

RESUMO DA AULA

Operador Spread no JavaScript

6 - Operador Spread O operador Spread (...) é uma das funcionalidades mais úteis e poderosas introduzidas no JavaScript moderno (ES6). Ele permite que a gente expanda os elementos de um objeto iterável (como um array, uma string ou até mesmo objetos) em elementos separados. Com o Spread, é possível manipular e combinar arrays e objetos de maneira muito mais intuitiva e simplificada. Vamos entender como o Spread funciona na prática! O que é o Operador Spread? O operador Spread é representado por três pontos (...). Quando usado em arrays ou objetos, ele espalha seus elementos, permitindo que sejam usados de maneira separada. Isso é especialmente útil para clonagem, combinação de arrays, criação de cópias de objetos, e muito mais. Como Usar o Operador Spread com Arrays Imagine que você tem um array simples: `var pessoas = ['Roberto', 'Ricardo', 'Raphael'];` Se dermos um `console.log` desse array, veremos todos os elementos juntos: `console.log(pessoas);` // ["Roberto", "Ricardo", "Raphael"] Mas, se aplicarmos o operador Spread, o que acontece? Vamos ver: `console.log(...pessoas);` // Roberto Ricardo Raphael Com o operador ..., o array foi expandido em elementos separados. Isso significa que cada valor foi espalhado individualmente. No caso acima, ele imprime cada nome do array separadamente. Exemplos de Uso do Spread com Arrays Um dos usos mais comuns do operador Spread é concatenar ou combinar arrays. Antes do ES6, tínhamos que usar o método `concat` para unir dois ou mais arrays: `var pessoas1 = ['Roberto', 'Ricardo', 'Raphael']; var pessoas2 = ['Pedro', 'João', 'Paulo']; var pessoasCombinadas = pessoas1.concat(pessoas2); console.log(pessoasCombinadas);` // ["Roberto", "Ricardo", "Raphael", "Pedro", "João", "Paulo"] Com o operador Spread, isso pode ser feito de maneira muito mais simples e legível: `var pessoasCombinadas = [...pessoas1, ...pessoas2]; console.log(pessoasCombinadas);` // ["Roberto", "Ricardo", "Raphael", "Pedro", "João", "Paulo"] Bastou usar os ... para expandir os elementos de `pessoas1` e `pessoas2` em um novo array, unindo-os de forma muito mais intuitiva. Clonagem de Objetos e Arrays com Spread O operador

Spread também é muito útil para clonar arrays ou objetos. Vamos ver como isso funciona: Clonando Arrays

```
var pessoasOriginal = ['Roberto', 'Ricardo', 'Raphael']; var pessoasClone = [...pessoasOriginal]; console.log(pessoasClone); // ["Roberto", "Ricardo", "Raphael"]
```

Nesse exemplo, `pessoasClone` é uma cópia independente de `pessoasOriginal`. Alterar `pessoasClone` não afeta `pessoasOriginal`.

Clonando Objetos O Spread também pode ser usado para clonar objetos de forma simples:

```
var pessoa1 = { nome: 'Roberto', idade: 33 }; var pessoaClonada = { ...pessoa1 }; console.log(pessoaClonada); // { nome: 'Roberto', idade: 33 }
```

Assim como no exemplo de arrays, `pessoaClonada` é uma cópia independente de `pessoa1`.

Combinando Objetos com Spread Outro uso interessante do Spread é combinar objetos. Por exemplo, podemos unir dois objetos diferentes em um só:

```
var pessoa1 = { nome: 'Roberto', idade: 33 }; var pessoa2 = { sobrenome: 'Dias', cidade: 'São Paulo' }; var pessoaCombinada = { ...pessoa1, ...pessoa2 }; console.log(pessoaCombinada); // { nome: 'Roberto', idade: 33, sobrenome: 'Dias', cidade: 'São Paulo' }
```

Aqui, combinamos `pessoa1` e `pessoa2` em um novo objeto `pessoaCombinada`, com todas as propriedades dos dois objetos originais.

Passando Parâmetros para Funções O Spread também pode ser usado para passar os elementos de um array como parâmetros para uma função. Antes do ES6, tínhamos que usar o método `apply` para isso:

```
function imprimirPessoa(nome, idade) { console.log(`Nome: ${nome}, Idade: ${idade}`); } imprimirPessoa.apply(null, ['Roberto', 33]); // Nome: Roberto, Idade: 33
```

Mas, com o Spread, podemos fazer isso de forma muito mais simples:

```
imprimirPessoa(...['Roberto', 33]); // Nome: Roberto, Idade: 33
```

Aqui, `...['Roberto', 33]` expande o array e envia os elementos como argumentos separados para a função `imprimirPessoa`.

Combinação de Objetos com Prioridade Quando combinamos objetos, o Spread segue uma ordem de prioridade. Se houver propriedades iguais, as propriedades do objeto mais à direita no Spread sobrescrevem as anteriores:

```
var pessoa1 = { nome: 'Roberto', idade: 33 }; var pessoa2 = { nome: 'Ricardo', cidade: 'São Paulo' }; var pessoaCombinada = { ...pessoa1, ...pessoa2 }; console.log(pessoaCombinada); // { nome: 'Ricardo', idade: 33, cidade: 'São Paulo' }
```

No exemplo acima, a propriedade `nome` foi sobrescrita pela segunda nome do objeto `pessoa2`. Por isso, `pessoaCombinada` tem nome: 'Ricardo' em vez de nome: 'Roberto'.

Conclusão O operador Spread é uma funcionalidade poderosa que simplifica a manipulação de arrays e objetos no

JavaScript. Ele facilita a concatenação, clonagem, combinação e até o uso de elementos de arrays como parâmetros de função. No próximo tópico, vamos aprender sobre outro operador interessante e complementar ao Spread: o Operador Rest, que tem algumas diferenças importantes e se aplica em outros contextos.
