

13 - Desestruturação

O que é Desestruturação?

Desestruturação é um recurso do ES6 que permite extrair dados de arrays ou propriedades de objetos em variáveis distintas de forma prática e direta. É como desempacotar uma caixa e colocar cada item em seu próprio lugar.

Desestruturação de Arrays

```
let numeros = [1, 2, 3];  
let [primeiro, segundo, terceiro] = numeros;  
console.log(primeiro, segundo, terceiro); // 1 2 3
```

Aqui, criamos um array `numeros` e, em seguida, usamos a desestruturação para criar três novas variáveis, cada uma recebendo um elemento do array. É uma maneira rápida de acessar itens individuais de um array.

Desestruturação de Objetos

```
let pessoa = { nome: 'João', idade: 25 };  
let { nome, idade } = pessoa;  
console.log(nome, idade); // João 25
```

Neste exemplo, extraímos as propriedades `nome` e `idade` do objeto `pessoa` para duas novas variáveis. Isso simplifica muito o acesso a propriedades específicas de um objeto.

Usos Práticos da Desestruturação

A desestruturação é extremamente útil em várias situações, como ao trabalhar com funções, arrays, ou quando lidamos com dados mais complexos.

Em Funções

Ao usar a desestruturação em parâmetros de função, tornamos o código mais limpo e direto, evitando a necessidade de acessar as propriedades do objeto dentro da função.

```
function exibirInfo({ nome, idade }) {  
  console.log(`Nome: ${nome}, Idade: ${idade}`);  
}  
  
exibirInfo(pessoa); // Nome: João, Idade: 25
```

Com Arrays Aninhados

```
let cores = ["vermelho", "azul", ["verde", "amarelo"]];  
let [primarias, secundarias] = cores;  
console.log(primarias, secundarias); // ["vermelho", "azul"] ["verde", "amarelo"]
```

Aqui, desestruturamos um array de arrays. Isso mostra como a desestruturação pode ser adaptada para estruturas de dados mais complexas.

Valores Padrão

```
let [a = 1, b = 2] = [3];  
console.log(a, b); // 3 2
```

Neste exemplo, definimos valores padrão para **a** e **b**. **a** recebe o valor do array, enquanto **b** usa o valor padrão, pois o array não tem um segundo elemento.

Trocando Valores de Variáveis

```
let x = 1, y = 2;  
[x, y] = [y, x];  
console.log(x, y); // 2 1
```

A desestruturação também pode ser usada para trocar valores entre variáveis de forma eficiente, sem a necessidade de uma variável temporária.

Exercícios de Desestruturação

1. **Extrair Dados de um Array:** Crie um array com três elementos e utilize a desestruturação para criar três variáveis, cada uma recebendo um elemento do array.
2. **Extrair Propriedades de um Objeto:** Crie um objeto com propriedades como **nome** e **idade**. Use a desestruturação para criar variáveis a partir dessas propriedades.
3. **Desestruturação em Funções:** Escreva uma função que recebe um objeto como parâmetro e use a desestruturação para trabalhar com as propriedades do objeto dentro da função.
4. **Desestruturação com Arrays Aninhados:** Crie um array de arrays e use a desestruturação para criar variáveis a partir dos elementos do array aninhado.
5. **Valores Padrão na Desestruturação:** Crie um array com um elemento e utilize a desestruturação para criar duas variáveis. A primeira deve receber o valor do array e a segunda deve usar um valor padrão.
6. **Trocar Valores de Variáveis com Desestruturação:** Crie duas variáveis e troque os valores entre elas utilizando desestruturação.