

RECURSIVIDADE (PARTE 2/2)

Como dito anteriormente, a recursividade é uma técnica de programação em que uma função chama a si mesma para resolver um problema. Ela é útil em diversas situações, como na realização de operações em arrays. Neste tutorial, iremos explorar a utilização da recursividade em operações em arrays, utilizando a linguagem de programação Javascript.

Recursividade em arrays

Uma das principais aplicações da recursividade em arrays é a realização de operações que envolvem a manipulação de cada elemento do array. Por exemplo, suponha que temos um array de números e desejamos calcular a soma de todos eles. Podemos utilizar a recursividade para percorrer o array e somar cada elemento com o resultado obtido na recursão anterior.

Exemplos de código fonte

A seguir, apresentamos alguns exemplos de códigos fonte em Javascript que utilizam recursividade para realizar operações em arrays.

Soma de elementos de um array

O código a seguir mostra uma função recursiva que calcula a soma de todos os elementos de um array:

```
1 function somaArray(arr) {  
2   if (arr.length === 0) {  
3     return 0;  
4   } else {  
5     return arr[0] + somaArray(arr.slice(1));  
6   }  
7 }  
8  
9 // Exemplo de uso:  
10 const arr = [1, 2, 3, 4, 5];  
11 const soma = somaArray(arr);  
12 console.log(soma); // Output: 15
```

Neste exemplo, a função `somaArray` recebe um array como parâmetro. Se o array estiver vazio, a função retorna 0. Caso contrário, a função retorna a soma do primeiro elemento do array com o resultado da chamada recursiva da função passando como parâmetro o array sem o primeiro elemento (utilizando o método `slice`).

Cálculo da média de elementos de um array

O código a seguir mostra uma função recursiva que calcula a média dos elementos de um array:

```
1 function mediaArray(arr) {
2   if (arr.length === 0) {
3     return null;
4   } else if (arr.length === 1) {
5     return arr[0];
6   } else {
7     const primeiro = arr[0];
8     const restante = arr.slice(1);
9     const mediaRestante = mediaArray(restante);
10    const tamanhoTotal = arr.length;
11    const mediaTotal = (primeiro + mediaRestante * (tamanhoTotal - 1)) / tamanhoTotal;
12    return mediaTotal;
13  }
14 }
15
16 // Exemplo de uso:
17 const arr = [1, 2, 3, 4, 5];
18 const media = mediaArray(arr);
19 console.log(media); // Output: 3
```

Neste exemplo, a função `mediaArray` recebe um array como parâmetro. Se o array estiver vazio, a função retorna `null`. Se o array possuir apenas um elemento, a função retorna o próprio elemento. Caso contrário, a função realiza a chamada recursiva passando como parâmetro o array sem o primeiro elemento. Em seguida, a função calcula a média dos elementos restantes do array e utiliza essa média e o primeiro elemento para calcular a média total do array.

Conclusão

A recursividade é uma técnica poderosa que pode ser utilizada em diversas situações, incluindo operações em arrays. Neste tutorial, vimos exemplos de como utilizar a recursividade para realizar operações como soma, busca em profundidade e cálculo de média em arrays utilizando a linguagem de programação Javascript. Com esses exemplos, esperamos que você tenha adquirido mais conhecimento sobre o assunto e possa aplicá-lo em seus próprios projetos.