

# Salvando Informações no LocalStorage

---

# O que é LocalStorage?

O Local Storage (armazenamento local) é uma tecnologia de armazenamento de dados disponível nos navegadores da web modernos. Ele permite que os sites e aplicativos da web armazenem dados de forma persistente no computador ou dispositivo do usuário.

Ao contrário dos cookies, que são limitados em tamanho e são enviados automaticamente para o servidor em cada solicitação, o Local Storage oferece uma capacidade de armazenamento maior (geralmente várias megabytes) e os dados não são enviados para o servidor automaticamente. Isso permite que os desenvolvedores armazenem informações locais no navegador do usuário, como preferências do usuário, configurações personalizadas, estado da aplicação e até mesmo dados offline.

# O que é local Storage?

O Local Storage é implementado usando a API do Web Storage, que fornece duas interfaces principais: **localStorage** e **sessionStorage**. Ambas as interfaces oferecem métodos simples para armazenar e recuperar dados no navegador. A diferença entre elas é que o **localStorage** armazena os dados de forma persistente, enquanto o **sessionStorage** armazena os dados apenas durante a sessão atual do navegador. Ou seja, os dados do **sessionStorage** são excluídos quando o navegador é fechado.

# O que é local Storage?

Os dados armazenados no Local Storage são acessíveis apenas pelo site ou aplicativo que os criou. Eles são armazenados como pares chave-valor, em que as chaves e valores são strings. No entanto, você pode converter outros tipos de dados em strings antes de armazená-los ou convertê-los de volta ao recuperá-los.

O Local Storage é amplamente utilizado pelos desenvolvedores da web para fornecer uma experiência personalizada aos usuários, lembrando suas preferências, configurações e históricos. No entanto, é importante observar que o Local Storage não é adequado para armazenar informações sensíveis, como senhas, pois os dados são acessíveis pelo JavaScript no contexto do site ou aplicativo.

# Características do Local Storage

Capacidade de armazenamento: A capacidade de armazenamento do Local Storage varia de navegador para navegador, mas geralmente é de pelo menos alguns megabytes (MB). No entanto, é importante ter em mente que esse espaço é compartilhado entre todos os sites e aplicativos da web no mesmo domínio.

Acesso aos dados: Os dados armazenados no Local Storage podem ser acessados e manipulados pelo JavaScript no contexto do site ou aplicativo que os criou. Isso permite que os desenvolvedores leiam, modifiquem ou removam os dados conforme necessário.

# Características do Local Storage

Domínio restrito: O Local Storage está restrito a um domínio específico. Os dados armazenados por um site ou aplicativo não podem ser acessados por outros sites ou aplicativos em domínios diferentes. Isso ajuda a garantir a segurança dos dados e a privacidade do usuário.

Uso de strings: O Local Storage armazena os dados como strings. Isso significa que, se você deseja armazenar outros tipos de dados, como números ou objetos JavaScript, eles precisam ser convertidos em strings antes de serem armazenados. Da mesma forma, ao recuperar os dados do Local Storage, você precisará convertê-los de volta ao tipo de dados original, se necessário.

# Características do Local Storage

Eventos de mudança: O Local Storage dispara eventos quando os dados são modificados. Os eventos "storage" podem ser usados para detectar alterações nos dados do Local Storage em diferentes guias ou janelas do navegador. Isso permite que os sites ou aplicativos atualizem e sincronizem as informações conforme necessário.

Mas como inserir e pegar itens do Local Storage?

# Parte prática – Local Storage

Para adicionar itens no local Storage a partir do Javascript, utilizamos a propriedade 'localStorage' e o método 'setItem()' disponíveis na API do Web Storage.



```
1  <script>
2
3      // Defina um valor no Local Storage
4      localStorage.setItem('nome', 'João');
5
6  </script>
```



# Parte prática – Local Storage

Neste exemplo, estamos definindo um item chamado 'nome' no Local Storage e atribuindo o valor 'João' a ele.

Você pode adicionar quantos itens desejar, basta repetir o código acima com chaves (chave) e valores diferentes.

# Parte prática – Local Storage

Agora, você tem três itens diferentes no Local Storage: 'nome', 'idade' e 'cidade'.

Lembre-se de que os dados no Local Storage são armazenados como strings.

```
// Defina mais itens no Local Storage  
localStorage.setItem('idade', '25');  
localStorage.setItem('cidade', 'São Paulo');
```

# Parte prática – Local Storage

Se você deseja armazenar outros tipos de dados, como números ou objetos, é necessário convertê-los em strings antes de armazená-los. Por exemplo:

```
// Armazene um número no Local Storage  
var idade = 25;  
localStorage.setItem('idade', idade.toString());  
  
// Armazene um objeto no Local Storage  
var pessoa = { nome: 'João', idade: 25, cidade: 'São Paulo' };  
localStorage.setItem('pessoa', JSON.stringify(pessoa));
```

# Parte prática – Local Storage

Ao armazenar objetos, é comum usar `JSON.stringify()` para converter o objeto em uma string JSON antes de armazená-lo no Local Storage.

Lembre-se de que o Local Storage tem uma capacidade limitada. Portanto, evite armazenar grandes quantidades de dados, especialmente se eles não forem necessários para a funcionalidade do seu site ou aplicativo.

Para acessar os itens armazenados no Local Storage posteriormente, você pode usar o método `getItem()` da seguinte maneira:

# Parte prática – Local Storage

Assim, você pode recuperar os valores dos itens armazenados fornecendo a chave correspondente ao método `getItem()`.

```
1 // Recupere um item do Local Storage
2 var nome = localStorage.getItem('nome');
3 console.log(nome); // imprime 'João'
```

# Vamos praticar

Vamos criar uma tela de login, com a opção de se cadastrar. Ao se cadastrar, os dados devem ser salvos no Local Storage. Ao Fazer login, leve para uma outra página em branco com a mensagem "Login efetuado com sucesso".