JavaScript

Programa do curso

- Aula 1: introdução e objetos
- **Aula 2: JavaScript integrado a HTML**
- Aula 3: DOM, seletores e elementos
- Aula 4: Eventos
- Aula 5: Formulários
- Aula 6: Ajax
- Aula 7: Exercício Integrador

1. AJAX

JSON

JSON

O formato de dados de JavaScript Object Notation, ou JSON para abreviar, é derivado dos literais da linguagem de programação JavaScript. Esse formato de troca de dados é um subconjunto da linguagem JavaScript, e não uma linguagem de programação.

Sua estrutura é muito parecida a um objeto literal de JavaScript, com algumas diferenças. Os nomes das propriedades são escritos entre aspas duplas, assim como os valores de strings.

```
let objetoEmFormatoJSON = '{ "atributo": "valor", "atributo1": 1, "atributo2": [], "atributo3": null, "atributo4": false }';
```

JSON

JavaScript tem um objeto JSON com os métodos stringify() e parse().

Stringify: O método stringfy permite passar um objeto ou valor de JavaScript para o formato JSON.

Parse: O método parse pega uma cadeia de caracteres em formato JSON e transforma em um objeto ou valor de JavaScript.

Graças a esses dois métodos, é possível utilizar o formato JSON para a troca de dados em formato de texto.

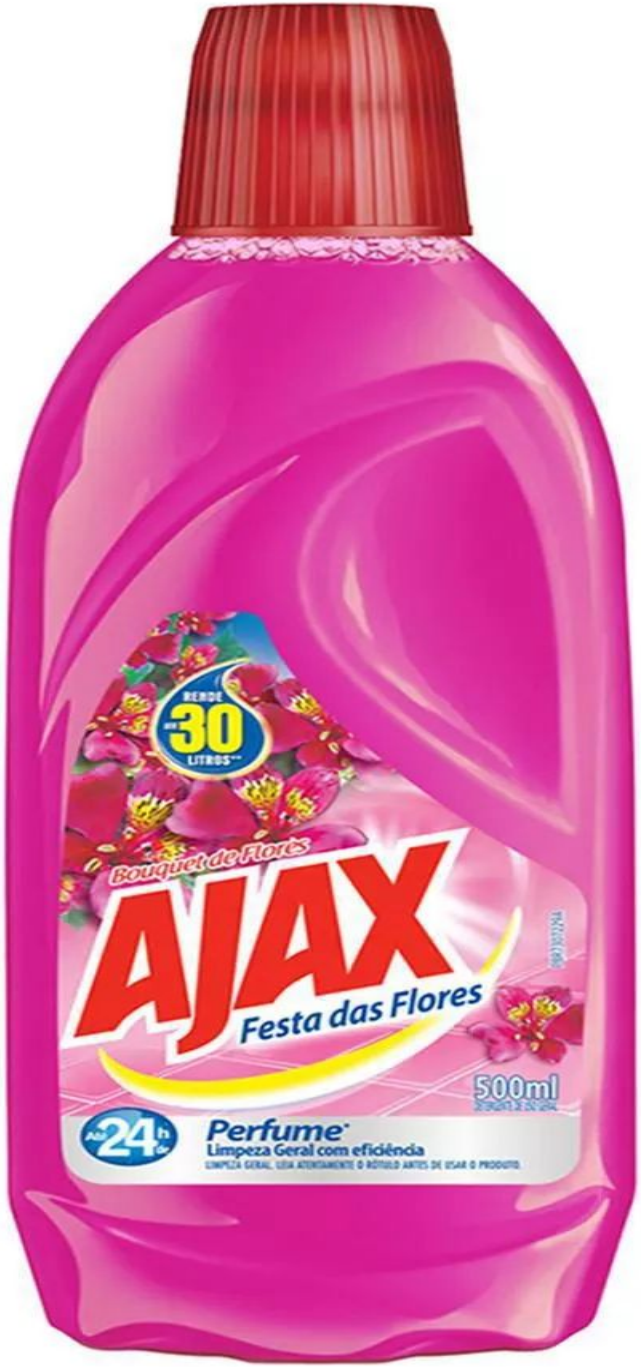
JSON

É possível acessar os atributos de uma variável em formato JSON da seguinte maneira:

```
let objetoJSON = JSON.parse( "alunos": [{“nome”:”Kleber”},{“nome”:”Klebinho”}]’);

objetoJSON.alunos[1].nome
```

AJAX


RENDE ATÉ 30 LITROS**
Bouquet de Flores
AJAX
Festa das Flores
500ml
Até 24h de
Perfume
Limpeza Geral com eficiência

AJAX

Ajax não é uma tecnologia, mas sim um termo implementado por Jesse James Garrett em 2005.

Ajax significa Asynchronous JavaScript and XML e se transformou em sinônimo do desenvolvimento moderno de front-end por um bom motivo. Essa técnica oferece a possibilidade de iniciar solicitações HTTP, como GET e POST, sem precisar sair da página web atual.

Por isso, podemos dizer que Ajax é uma técnica para criar páginas dinâmicas.

AJAX

AJAX permite:

- Atualizar o conteúdo de uma página sem recarregar
- Pedir informações a um servidor
- Receber informações de um servidor
- Enviar informações a um servidor

AJAX - Passos

```
let xmlhttp = new XMLHttpRequest();
```

AJAX - let xmlhttp = new XMLHttpRequest();

O único método global da interface XMLHttpRequest é o do construtor que, quando invocado, retorna ao aplicativo uma nova instância do objeto XMLHttpRequest. Através da interface herdada por esse objeto, vamos iniciar e gerenciar solicitações.

Como a solicitação HTTP é produzida de forma assíncrona, é necessário que o aplicativo seja notificado sobre qualquer mudança de estado enquanto a solicitação estiver vigente, por exemplo se a resposta tiver sido fornecida ou se a conexão tiver expirado.

Essa é uma forma fácil de obter informações de uma URL sem precisar recarregar toda a página. XMLHttpRequest é muito usado na programação AJAX.

AJAX - Passos

```
let xmlhttp = new XMLHttpRequest();

xmlhttp.onreadystatechange = function() {
    if (xmlhttp.readyState == 4 && xmlhttp.status == 200) {
        console.log(xmlhttp.responseText);
        //Meu código
    }
};
```

xmlhttp.onreadystatechange

Uma função do objeto JavaScript invocado quando o atributo readyState muda. O callback é invocado a partir da interface do usuário.

xmlhttp.readyState

0 se não se inicializou, 1 se está carregando, 2 se a solicitação já foi enviada, 3 se a resposta está sendo baixada e 4 se terminou

xmlhttp.status

O status da resposta ao pedido. Este é o código HTTPresult (por exemplo, o status é 200 no caso de uma solicitação bem-sucedida). Somente leitura.

xmlhttp.responseText

A resposta ao pedido como texto, ou null se o pedido não teve sucesso ou se ainda não foi enviado. Somente leitura.

AJAX - Passos

```
let xmlhttp = new XMLHttpRequest();

xmlhttp.onreadystatechange = function() {
   if (xmlhttp.readyState == 4 && xmlhttp.status == 200) {
          console.log(xmlhttp.responseText);
          //Meu código
   }
};

xmlhttp.open("GET", "url", true);
```

AJAX - xmlhttp.open("GET", "url", true);

Inicializa o pedido. Este método deve ser usado a partir do código JavaScript.

method
O método HTTP que será usado: "POST" ou "GET". É ignorado para URLs que não são de HTTP.

url
A URL para a qual o pedido é enviado.

async
Um parâmetro opcional, booleano, que por padrão é true. Indica se a operação é realizada de forma assíncrona.

AJAX - Passos

```
let  xmlhttp = new XMLHttpRequest();

xmlhttp.onreadystatechange = function() {
if (xmlhttp.readyState == 4 && xmlhttp.status == 200) {
        console.log(xmlhttp.responseText);
        //Meu código
        }
};

xmlhttp.open("GET", "url", true);
xmlhttp.send();
```

AJAX - Parâmetros

```
xmlhttp.onreadystatechange = function() {
if (xmlhttp.readyState == 4 && xmlhttp.status == 200) {
        console.log(xmlhttp.responseText);
        //Meu código
        }
};
xhttp.open("POST", "url", true);

xhttp.setRequestHeader("Content-type",
"application/x-www-form-urlencoded");

let params = “nomes=João&sobrenomes=Silva”

xhttp.send(params);
```

Vamos praticar!

Prática 1– **JSON e AJAX**

Obrigado!

Perguntas?