

// Definição do objeto Carro

```
class Carro {  
  constructor(modelo, cor) {  
    this.modelo = modelo;  
    this.cor = cor;  
    this.velocidadeAtual = 0;  
  }  
  
  ligar() {  
    console.log("O carro está ligado.");  
  }  
  
  acelerar(velocidade) {  
    this.velocidadeAtual += velocidade;  
    console.log(`Acelerando para ${this.velocidadeAtual} km/h.`);  
  }  
  
  parar() {  
    this.velocidadeAtual = 0;  
    console.log("O carro parou.");  
  }  
}
```

// Exemplo de uso do objeto Carro

```
const meuCarro = new Carro("Sedan", "Azul");  
meuCarro.ligar();  
meuCarro.acelerar(60);  
meuCarro.parar();
```

// Definição do objeto Livro

```
class Livro {  
  constructor(titulo, autor, numPaginas) {
```

```
this.titulo = titulo;
this.autor = autor;
this.numPaginas = numPaginas;
this.paginaAtual = 1;
}

abrir() {
  console.log("O livro está aberto na página 1.");
}

virarPagina() {
  this.paginaAtual++;
  console.log(`Virando para a página ${this.paginaAtual}.`);
}

fechar() {
  console.log("O livro está fechado.");
}
}
```

// Exemplo de uso do objeto Livro

```
const meuLivro = new Livro("Aventuras Fantásticas", "Autor Desconhecido", 200);
meuLivro.abrir();
meuLivro.virarPagina();
meuLivro.fechar();
```

// Definição do objeto Conta Bancária

```
class ContaBancaria {
  constructor(numeroConta, saldo, titular) {
    this.numeroConta = numeroConta;
    this.saldo = saldo;
    this.titular = titular;
  }
}
```

```
}
```

```
consultarSaldo() {  
  console.log(`Saldo atual: ${this.saldo}`);  
}
```

```
depositar(valor) {  
  this.saldo += valor;  
  console.log(`Depositado ${valor}. Novo saldo: ${this.saldo}`);  
}
```

```
retirar(valor) {  
  if (valor <= this.saldo) {  
    this.saldo -= valor;  
    console.log(`Retirado ${valor}. Novo saldo: ${this.saldo}`);  
  } else {  
    console.log("Saldo insuficiente.");  
  }  
}  
}
```

```
// Exemplo de uso do objeto Conta Bancária  
const minhaConta = new ContaBancaria("123456", 1000, "João");  
minhaConta.consultarSaldo();  
minhaConta.depositar(500);  
minhaConta.retirar(200);
```

```
// Definição do objeto Calendário  
class Calendario {  
  constructor(mesAtual, anoAtual, diaSemana) {  
    this.mesAtual = mesAtual;  
    this.anoAtual = anoAtual;
```

```
    this.diaSemana = diaSemana;  
}
```

```
avancarMes() {  
    this.mesAtual++;  
    console.log(`Avançou para o próximo mês: ${this.mesAtual}`);  
}
```

```
retrocederMes() {  
    this.mesAtual--;  
    console.log(`Retrocedeu para o mês anterior: ${this.mesAtual}`);  
}
```

```
consultarData() {  
    console.log(`Data atual: ${this.diaSemana}, ${this.mesAtual}/${this.anoAtual}`);  
}  
}
```

// Exemplo de uso do objeto Calendário

```
const meuCalendario = new Calendario(11, 2023, "Quinta-feira");  
meuCalendario.avancarMes();  
meuCalendario.retrocederMes();  
meuCalendario.consultarData();
```