

Diferença entre `localStorage` e `sessionStorage`

Persistência dos Dados

localStorage: Os dados persistem indefinidamente, mesmo após o navegador ser fechado. Devem ser removidos manualmente ou por código.

sessionStorage: Os dados são apagados ao fechar a aba ou janela do navegador. Ideal para dados que só fazem sentido durante a sessão atual.

Escopo

localStorage: Compartilha dados entre todas as abas e janelas do mesmo navegador e domínio.

sessionStorage: Escopo limitado à aba ou janela onde foi criado, mesmo no mesmo domínio.

Capacidade de Armazenamento

Ambos suportam até **5MB por domínio**, dependendo do navegador.

Uso e APIs

Ambos possuem os mesmos métodos para manipular dados:

- `setItem(key, value)`: Armazena um valor com uma chave.
- `getItem(key)`: Recupera o valor de uma chave.
- `removeItem(key)`: Remove o valor de uma chave.
- `clear()`: Remove todos os itens armazenados.

Exemplo de Código

```
// localStorage
localStorage.setItem("username", "Pablo");
console.log(localStorage.getItem("username")); // Pablo
localStorage.removeItem("username");

// sessionStorage
```

```
sessionStorage.setItem("sessionKey", "12345");  
console.log(sessionStorage.getItem("sessionKey")); // 12345  
sessionStorage.clear(); // Limpa todos os dados desta sessão
```

Quando Usar Cada Um?

localStorage: Use para dados que precisam persistir entre sessões, como:

- Tokens de autenticação.
- Preferências do usuário (tema, configurações, etc.).
- Histórico de pesquisa.

sessionStorage: Use para dados temporários, como:

- Dados de formulários ou carrinhos de compras.
- Estado de navegação temporário entre páginas.

[Ir para o exercício](#)