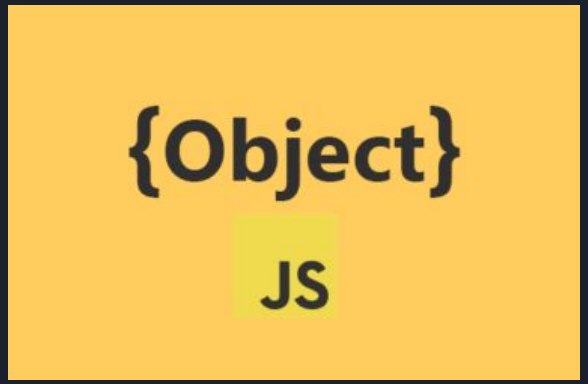
EDUCACIÓN IT

JavaScript

Iteración de Objetos

La instrucción **for...in** se utiliza para iterar sobre las propiedades enumerables de un objeto. Donde la prop es una variable que toma el valor de la propiedad en cada iteración y el objeto es el cual se está iterando.

```
for (prop in objeto) {
        // Código a ejecutar en cada iteración
}
```



EDUCACIÓN IT

JavaScript

Cookies

Una cookie HTTP, cookie web o cookie de navegador es una pequeña pieza de datos que un servidor envía al navegador web del usuario. El navegador guarda estos datos y los envía de regreso junto con la nueva petición al mismo servidor. Las cookies se utilizan principalmente con tres propósitos:

- Gestión de Sesiones: Inicios de sesión, carritos de compras, puntajes de juegos, etc.
- Personalización: Preferencias de usuario, temas y otras configuraciones.
- Rastreo: Guardar y analizar el comportamiento del usuario.

Características:

- Los datos se almacenan en pares clave-valor (solo permite valores de tipo string).
- Los datos perduran cuando se cierra la pestaña o ventana.
- Se permite el acceso a los datos entre las diferentes pestañas o ventanas del navegador.
- Pueden tener fecha de expiración.

Comandos:

- Agregar o modificar cookie: `document.cookie = "marca=ford; expires=Fri, 16 Aug 2024 00:00:00 UTC;";`
- Obtener todas las cookies: `document.cookie;`
- Eliminar cookie: `document.cookie = "marca=; expires=Thu, 01 Jan 1970 00:00:00 UTC;";`

EDUCACIÓN IT

JavaScript

LocalStorage

La propiedad **localStorage** es un objeto de almacenamiento en el lado del cliente que permite a una aplicación web almacenar datos de manera local.

Características:

- Los datos se almacenan en pares clave-valor (solo permite valores de tipo string).
- Los datos perduran cuando se cierra la pestaña o ventana.
- Se permite el acceso a los datos entre las diferentes pestañas o ventanas del navegador.
- No tiene fecha de expiración.

Comandos:

- Agregar o modificar ítem: `localStorage.setItem("marca", "Ford");`
- Obtener ítem: `localStorage.getItem("marca");`
- Quitar ítem: `localStorage.removeItem("marca");`
- Quitar todos los ítems: `localStorage.clear();`

EDUCACIÓN IT

JavaScript

SessionStorage

La propiedad **sessionStorage** es un objeto de almacenamiento en el lado del cliente que permite a una aplicación web almacenar datos de manera temporal durante la duración de una sesión.

Características:

- Los datos se almacenan en pares clave-valor (solo permite valores de tipo string).
- Los datos se pierden cuando se cierra la pestaña o ventana.
- No se permite el acceso a los datos entre las diferentes pestañas o ventanas del navegador.
- No tiene fecha de expiración.

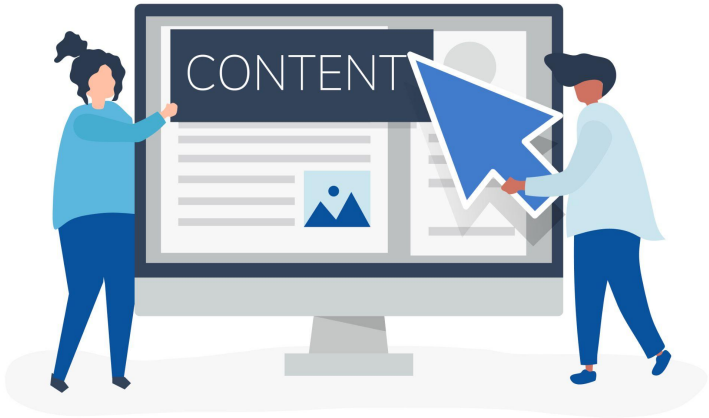
Comandos:

- Agregar o modificar ítem: `sessionStorage.setItem("marca", "Ford");`
- Obtener ítem: `sessionStorage.getItem("marca");`
- Quitar ítem: `sessionStorage.removeItem("marca");`
- Quitar todos los ítems: `sessionStorage.clear();`

EDUCACIÓN IT

Bonus Extra

El siguiente contenido es opcional y profundiza sobre patrones de diseño

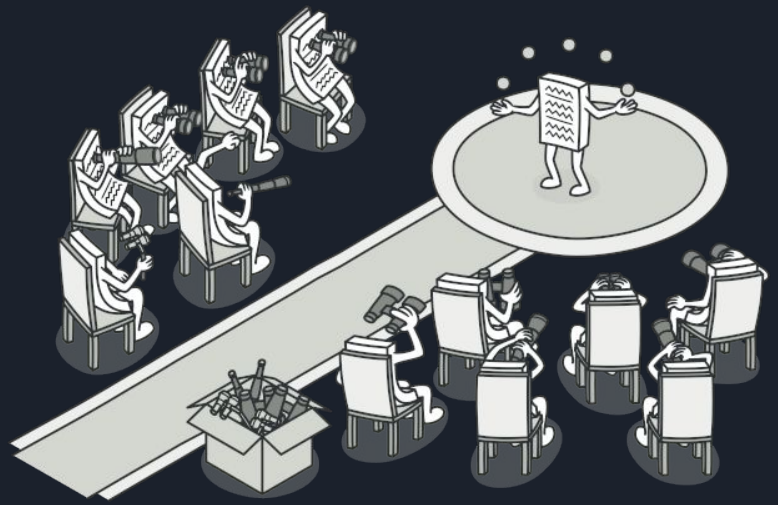


EDUCACIÓN IT

JavaScript

Patrón Observer

- **Observer** es un patrón de diseño de comportamiento que te permite definir un mecanismo de suscripción para notificar a varios objetos sobre cualquier evento que le suceda al objeto que están observando.



EDUCACIÓN IT

CIERRE DE CLASE

Continuaremos en la próxima clase