

10 - Classes

Se você tá pensando em aprender js, entender classes é crucial. Elas são como moldes para criar objetos mais organizados e reutilizáveis. Então, sem mais delongas, bora lá!

O que são Classes?

Classes, introduzidas no ES6, são um recurso do JavaScript que facilita a criação de objetos. Elas agem como um template ou um molde, a partir do qual você pode criar novos objetos que seguem um padrão.

Criando uma Classe

```
class Pessoa {  
  constructor(nome, idade) {  
    this.nome = nome;  
    this.idade = idade;  
  }  
  
  falar() {  
    console.log(`Oi, meu nome é ${this.nome} e tenho ${this.idade} anos.`);  
  }  
}
```

Aqui criamos uma classe **Pessoa** com um construtor e um método **falar**, métodos nada mais são do que funções que estão dentro de uma classe.

O construtor é chamado quando criamos uma nova instância da classe.

this.nome e **this.idade** são propriedades da classe.

Criando Objetos a partir de Classes

Para criar um objeto a partir de uma classe, usamos a palavra-chave **new**.

```
let joao = new Pessoa('João', 25);
```

Neste exemplo, **joao** é uma nova instância da classe **Pessoa**. Ele tem as propriedades **nome** e **idade** que definimos no construtor.

Pense que quando você chama a classe e abre os (), você está passando dados para o construtor.

Chamando Métodos da Classe

Vamos ver como chamar métodos definidos na classe.

```
joao.falar();
```

Aqui, **joao** chama o método **falar**. Isso vai imprimir a mensagem com o nome e a idade.

Herança de Classes

Uma das grandes vantagens das classes é a habilidade de herdar propriedades e métodos de outras classes.

Exemplo de Herança

```
class Estudante extends Pessoa {  
  constructor(nome, idade, curso) {  
    super(nome, idade);  
    this.curso = curso;  
  }  
  
  estudar() {  
    console.log(`Estudando ${this.curso}`);  
  }  
}
```

Criamos uma classe **Estudante** que herda de **Pessoa**.

Chamar a função **super(nome, idade)** chama o construtor da classe pai, ou seja, vai passar os dados para o construtor da classe **Pessoa**.

Agora, **Estudante** tem todas as propriedades e métodos de **Pessoa**, e de bônus os seus próprios (curso e estudar).

Criando Instância da Classe Herdada

Vamos criar um objeto da classe **Estudante**.

```
let ana = new Estudante('Ana', 22, 'Engenharia');
```

ana é uma instância de **Estudante**, com as propriedades de **Pessoa** e também **curso**.

Métodos Estáticos

As classes também podem ter métodos estáticos. Estes são métodos que você pode chamar diretamente na classe, sem criar uma instância dela.

Exemplo de Método Estático

```
class Utilitario {  
  static numeroAleatorio() {  
    return Math.floor(Math.random() * 100);  
  }  
}  
  
console.log(Utilitario.numeroAleatorio());
```

Aqui, **numeroAleatorio** é um método estático da classe **Utilitario**. Você pode chamá-lo diretamente, sem precisar instanciar a classe.

Propriedades Privadas

Com a introdução do ES2020, agora é possível definir propriedades privadas em classes, usando o **#**.

Exemplo de Propriedades Privadas

```
class ContaBancaria {  
  #saldo;  
  
  constructor(saldoInicial) {  
    this.#saldo = saldoInicial;  
  }  
  
  depositar(valor) {  
    this.#saldo += valor;  
  }  
  
  getSaldo() {  
    return this.#saldo;  
  }  
}  
  
let conta = new ContaBancaria(1000);  
conta.depositar(500);  
console.log(conta.getSaldo()); // 1500
```

A propriedade `#saldo` é privada. Isso significa que ela só pode ser acessada ou modificada dentro da classe `ContaBancaria`.

Exercícios para praticar

1. Criar uma Classe Animal: Crie uma classe **Animal** com um construtor que recebe o nome e o som que o animal faz. Adicione um método **falar** que imprime o som do animal.
2. Classe Veículo com Método de Velocidade: Desenvolva uma classe **Veiculo** com propriedades como **marca**, **modelo** e **velocidadeMaxima**. Adicione um método **descrever** que imprime uma descrição do veículo.
3. Classe Estudante com Notas: Amplie a classe **Estudante** para incluir um array de notas. Adicione um método **calcularMedia** que retorna a média das notas.
4. Método Estático para Conversão de Temperatura: Na classe **Utilitario**, adicione um método estático que converte temperatura de Celsius para Fahrenheit.
5. Jogo Simples com Classe: Crie uma classe **Jogo** com um método estático que gera um número aleatório e permite ao usuário adivinhar. O método retorna se o usuário acertou ou não.
6. Classe ContaBancaria com Método de Saque: Na classe **ContaBancaria**, adicione um método **sacar** que permite sacar um valor do saldo, garantindo que o saldo não fique negativo.