

7- Objetos

Vamos falar sobre uma das estruturas mais poderosas e fundamentais em JavaScript: os Objetos. Se você quer mandar bem em backend com Node.js, precisa entender isso daqui. Vou explicar de um jeito simples, mas bem completo.

O que são Objetos?

No JavaScript, um objeto é uma coleção de propriedades, e uma propriedade é uma associação entre uma chave e um valor. O valor de uma propriedade pode ser uma função, que é então conhecida como um método do objeto.

Criação de Objeto:

```
let pessoa = {  
  nome: 'João',  
  idade: 25,  
  falar: function() {  
    console.log(`Oi, meu nome é ${this.nome}`);  
  }  
};
```

Aqui, criamos um objeto **pessoa** com três propriedades: **nome**, **idade** e **falar**. **falar** é um método, ou seja, uma função dentro de um objeto.

Acessando Propriedades e Métodos:

Para acessar as propriedades ou métodos de um objeto, usamos a notação de ponto ou colchetes.

Exemplo:

```
console.log(pessoa.nome); // Acessando a propriedade 'nome'  
pessoa.falar();           // Chamando o método 'falar'
```

Acessamos a propriedade **nome** do objeto **pessoa** e chamamos o método **falar**.

Alterando e Adicionando Propriedades

Você pode alterar as propriedades de um objeto ou adicionar novas propriedades a ele após sua criação.

```
pessoa.idade = 30;           // Alterando a idade
pessoa.sobrenome = 'Silva'; // Adicionando nova propriedade 'sobrenome'
```

Alteramos a propriedade `idade` e adicionamos uma nova propriedade `sobrenome` ao objeto `pessoa`.

Iterando sobre Propriedades

Podemos percorrer todas as propriedades de um objeto usando um loop `for...in`.

```
for (let chave in pessoa) {
  console.log(`${chave}: ${pessoa[chave]}`);
}
```

Com o loop `for...in`, passamos por cada propriedade do objeto `pessoa`, imprimindo seu nome e valor.

Objetos como Referências:

Em JavaScript, objetos são referenciados. Quando você atribui um objeto a uma variável, está passando uma referência a esse objeto.

Ou seja, se você faz uma “cópia” de um objeto, e altera alguma propriedade, o objeto original também será alterado.

```
let outraPessoa = pessoa;
outraPessoa.nome = 'Maria';
console.log(pessoa.nome); // Maria
```

`outraPessoa` é uma referência ao mesmo objeto que `pessoa`. Ao mudar `nome` através de `outraPessoa`, o objeto original `pessoa` também é alterado.

Métodos Avançados:

JavaScript oferece métodos avançados para trabalhar com objetos, como `Object.keys()`, `Object.values()`, e `Object.entries()`.

```
console.log(Object.keys(pessoa)); // ['nome', 'idade', 'falar', 'sobrenome']
```

`Object.keys(pessoa)` retorna um array com os nomes das propriedades do objeto `pessoa`.

Exercícios para Praticar Objetos:

1. Criar e Acessar um Objeto: Crie um objeto chamado `carro` com propriedades como `marca`, `modelo` e `ano`. Em seguida, acesse e imprima a propriedade `modelo` do objeto.
2. Adicionar e Modificar Propriedades de um Objeto: A partir do objeto `carro` criado, adicione uma propriedade `cor` e modifique o `ano` do carro. Imprima o objeto modificado.
3. Criar um Método em um Objeto: Crie um objeto chamado `calculadora` com um método `soma` que aceite dois números e retorne a soma deles.
4. Iterar Sobre as Propriedades de um Objeto: Dado um objeto, use um loop `for...in` para iterar sobre todas as suas propriedades e imprimir cada uma delas.
5. Entendendo Referências de Objetos: Crie um objeto `livro` com propriedades como `titulo` e `autor`. Em seguida, crie outra variável que referencie o objeto `livro` e modifique a propriedade `titulo`. Verifique se o objeto original foi alterado.
6. Usando Métodos Avançados de Objeto: Dado um objeto, use o método `Object.keys()` para listar todas as chaves do objeto. Repita o processo com `Object.values()` para listar todos os valores.