

1. LOOPS EM JAVASCRIPT

Loops são estruturas de controle que permitem executar um bloco de código repetidamente enquanto uma condição específica for verdadeira. Eles são muito úteis para realizar tarefas repetitivas. Vamos explorar os três principais tipos de loops em JavaScript.

1.1. While

O loop while executa um bloco de código enquanto uma condição especificada for verdadeira. A estrutura básica é a seguinte:

```
while (condição) {  
    // Código a ser executado  
}
```

Exemplo no console:

```
let contador = 0;  
  
while (contador < 5) {  
    console.log(contador);  
    contador++;  
}
```

1.2. do while

O loop do...while é semelhante ao while, mas garante que o código seja executado pelo menos uma vez, mesmo se a condição for falsa. A estrutura é a seguinte:

```
do {  
    // Código a ser executado  
} while (condição);
```

Exemplo no console:

```
let contador = 0;

do {
  console.log(contador);
  contador++;
} while (contador < 5);
```

1.3. for

O loop for é uma estrutura de controle de repetição com uma sintaxe mais concisa. É frequentemente usado quando você sabe exatamente quantas vezes deseja que o loop seja executado. A estrutura básica é a seguinte:

```
for (inicialização; condição; incremento) {
  // Código a ser executado
}
```

Exemplo no console:

```
for (let i = 0; i < 5; i++) {
  console.log(i);
}
```

2. LOOPS EM JAVASCRIPT DENTRO DE UM DOCUMENTO HTML

Segue abaixo exemplos de como você pode usar loops em JavaScript dentro de um documento HTML.

2.1. FOR

```
<!DOCTYPE html>
<html>
<head>
  <title>Exemplo de Loops em JavaScript</title>
</head>
<body>
  <button onclick="exibirContagem()">Clique para exibir
a contagem</button>
  <div id="contagem"></div>

  <script>
    function exibirContagem() {
      let contagemDiv =
document.getElementById("contagem");
      contagemDiv.innerHTML = ""; // Limpa o conteúdo
anterior

      for (let i = 1; i <= 10; i++) {
        contagemDiv.innerHTML += i + "<br>";
      }
    }
  </script>
</body>
</html>
```

Neste exemplo, quando você clica no botão, a função `exibirContagem()` é chamada, e um for loop é usado para preencher a `<div>` com os números de 1 a 10.

2.2. WHILE

```
<!DOCTYPE html>
<html>
<head>
```

```

<title>Exemplo de Loop While em JavaScript</title>
</head>
<body>
  <button onclick="iniciarContagemRegressiva()">Iniciar
Contagem Regressiva</button>
  <div id="contagem"></div>

  <script>
    function iniciarContagemRegressiva() {
      let contagemDiv =
document.getElementById("contagem");
      contagemDiv.innerHTML = ""; // Limpa o conteúdo
anterior

      let contador = 5; // Inicializa o contador com 5

      while (contador >= 1) {
        contagemDiv.innerHTML += contador + "<br>";
        contador--; // Decrementa o contador em cada
iteração
      }
    }
  </script>
</body>
</html>

```

Neste exemplo, quando você clica no botão "Iniciar Contagem Regressiva", a função `iniciarContagemRegressiva()` é chamada. Dentro desta função, utilizamos um loop `while` para exibir os números de 5 a 1 na `<div>` com a ID "contagem".

2.3. DO WHILE

```

<!DOCTYPE html>

```

```
<html>
<head>
  <title>Exemplo do...while em JavaScript</title>
</head>
<body>
  <h1>Contagem de 1 a 5</h1>

  <p>Números:</p>

  <p id="resultado"></p>

  <script>
    var numero = 1;
    var resultado = "";

    do {
      resultado += numero + " ";
      numero++;
    } while (numero <= 5);

    document.getElementById("resultado").textContent =
    resultado;
  </script>
</body>
</html>
```

Neste exemplo, o loop do...while inicia com número igual a 1 e continua incrementando-o até que seja maior do que 5. A cada iteração, o número é adicionado à variável resultado, que é exibida na página quando o loop termina. O resultado será uma lista de números de 1 a 5.