Aula 20

Dando Vida aos Arrays

Revisão da Aula 11: Introdução aos Arrays (Vetores)

- **O que são Arrays?**
  - São listas ou coleções de dados armazenadas em uma única variável.
  - Criamos arrays usando colchetes `[]`.
- **A Regra de Ouro do Índice:**
  - Acessamos os elementos pela sua posição (índice).
  - A contagem de índices **SEMPRE** começa em **ZERO**.
  - Para acessar: `meuArray[0]` pega o primeiro item.
- **Propriedade `.length`:**
  - `meuArray.length` nos informa quantos elementos existem na lista.
  - Vimos a fórmula para pegar o último item: `meuArray[meuArray.length - 1]`.

Slide 2/4: Dando Vida aos Arrays

Aula 12: Manipulando e Percorrendo Arrays

- **Objetivo de Hoje:**
  - Aprender a adicionar e remover itens de um array de forma dinâmica.
  - Conectar o poder dos laços `for` com os arrays para automatizar tarefas.
- **Manipulando as Pontas do Array:**
  - **Para o FINAL da lista:**
    - `push()`: Adiciona um item.
    - `pop()`: Remove o último item.
  - **Para o INÍCIO da lista:**
    - `unshift()`: Adiciona um item.
    - `shift()`: Remove o primeiro item.

Slide 3/4: A Combinação Perfeita: `for` + Array

Percorrendo (Iterando) um Array Automaticamente

- **O Desafio:**
  - Como fazer o computador passar por **todos** os itens de um array, um por um, sem escrever `console.log(array[0])`, `console.log(array[1])`, etc.?
- **A "Fórmula Mágica":**
  - Usamos o laço `for` com a propriedade `.length`.

JavaScript

```
for (let i = 0; i < meuArray.length; i++) {
  // O código aqui dentro vai repetir para cada item!
}
```

- **A Lógica:** A variável `i` do laço `for` servirá como o **índice** para acessar cada elemento do array em sequência, do primeiro ao último.

Slide 4/4: Acessando Elementos no Laço

Como o `for` e o Array Trabalham Juntos

- **O Processo:**

JavaScript

```
let nomes = ["Ana", "Bia", "Carlos"];
// O laço vai de i = 0 até i < 3 (ou seja, i = 0, 1, 2)
for (let i = 0; i < nomes.length; i++) {
    // Na 1ª volta (i=0), acessamos nomes[0] -> "Ana"
    // Na 2ª volta (i=1), acessamos nomes[1] -> "Bia"
    // Na 3ª volta (i=2), acessamos nomes[2] -> "Carlos"
    let alunoAtual = nomes[i];
    console.log("Olá, " + alunoAtual);
}
```

- **Meta da Aula:**
  - Ser capaz de criar programas que processam listas de dados, não importa o tamanho, de forma totalmente automática.

Prática

JS script.js > ...

```
// ================================================================
// 1. MANIPULANDO O FINAL DO ARRAY: `push` e `pop`
// ================================================================

let frutas = ['Maçã', 'Banana']
console.log('Array inicial:', frutas)

// 1.1. .push(): Adiciona um ou mais elementos ao FINAL do array.
frutas.push('Laranja')
console.log("Depois do push('Laranja'):", frutas) // ["Maçã", "Banana", "Laranja"]

frutas.push('Uva', 'Morango')
console.log("Depois de push('Uva', 'Morango'):", frutas)

// 1.2. .pop(): Remove o ÚLTIMO elemento do array e o retorna.
let frutaRemovida = frutas.pop() // Remove "Morango"
console.log('Fruta removida com pop():', frutaRemovida)
console.log('Array depois do pop():', frutas) // ["Maçã", "Banana", "Laranja", "Uva"]

```

```
// ============================================================
// 2. MANIPULANDO O INÍCIO DO ARRAY: `unshift` e `shift`
// ============================================================

let filaAtendimento = ['Cliente A', 'Cliente B', 'Cliente C']
console.log('\nFila inicial:', filaAtendimento)

// 2.1. .unshift(): Adiciona um ou mais elementos ao INÍCIO do array.
filaAtendimento.unshift('Cliente VIP')
console.log("Depois do unshift('Cliente VIP'):", filaAtendimento)

// 2.2. .shift(): Remove o PRIMEIRO elemento do array e o retorna.
let proximoASerAtendido = filaAtendimento.shift() // Remove "Cliente VIP"
console.log('Próximo a ser atendido com shift():', proximoASerAtendido)
console.log('Fila depois do shift():', filaAtendimento) // ["Cliente A", "Cliente B", "Cliente
C"]

```

```
// ============================================================
// 3. PERCORRENDO (ITERANDO) UM ARRAY COM `for`
// ============================================================
// Esta é a forma clássica de passar por todos os elementos de um array.

let notas = [8.5, 9.0, 7.2, 10, 6.7]
console.log('\nPercorrendo o array de notas...')

// O laço `for` vai começar o `i` em 0 (primeiro índice) e vai até `i` ser
// menor que o tamanho do array (`notas.length`).
for (let i = 0; i < notas.length; i++) {
  // A cada volta, `i` representa o índice atual.
  // `notas[i]` acessa o elemento naquele índice.
  console.log('Na posição', i, 'temos a nota:', notas[i])
}

```

```
// ============================================================
// 4. EXEMPLO PRÁTICO: CALCULANDO A MÉDIA DE FORMA AUTOMÁTICA
// ============================================================
// Vamos usar o `for` para somar todas as notas de um array, não importa o tamanho.

let notasAluno = [7, 8, 5, 10, 9]
let soma = 0 // Acumulador começa em 0

console.log('\nCalculando a média do aluno...')

for (let i = 0; i < notasAluno.length; i++) {
  soma += notasAluno[i] // soma = soma + notasAluno[i]
  console.log('Somando', notasAluno[i], '-> Soma parcial:', soma)
}

let media = soma / notasAluno.length
console.log('O array de notas é:', notasAluno)
console.log('A soma total das notas é:', soma)
console.log('A média final do aluno é:', media)

console.log(
  'Fim da Aula 12! Agora podemos processar listas de qualquer tamanho!'
)
// Próxima aula: Vamos aprender a criar nossos próprios blocos de código reutilizáveis com
Funções!
```

ATIVIDADE

Faça cada exercício em um JS diferente

1. **Montando uma Playlist:**

   Crie um array vazio chamado `playlist`. Use `push()` para adicionar 3 das suas músicas favoritas. Depois, use `unshift()` para adicionar uma música no início da lista. Exiba a playlist final no console.

Faça cada exercício em um JS diferente

2. **Fila de Supermercado:**

   Comece com o array `fila = ["Maria", "João", "Ana"]`.

   1. Use `push()` para adicionar "Carlos" ao final da fila.
   2. Use `shift()` para remover a primeira pessoa ("Maria") e exiba no console "Maria foi atendida.".
   3. Exiba a fila atualizada.

Faça cada exercício em um JS diferente

3. **Listando os Números:**

   Crie um array com os números de 1 a 5: `[1, 2, 3, 4, 5]`. Use um laço `for` para exibir cada número no console com a mensagem: "O número é [número]".

Faça cada exercício em um JS diferente

4. **Soma Total:**

   Crie um array de números (ex: `[10, 20, 30, 40, 50]`). Use um laço `for` para calcular a **soma** de todos os números do array. Exiba o resultado final no console.

Faça cada exercício em um JS diferente

Desafio 1: Encontrando Pares e Ímpares

1. Crie um array com vários números inteiros (ex: `[3, 8, 12, 5, 6, 10, 7, 2, 9, 14]`).
2. Crie dois arrays vazios: `numerosPares` e `numerosImpares`.
3. Use um laço `for` para percorrer o array original de números.
4. Dentro do laço, use um `if` para verificar se o número atual é par (`numero % 2 === 0`).
    - Se for par, adicione-o ao array `numerosPares` com `push()`.
    - Se for ímpar, adicione-o ao array `numerosImpares` com `push()`.
5. No final, exiba os três arrays no console: o original, o de pares e o de ímpares.

Faça cada exercício em um JS diferente

Desafio 2: Buscador de Nomes

1. Crie um array com uma lista de nomes de pelo menos 5 pessoas.
2. Peça ao usuário para digitar um nome que ele deseja buscar na lista.
3. Crie uma variável booleana `encontrado = false;`.
4. Use um laço `for` para percorrer a lista de nomes.
5. Dentro do laço, use um `if` para comparar o nome digitado pelo usuário com o nome atual do array (`nomes[i]`).
6. Se encontrar o nome:
   - Altere a variável `encontrado` para `true`.
   - Use a palavra-chave `break` para interromper o laço, já que não precisamos procurar mais.
7. Após o laço, use um `if/else` para verificar o valor da variável `encontrado` e exibir um `alert`: "Nome encontrado!" ou "Nome não encontrado.".

Faça cada exercício em um JS diferente