

RESUMO DA AULA

Parâmetro REST no JavaScript

7 - Par metro REST O par metro Rest (...) é uma funcionalidade introduzida no ES6 que permite que uma função receba um número indefinido de argumentos e os trate como um array dentro da função. Ele é particularmente útil quando não sabemos de antemão quantos argumentos uma função vai receber ou quando queremos manipular vários valores passados como parâmetros de uma só vez. Apesar de o Rest usar a mesma sintaxe do operador Spread (...), eles têm funcionalidades diferentes. Enquanto o Spread espalha elementos, o Rest coleta os elementos em um array. Vamos entender na prática como usar o par metro Rest e por que ele é tão útil! Sintaxe do Par metro Rest A sintaxe para usar o Rest é bem simples: basta adicionar três pontos (...) antes do nome do parâmetro. Por exemplo: `function exemploRest(param1, param2, ...theArgs) { console.log(param1); // Exibe o primeiro parâmetro console.log(param2); // Exibe o segundo parâmetro console.log(theArgs); // Exibe todos os demais parâmetros como um array }` Nesse exemplo, `theArgs` será um array contendo todos os parâmetros passados após `param2`. Exemplo Prático com Par metro Rest Vamos ver um exemplo prático. Imagine que queremos criar uma função que exiba uma mensagem de incentivo para vários desenvolvedores de uma vez: `function incentivarQuester(mensagem, ...nomesQuesters) { nomesQuesters.map(nome => console.log(`${mensagem} ${nome}!`)); } incentivarQuester('Parabéns por ter chegado ao m dulo de JS', 'Roberto', 'Ricardo', 'Raphael');` Nesse exemplo, o primeiro parâmetro é uma mensagem, que é utilizada como base. Os outros parâmetros ('Roberto', 'Ricardo' e 'Raphael') são passados para o parâmetro Rest `...nomesQuesters`, que os agrupa em um array. Dentro da função, usamos `map` para iterar sobre cada nome e exibir a mensagem para cada um deles. O resultado no console será:
Parabéns por ter chegado ao m dulo de JS Roberto! Parabéns por ter chegado ao m dulo de JS Ricardo! Parabéns por ter chegado ao m dulo de JS Raphael! Detalhes Importantes do Par metro Rest O Par metro Rest deve ser o último parâmetro de uma função. Como ele coleta todos os parâmetros

metros restantes, ele sempre deve estar na última posição. Caso contrário, você terá um erro de sintaxe. // C digo correto: `function exemploCorreto(param1, param2, ...outrosParams) {}` // C digo incorreto (gerar um erro): `function exemploIncorreto(...outrosParams, param1) {}` O Rest cria um array mesmo se não houver parâmetros adicionais. Se a função for chamada apenas com os parâmetros iniciais (sem parâmetros extras), o Rest será um array vazio. `function mostrarParametros(param1, ...restParams) { console.log(restParams); // [] } mostrarParametros('Apenas um parâmetro');` // []

Compara o entre Rest e Spread Embora o Rest e o Spread utilizem a mesma sintaxe (...), a principal diferença entre eles é: Rest: Coleta múltiplos elementos e os coloca dentro de um array. Spread: Expande elementos de um array (ou objeto) em elementos individuais. No código, o contexto de uso define se estamos lidando com Rest ou Spread.

Uso Comum do Parâmetro Rest O parâmetro Rest é muito útil em funções que precisam lidar com múltiplos argumentos. Veja um exemplo clássico de somar vários números passados como argumentos: `function somarTodos(...numeros) { return numeros.reduce((total, numeroAtual) => total + numeroAtual, 0); } console.log(somarTodos(1, 2, 3, 4, 5)); // 15` Aqui, ...numeros coleta todos os argumentos passados para a função e os transforma em um array. Usamos o método reduce para somar todos os números do array.

Aplicação em Funções de Callbacks O Rest também é muito útil quando estamos lidando com funções de callback que recebem vários argumentos, mas você quer capturar apenas alguns: `function capturarPrincipais(param1, param2, ...outros) { console.log(`Os principais são: ${param1}, ${param2}`); console.log(`Os demais são: ${outros.join(', ')}`); } capturarPrincipais('Roberto', 'Ricardo', 'Raphael', 'Pedro', 'João');` // Saída: // Os principais são: Roberto, Ricardo // Os demais são: Raphael, Pedro, João

Nesse exemplo, param1 e param2 capturam os primeiros argumentos, e ...outros agrupa todos os argumentos restantes.

Resumo O Parâmetro Rest (...) é usado para capturar múltiplos argumentos e armazená-los em um array. Ele sempre deve estar no final dos parâmetros da função. É útil para funções que lidam com um número variável de argumentos, permitindo uma manipulação mais flexível e intuitiva.

Conclusão Nessa aula, vimos como o parâmetro Rest nos permite trabalhar com um número indefinido de argumentos em uma função de maneira prática e eficiente. Ele é um recurso indispensável do JavaScript moderno e

ser especialmente útil em funções complexas, onde precisamos manipular vários argumentos ao mesmo tempo. Na próxima aula, vamos explorar os métodos `setInterval` e `setTimeout`, que são essenciais para controlar o tempo de execução no JavaScript.
