

Aula 17

laço for

Slide 1/4: Revisão da Aula 9

Revisão da Aula 9: O Laço `do...while`

- **A Estrutura** `do...while` :
 - Aprendemos um laço de repetição onde o bloco de código é executado **primeiro** e a condição é testada **depois**.
- **Garantia de Execução:**
 - A principal característica do `do...while` é que ele **sempre executa o código pelo menos uma vez**, mesmo que a condição inicial seja falsa.
- **Caso de Uso Principal:**
 - Ideal para validar entradas do usuário, garantindo que ele forneça um dado válido antes de o programa continuar. Por exemplo, pedir uma senha até que ela seja a correta.

Slide 2/4: O Laço Mais Comum

Aula 10: O Laço de Repetição `for` (Para)

- **Objetivo de Hoje:**
 - Aprender o laço de repetição mais estruturado e frequentemente usado: o laço `for`.
- **Quando Usar o `for` ?**
 - É a escolha perfeita quando sabemos **exatamente quantas vezes** queremos que um bloco de código se repita.
 - É o equivalente direto do `PARA` do VisualG.
- **A Vantagem do `for` :**
 - Ele concentra toda a lógica de controle da repetição (início, condição de parada e passo) em uma única linha, tornando o código mais limpo e organizado.

Entendendo as Três Partes Essenciais

- **Sintaxe Completa:**

JavaScript



```
for ( inicialização ; condição ; incremento ) {  
    // Bloco de código a ser repetido  
}
```

- **As 3 Etapas:**

1. **Inicialização:** Onde o contador começa (ex: `let i = 0`). Executada **apenas uma vez**, no início.
2. **Condição:** O teste que decide se o laço continua (ex: `i < 10`). Verificada **antes** de cada repetição.
3. **Incremento:** O que acontece ao **final** de cada repetição (ex: `i++`).

Slide 4/4: Flexibilidade e Exemplos

Variações e Aplicações do Laço `for`

- **Não é só para contar para cima:**
 - **Contagem Regressiva:** Podemos fazer o contador diminuir (`for (let i = 10; i >= 0; i--)`).
 - **Passos Diferentes:** Podemos pular números, contando de 2 em 2, 5 em 5, etc. (`for (let i = 0; i <= 20; i += 2)`).
- **Casos de Uso Práticos:**
 - Exibir uma tabuada.
 - Repetir uma ação um número fixo de vezes (ex: "Enviar 5 e-mails").
 - Percorrer estruturas de dados (veremos isso em breve com Arrays!).
- **Meta da Aula:**
 - Dominar o laço `for` para executar tarefas repetitivas de forma controlada, previsível e elegante.

Prática

JS script.js > ...

```
1 // =====
2 // 1. A SINTAXE DO LAÇO `for`
3 // =====
4 // A estrutura `for` agrupa a lógica de controle do laço em uma única linha.
5
6 console.log("--- Exemplo 1: Contando de 0 a 4 ---");
7 // for (INICIALIZAÇÃO; CONDIÇÃO; INCREMENTO)
8 for (let i = 0; i < 5; i++) {
9     // 1. `let i = 0`: A variável `i` (de índice ou iteração) é criada e começa em 0. (Executa 1 vez)
10    // 2. `i < 5`: A condição é verificada. Se for `true`, o código do bloco é executado.
11    // 3. O código dentro do bloco é executado.
12    // 4. `i++`: O incremento é executado no final da iteração.
13    // 5. Volta para o passo 2.
14    console.log("O valor de i nesta volta é:", i);
15 }
16
```

```
17 // =====
18 // 2. COMPARAÇÃO: `for` vs `while`
19 // =====
20 // A mesma tarefa de contar de 1 a 3, feita das duas formas.
21
22 console.log("\n--- Exemplo 2: `for` vs `while` ---");
23
24 // Com `while`:
25 console.log("Contagem com `while`:");
26 let contadorWhile = 1; // Inicialização fora
27 while (contadorWhile <= 3) { // Condição
28     console.log(contadorWhile);
29     contadorWhile++; // Incremento dentro
30 }
31
32 // Com `for`:
33 console.log("Contagem com `for`:");
34 for (let i = 1; i <= 3; i++) { // Inicialização, condição e incremento juntos!
35     console.log(i);
36 }
37 // Note como o `for` é mais compacto e organizado para este tipo de tarefa.
```



```
40 // 3. VARIAÇÕES DO LAÇO `for`
41 // =====
42 // O `for` é muito flexível.
43
44 console.log("\n--- Exemplo 3: Variações ---");
45
46 // 3.1. Contagem Regressiva:
47 console.log("Contagem regressiva:");
48 for (let i = 5; i >= 1; i--) {
49 |   console.log(i);
50 }
51
52 // 3.2. Passos maiores (pulando de 2 em 2):
53 console.log("Contando os pares de 0 a 10:");
54 for (let i = 0; i <= 10; i += 2) {
55 |   console.log(i);
56 }
57
```

```
58 // =====
59 // 4. EXEMPLO PRÁTICO: TABUADA
60 // =====
61 // Um uso clássico e muito útil do laço `for`.
62
63 console.log("\n--- Exemplo 4: Tabuada ---");
64 let numeroTabuada = prompt("Qual tabuada você quer ver (de 1 a 10)?");
65 numeroTabuada = Number(numeroTabuada);
66
67 console.log("Calculando a tabuada do", numeroTabuada);
68
69 for (let multiplicador = 1; multiplicador <= 10; multiplicador++) {
70     let resultado = numeroTabuada * multiplicador;
71     console.log(numeroTabuada + " x " + multiplicador + " = " + resultado);
72 }
73
74 console.log("Fim da Aula 10! O laço `for` é uma ferramenta poderosa para repetições controladas!");
75 // Próxima aula: Vamos começar a agrupar dados com Vetores (Arrays)!
```

ATIVIDADE

**Faça cada exercício em um JS
diferente**

1. Série de Números:

Escreva um `for` loop que exiba no console os números de 1 a 15.

Faça cada exercício em um JS diferente

2. Múltiplos de 5:

Escreva um `for` loop que exiba no console todos os múltiplos de 5, de 5 até 50 (inclusive).
(Dica: use `i += 5` no incremento).

Faça cada exercício em um JS diferente

3. Contagem Regressiva:

Crie um `for` loop que faça uma contagem regressiva de 20 para 0. Após o loop, exiba a mensagem "Decolar!".

Faça cada exercício em um JS diferente

4. Repetidor de Frases:

Peça ao usuário uma frase e um número. Use um `for` loop para exibir a frase no console o número de vezes que o usuário indicou.

Faça cada exercício em um JS diferente

Use o exercício 3 e crie um sistema com interface utilizando IA – quero uma nave decolando e os controladores de funções da nave (Use sua criatividade)