

# Parte 1: O que é JSON?

## Definição

- JSON (JavaScript Object Notation) é um formato leve de troca de dados.
- É baseado em texto e fácil de ler e escrever para humanos, bem como de interpretar e gerar para máquinas.

## Estrutura do JSON

JSON é composto por:

1. **Objetos:** Representados por chaves `{}`.
2. **Arrays:** Representados por colchetes `[]`.
3. **Valores:** Podem ser strings, números, booleanos, arrays, objetos ou `null`.

## Exemplo de JSON:

```
Unset
{
  "nome": "João",
  "idade": 25,
  "habilidades": ["JavaScript", "HTML", "CSS"],
  "empregado": true
}
```

## Vantagens do JSON

- Leve e rápido.
- Facilmente integrado com linguagens de programação modernas.
- Muito utilizado em APIs.

---

# Parte 2: Introdução ao Local Storage

## O que é Local Storage?

- É uma API do navegador que permite armazenar dados localmente no dispositivo do usuário.
- Os dados permanecem salvos mesmo após o fechamento do navegador.

## Como funciona o Local Storage?

- Os dados são armazenados como pares **chave-valor**.
- Todos os valores são armazenados como strings.

## Comandos básicos

### 1. Salvar dados:

Unset

2.

```
localStorage.setItem("chave", "valor");
```

3.

### Recuperar dados:

Unset

4.

```
const valor = localStorage.getItem("chave");
```

5.

### Remover dados:

Unset

6.

```
localStorage.removeItem("chave");
```

7.

### Limpar tudo:

Unset

8.

```
localStorage.clear();
```

---

## Parte 3: Diferença entre Local Storage e Session Storage

### Local Storage

- Dados persistem até serem removidos manualmente.
- Ideal para informações que precisam ser salvas por mais tempo, como configurações de usuário.

### Session Storage

- Dados são apagados ao fechar a aba ou o navegador.
- Melhor para informações temporárias, como dados de uma sessão específica.

### Comparativo:

Característica	Local Storage	Session Storage
Persistência	Permanente	Temporária
Escopo	Por origem (domain)	Por aba
Tamanho máximo	Geralmente 5 MB por origem	Geralmente 5 MB por origem

---

## Parte 4: Trabalhando com JSON e Local Storage

### Exemplo prático

Vamos salvar e recuperar um objeto JSON no Local Storage:

```
Unset
// 1. Criando um objeto JavaScript
const usuario = {
  nome: "Ana",
  idade: 30,
  preferencias: ["filmes", "livros"]
};

// 2. Convertendo para JSON e salvando no Local Storage
localStorage.setItem("usuario", JSON.stringify(usuario));

// 3. Recuperando o JSON do Local Storage e convertendo para
um objeto JavaScript
const dados = localStorage.getItem("usuario");
const usuarioRecuperado = JSON.parse(dados);

// 4. Exibindo os dados no console
console.log(usuarioRecuperado);
```

### Observação

- Sempre use `JSON.stringify` para converter objetos em strings antes de salvar no Local Storage.

- Use `JSON.parse` para converter as strings recuperadas em objetos.
-