

Aula 5

Introdução à Programação Orientada a Objetos

Prof. Bruno Torres

Objetivos

- Entender os fundamentos de POO
- Aprender a criar classes e objetos em JavaScript
- Aplicar encapsulamento e herança
- Implementar um sistema simples com POO

O que é POO?

- Paradigma baseado em **objetos**
- Objetos têm **propriedades** (atributos) e **métodos** (ações)
- Inspiração em entidades do mundo real

Classe e Objeto

```
class Pessoa {  
    constructor(nome, idade) {  
        this.nome = nome;  
        this.idade = idade;  
    }  
  
    apresentar() {  
        return `Olá, meu nome é ${this.nome}`;  
    }  
}
```

Criando Objetos

```
const joao = new Pessoa("João", 30);  
console.log(joao.apresentar());
```

- `joao` é uma instância da classe `Pessoa`

Métodos da Classe

```
class Produto {  
    constructor(nome, preco) {  
        this.nome = nome;  
        this.preco = preco;  
    }  
  
    aplicarDesconto(percentagem) {  
        this.preco *= (1 - percentagem / 100);  
    }  
}
```

Encapsulamento

- Organização e proteção dos dados internos

```
class Conta {  
    #saldo = 0;  
  
    depositar(valor) {  
        this.#saldo += valor;  
    }  
  
    verSaldo() {  
        return this.#saldo;  
    }  
}
```

Getters e Setters

```
class Cliente {  
    constructor(nome) {  
        this._nome = nome;  
    }  
  
    get nome() {  
        return this._nome;  
    }  
  
    set nome(novoNome) {  
        this._nome = novoNome;  
    }  
}
```


Herança

```
class Animal {  
    falar() {  
        console.log("Som genérico");  
    }  
}  
  
class Cachorro extends Animal {  
    falar() {  
        console.log("Latido");  
    }  
}
```

Polimorfismo

```
const pet = new Cachorro();  
pet.falar(); // "Latido"
```

- Mesmo método, comportamento diferente

Classes como componentes reutilizáveis

- Classes facilitam organização do código
- Cada classe tem responsabilidade única

Projeto prático

Criar um sistema de cadastro de usuários com:

- Nome, idade, email
- Validação simples nos métodos
- Listagem de usuários no console

HTML base

```
<input id="nome" />  
<input id="idade" />  
<input id="email" />  
<button id="cadastrar">Cadastrar</button>
```

JS base

```
class Usuario {  
  constructor(nome, idade, email) {  
    this.nome = nome;  
    this.idade = idade;  
    this.email = email;  
  }  
  
  descrever() {  
    return `${this.nome}, ${this.idade} anos, ${this.email}`;  
  }  
}
```

Cadastro de usuários

```
const usuarios = [];  
  
document.getElementById("cadastrar").addEventListener("click", () => {  
  const user = new Usuario(nome.value, idade.value, email.value);  
  usuarios.push(user);  
  console.log(user.descrever());  
});
```

Conclusão

- POO organiza e simplifica grandes projetos
- Objetos modelam entidades reais
- JS moderno suporta POO com `class`

Obrigado!

Dúvidas?

Vamos praticar!