

APOSTILA LISTAS

JAVASCRIPT

Introdução

As listas são uma das estruturas de dados mais utilizadas na programação. Elas são úteis para armazenar e manipular coleções de dados de maneira organizada e eficiente. Em JavaScript, as listas podem ser implementadas de várias maneiras, mas as duas mais comuns são arrays e objetos.

Listas com Arrays em JavaScript

Os arrays em JavaScript são objetos que permitem armazenar vários valores em uma única variável. Eles podem conter diferentes tipos de dados, como strings, números e objetos. Os arrays são indexados, o que significa que cada elemento é identificado por um índice numérico, começando em zero.

Criação de uma lista com array

Para criar um array em JavaScript, podemos utilizar a seguinte sintaxe:

```
1 const myArray = [];
```

Este código cria um novo array vazio, que pode ser preenchido com elementos posteriormente. Para criar um array com elementos iniciais, podemos usar a seguinte sintaxe:

```
1 const myArray = [element1, element2, element3];
```

Acessando elementos de uma lista com array

Para acessar um elemento específico de um array, podemos utilizar o índice numérico correspondente ao elemento desejado. Por exemplo, para acessar o segundo elemento de um array chamado myArray, podemos usar a seguinte sintaxe:

```
1 const myArray = [element1, element2, element3];  
2 const secondElement = myArray[1];
```

Nesse exemplo, a variável secondElement receberá o valor do segundo elemento do array, que é o valor correspondente ao índice 1.

Adicionando elementos a uma fila com array

Para adicionar um novo elemento ao final de um array em JavaScript, podemos utilizar o método **push()**, também utilizado em pilhas e filas. Por exemplo:

```
1 const myArray = [element1, element2, element3];  
2 myArray.push(element4);
```

Nesse exemplo, o método push() adiciona o valor element4 ao final do array myArray.

Removendo elementos de uma lista com array

Para remover o último elemento de um array em JavaScript, podemos utilizar o método **pop()** ou **shift()**, sendo, o pop() normalmente mais utilizado. Por exemplo:

```
1 const myArray = [element1, element2, element3];  
2 myArray.pop();
```

Nesse exemplo, o método pop() remove o último elemento do array myArray.

Ordenando elementos de um array

Para ordenar os elementos de um array em JavaScript, podemos utilizar o método sort(). Por exemplo:

```
1 const myArray = [element3, element1, element2];  
2 myArray.sort();
```

Nesse exemplo, o método **sort()** ordena os elementos do array myArray em ordem alfabética ou numérica, dependendo do tipo de dados.

Conclusão

As listas em JavaScript são uma estrutura de dados essencial para a programação. Elas permitem armazenar e manipular coleções de dados de maneira organizada e eficiente. Nesta apostila, vimos como criar e manipular listas utilizando arrays em JavaScript. Com essas habilidades básicas, é possível criar programas complexos e eficientes que manipulam dados de maneira sofisticada.