

Instituto Federal de Educação, Ciência e Tecnologia do Paraná - IFPR

disciplina de
DESENVOLVIMENTO WEB

JAVASCRIPT II

Professor Jefferson de Oliveira Chaves
jefferson.chaves@ifpr.edu.br

Agenda

- O formato JSON;
- Armazenando Dados Localmente;
- LocalStorage;

O formato **JSON**

- JSON é um acrônimo para **Javascript Object Notation**;
- Formato compacto, de padrão aberto de troca de dados simples e rápida (parsing) entre sistemas;
- Embora seja derivado do JavaScript, o JSON se tornou padrão para troca de dados na web;
- É independente de linguagem de programação;
- Praticamente toda linguagem (menos Portugol) consegue lidar com JSON:
 - PHP, Java, JavaScript, Python e etc.

O formato **JSON**

- Objetos JSON podem possuir atributos e métodos, assim como em outras linguagens orientadas a objetos;

```
let carro = {  
  placa: " ABC-1234",  
  ano: 2013,  
  acessorios : ["ar-condicionado", "rodas esportivas"]  
}
```

- Para usar os atributos:
 - carro.ano = 2012;
 - console.log(carro.placa);
- Para adicionar novo atributo:
 - carro.cor = "dourado";

O formato **JSON**

```
let carro = {  
  "turma": "2º ano",  
  "ano": 2023,  
  "alunos": [  
    { "nome": "Edson Sales", "notas": [8, 9, 5] },  
    { "nome": "Luiz Velli ", "notas": [8, 10, 7] },  
    { "nome": "Caio Plata", "notas": [10, 10, 9] }  
  ]  
}
```

Tipos de dados Suportados

- Os Tipos de dados que podem ser representados pelo JSON:
 - Strings: "exemplo"
 - Números: 42
 - Booleanos: true ou false
 - Arrays: [1, 2, 3]
 - Objetos: { "chave": "valor" }
 - Null: null

Prática com JSON

- Existem alguns métodos para manipulação de **JSON** :
 - **JSON.stringify()** para converter um objeto JavaScript em JSON.
 - **JSON.parse()** para converter uma string JSON em um objeto JavaScript.

```
let objeto = { nome: "Ana", idade: 25 };  
let json = JSON.stringify(objeto); // Converte objeto em string JSON  
console.log(json); // {"nome":"Ana","idade":25}
```

```
let jsonString = '{"nome":"Carlos","idade":28}';  
let obj = JSON.parse(jsonString); // Converte JSON para objeto  
console.log(obj.nome); // Carlos
```

Armazenando dados com **LocalStorage**



O que é **LocalStorage**?

- O **localStorage** é um conjunto de mecanismos que permitem que os navegadores armazenem informações seguindo estrutura do **JSON**, ou seja, **chave:valor**;
- Os pares de **chave:valor** representam objetos em formato de strings;
- Você pode acessar esses valores como um objeto ou usando o método **getItem()** (mais sobre isso posteriormente);
- Normalmente os dados são organizados **segundo o modelo JSON**.

O que é **LocalStorage**?

- O **localStorage** não é adequado para grandes volumes de dados;
- Os dados armazenados são acessíveis a qualquer pessoa que usa o dispositivo, portanto, não deve ser usado para armazenar informações confidenciais;
- Pode ser usado para armazenar as preferências do usuário, como idioma ou tema;
- Também é possível usá-lo para armazenar dados em cache se usá-lo com frequência;
- LocalStorage pode armazenar dados de formulário que serão perdidos se o usuário fechar o navegador.

Propriedade e métodos do **LocalStorage**

- Existem 5 principais métodos para manipular o localStorage:
 - **setItem()**: Adicionar chave e valor ao localStorage;
 - **getItem()**: É assim que você obtém itens do localStorage;
 - **removeItem()**: Remove um item por chave;
 - **clear()**: Limpa todo o armazenamento local;
 - **key()**: Usa um índice para recuperar a chave de um localStorage.

Propriedade e métodos do **LocalStorage**

```
//registrar dados
localStorage.setItem('nome', 'João');
localStorage.setItem('idade', '30');

// recuperar dados
let nome = localStorage.getItem('nome');
let idade = localStorage.getItem('idade');
console.log(nome); // João
console.log(idade); // 30

//limpar dados
localStorage.clear(); // Remove todos os dados armazenados
```


Desafio 1

- Crie uma página em que o usuário possa salvar seu nome e preferências (tema escuro ou claro e tamanho da fonte) no LocalStorage e, quando ele voltar à página, esses dados sejam automaticamente carregados.

Trabalhando com Objetos no **LocalStorage**

- O **LocalStorage** armazena apenas strings;
- Para armazenar objetos, precisamos converter o objeto para uma string JSON com `JSON.stringify()` e usar `JSON.parse()` para convertê-lo de volta a um objeto.

Trabalhando com Objetos no **LocalStorage**

```
let aluno = {  
  nome: 'Ana',  
  idade: 25,  
  curso: 'Desenvolvimento Web'  
};
```

```
// Salvando o objeto como string JSON
```

```
localStorage.setItem('aluno', JSON.stringify(aluno));
```

```
// Recuperando o objeto e convertendo de volta para um objeto JavaScript
```

```
let alunoRecuperado = JSON.parse(localStorage.getItem('aluno'));
```

```
console.log(alunoRecuperado.nome); // Ana
```

Trabalhando com Objetos no **LocalStorage**

```
//registrar dados
localStorage.setItem('nome', 'João');
localStorage.setItem('idade', '30');

// recuperar dados
let nome = localStorage.getItem('nome');
let idade = localStorage.getItem('idade');
console.log(nome); // João
console.log(idade); // 30

//limpar dados
localStorage.clear(); // Remove todos os dados armazenados
```


Desafio 2

- Desafio avançado: implemente uma mini aplicação de notas, onde os usuários possam adicionar, editar e remover notas (lembretes), que serão salvas no LocalStorage.

criado elementos HTML com JavaScript



Obrigado

