

# Bootcamp de Full Stack

Bienvenidos a la clase N°35

- Cookies
- LocalStorage
- SessionStorage
- Proyecto Integrador F1
  - Espacio de consultas

# JavaScript

## Cookies

Una cookie HTTP (también conocida como cookie web o cookie de navegador) es un pequeño fragmento de información que un servidor web envía al navegador del usuario. El navegador almacena esta información localmente y la envía de vuelta automáticamente en cada solicitud posterior al mismo servidor, permitiendo mantener contexto entre interacciones.

¿Para qué se utilizan las cookies?

- **Gestión de sesiones:** por ejemplo, recordar que un usuario inició sesión, mantener el contenido de un carrito de compras, guardar un progreso de juego.
- **Personalización:** Ej: guardar preferencias de idioma, tema claro/oscuro, o configuraciones específicas del usuario.
- **Rastreo y análisis:** Ej: seguimiento de comportamiento del usuario, visitas a páginas, clics.

# JavaScript

## Cookies

### Características:

- Los datos se almacenan como pares **clave-valor** (siempre como **strings**).
- Son compartidas entre pestañas o ventanas que apuntan al **mismo dominio**.
- Se mantienen después de cerrar el navegador si tienen una **fecha de expiración definida**.
- Se pueden configurar para:
  - Definir una fecha de expiración o duración personalizada usando **expires** o **max-age**.
  - Establecer las rutas del sitio web donde estará disponible la cookie utilizando **path**.
  - Evitar que JavaScript acceda a la cookie con **HttpOnly**, protegiendo contra ataques XSS.
  - Enviar solo por conexiones HTTPS usando **Secure** para prevenir exposición en redes inseguras.

*Dato extra: XSS (Cross-Site Scripting) es un ataque donde un atacante inserta código malicioso en una página web para que se ejecute en el navegador de otros usuarios, permitiéndole robar datos como cookies o realizar acciones sin permiso.*

# JavaScript

## LocalStorage

### ¿Qué es `localStorage`?

Es una API de almacenamiento web proporcionada por los navegadores que permite guardar datos de forma persistente en el navegador del cliente. A diferencia de las cookies, los datos almacenados en `localStorage` no se envían al servidor en cada solicitud HTTP y no caducan automáticamente. Es ideal para almacenar información que necesita mantenerse disponible incluso después de cerrar el navegador.

#### Características:

- Los datos se almacenan en pares **clave-valor** (solo permite valores de tipo **string**).
- Los **datos se mantienen** incluso después de cerrar el navegador.
- Se **comparte el acceso** a los datos entre las diferentes pestañas o ventanas del navegador.
- No tiene fecha de expiración.
- Tiene un espacio limitado (entre 5 y 10 MB por dominio, según el navegador).

# JavaScript

## SessionStorage

### ¿Qué es `sessionStorage`?

Es una API de almacenamiento web del navegador que permite guardar datos de forma temporal durante una única sesión del navegador. A diferencia de `localStorage`, los datos en `sessionStorage` se eliminan automáticamente cuando se cierra la pestaña o ventana donde fueron creados. Es útil para guardar información transitoria que no necesita persistencia entre sesiones.

#### Características:

- Los datos se almacenan en pares **clave-valor** (solo permite valores de tipo **string**).
- Los **datos no se mantienen** después de cerrar el navegador.
- Los **datos no son compartidos** porque cada pestaña o ventana tiene su propia sesión.
- No tiene fecha de expiración.
- Tiene un espacio limitado (entre 5 y 10 MB por dominio, según el navegador).

# BREAK

Descansemos 10 minutos



# Proyecto Integrador F1

## Espacio De Consultas

- ? ¿Crearon el repositorio del proyecto de forma personal en GitHub?
- ? ¿Lograron compilar Sass sin errores y utilizaron el patrón 7.1?
- ? ¿Establecieron el uso de variables de Sass en todo el código?
- ? ¿Emplearon la nomenclatura BEM, convención camelcase para JS y kebab-case para html y css?
- ? ¿Implementaron el closure en el contador de unidades destinadas al carrito?
- ? ¿Validaron el HTML y el CSS con las herramientas del W3C? ¿Qué errores detectaron?
- ? ¿Cómo se aseguraron de que el sitio se vea bien en distintos tamaños de pantalla?
- ? ¿Probaron el sitio en más de un navegador? ¿Detectaron diferencias?

Consigna del proyecto integrador F1:

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

# CIERRE DE CLASE

Continuaremos en la próxima clase

