EJEMPLO TRATAMIENTO DE VARIABLES EN JAVASCRIPT

Ejemplo:

Tienes varias páginas HTML (pagina1.html, pagina2.html, etc.), cada una con su propio CSS y JS, todas en el mismo dominio (ej: tudominio.com/proyecto/...).

Se pueden pasar variables entre páginas?

Sí, puedes pasar datos/valores entre páginas de distintas formas, pero NO directamente como variables de JS entre archivos, porque cada página recarga su propio entorno. Sin embargo, puedes compartir esos valores usando:

1. localStorage (la más simple y potente)

Guarda valores que persisten entre páginas mientras no los borres.

javascript

```
// En la página 1 (antes de ir a la página 2)
localStorage.setItem("usuario", "Mar");
```

javascript

```
// En la página 2
const nombre = localStorage.getItem("usuario");
console.log(nombre); // "Mar"
```

Perfecto para pasar nombre de usuario, idioma, selección, configuraciones, etc.

2. sessionStorage

Igual que localStorage, pero los datos se borran cuando cierras el navegador o la pestaña.

javascript

```
sessionStorage.setItem("tema", "oscuro");
```

3. Query Parameters (URL)

Pasas datos como parámetros en la URL:

javascript

```
// En la página 1
window.location.href = "pagina2.html?color=azul&user=mar";
```

javascript

```
// En la página 2 (leer parámetros)
const params = new URLSearchParams(window.location.search);
const color = params.get("color"); // azul
```

Muy útil para filtros, resultados, mensajes, selecciones, etc.

4. Cookies

Menos usado hoy para lógica cliente, pero válido. Sirve si quieres que el dato se mantenga por días o meses.

Lo que no puedes hacer:

No puedes compartir directamente una variable declarada con let o const entre páginas como si fuera global, porque cada HTML carga su propio archivo JS y reinicia su contexto.

No puedes acceder a variables JS de una página desde otra sin pasar los datos explícitamente.

Y sobre los CSS y JS

Si todos tus archivos .html hacen referencia a los mismos CSS y JS, entonces:

Puedes usar el mismo diseño y funcionalidad compartida.

Puedes detectar los datos pasados (por localStorage, URL, etc.) y modificar el comportamiento de cada página con JavaScript según lo recibido.

Ejemplo real de uso:

html

```
<!-- pagina1.html -->
<script>
  localStorage.setItem("colorFondo", "blue");
  window.location.href = "pagina2.html";
</script>
```

html

```
<!-- pagina2.html -->
<script>
  const color = localStorage.getItem("colorFondo");
  document.body.style.backgroundColor = color;
</script>
```

Pasar datos entre páginas HTML con JavaScript

Dos formas comunes de pasar información entre páginas HTML en un mismo dominio usando JavaScript:

1. Usando parámetros en la URL

Puedes enviar datos agregándolos a la URL como parámetros. Esto es útil para filtros, configuraciones rápidas o selección de estilos.

Ejemplo:

pagina2.html?color=#00ff00
En la página destino puedes obtener el valor con URLSearchParams:

```
const params = new URLSearchParams(window.location.search);
const color = params.get('color');
```

2. Usando sessionStorage

`sessionStorage` permite guardar datos durante toda la sesión (hasta cerrar la pestaña o el navegador).

Esto es útil cuando no quieres mostrar los datos en la URL, pero necesitas mantenerlos entre páginas.

Ejemplo para guardar el valor:

```
sessionStorage.setItem('colorSeleccionado', '#ffcc00');
```

Y para recuperarlo en otra página:

```
const color = sessionStorage.getItem('colorSeleccionado');
```

Ventajas de cada método

Método	Ventajas	Ideal para
Parámetros en URL	- Visible, fácil de compartir - No necesita almacenamiento local	Filtros, enlaces rápidos, navegación por estado
sessionStorage	- Oculto al usuario - Persistente durante la sesión	Preferencias de usuario, configuraciones temporales

Ambos métodos son compatibles con todos los navegadores modernos y útiles para mejorar la experiencia entre múltiples páginas HTML.