

8 - Estrutura de Dados Map

Map é uma daquelas ferramentas que podem fazer a diferença entre um código ok e um código top!

O que é um Map?

Map é uma coleção de pares chave-valor. Diferente dos objetos, no Map, as chaves podem ser de qualquer tipo - até objetos ou funções! Além disso, ele mantém a ordem de inserção, o que pode ser mega útil em algumas situações.

```
let meuMap = new Map();
```

Aqui, estamos criando um novo Map. Pensa nele como um super objeto, só que mais flexível e com superpoderes!

Adicionando Elementos:

Vamos adicionar alguns elementos ao nosso Map usando o método `set`.

```
meuMap.set('chave1', 'valor1');  
meuMap.set('chave2', 'valor2');
```

Nesse exemplo, adicionamos duas chaves ao Map: 'chave1' com o valor 'valor1' e 'chave2' com 'valor2'. O legal do Map é que ele aceita qualquer coisa como chave!

Acessando Elementos:

Agora, vamos ver como pegar valores de volta do nosso Map.

```
console.log(meuMap.get('chave1')); // valor1
```

Usamos o método `get` com a chave 'chave1' para pegar o valor associado. Aqui, ele retorna 'valor1'.

Iterando sobre um Map:

Maps são iteráveis, então podemos percorrer eles de várias maneiras.

```
for (let [chave, valor] of meuMap) {  
  console.log(`${chave}: ${valor}`);  
}
```

Neste loop `for...of`, desestruturamos cada par chave-valor e imprimimos eles. É uma maneira super prática de ver tudo o que tá no Map.

Métodos Úteis do Map:

Maps vêm com alguns métodos bem práticos.

```
console.log(meuMap.size);           // 2  
meuMap.delete('chave2');           // Remove a chave 'chave2'  
console.log(meuMap.has('chave2')); // false
```

- `size` nos mostra quantos pares chave-valor temos no Map.
- `delete` remove um par baseado na chave.
- `has` verifica se uma chave específica existe no Map.

Quando Usar um Map?

Map é perfeito quando você precisa de uma coleção de pares chave-valor, onde as chaves são únicas e podem ser de qualquer tipo. É uma escolha melhor que objetos quando as chaves são desconhecidas até o tempo de execução, ou quando todas as chaves são do mesmo tipo e os valores também.

Convertendo Map para Objetos e Arrays

Às vezes, você pode querer converter seu Map em um objeto ou array.

```
let mapParaObjeto = Object.fromEntries(meuMap);
console.log(mapParaObjeto); // { chave1: 'valor1' }

let chavesArray = [...meuMap.keys()];
console.log(chavesArray); // ['chave1']
```

- **Object.fromEntries** transforma o Map em um objeto.
- Usando spread (...), pegamos todas as chaves ou valores do Map e colocamos em um array.

Exercícios para Praticar:

1. Criar e Adicionar Pares Chave-Valor no Map: Crie um **Map** chamado **livros** e adicione nele três pares de chave-valor, onde a chave é o título do livro e o valor é o autor.
2. Acessar Valor no Map: Acesse o valor associado a uma das chaves no **Map livros** que você criou e imprima o nome do autor.
3. Iterar Sobre um Map: Use um loop **for...of** para iterar sobre o **Map livros** e imprimir todos os pares chave-valor.
4. Usando Métodos do Map: size, delete, has: No seu **Map livros**, use o método **size** para imprimir o número de livros, **delete** para remover um livro pelo título e **has** para verificar se um livro ainda está no **Map**.
5. Map com Tipos de Chaves Diferentes: Crie um **Map** chamado **colecacao** que contém diferentes tipos de chaves (como **string**, **number**, **object**) e seus respectivos valores.