

FULL STACK WEB DEVELOPER

FERNANDO LIRA

FLAG

JAVASCRIPT

APRESENTAÇÃO – FERNANDO LIRA

3



it.fernandolira@gmail.com



<https://www.linkedin.com/in/fernandolira74/>



+351 93 317 99 21



@fernandolira74



Definição de objeto – conceito de chave / valor

```
const produto = {  
  nome: 'Caneta',  
  preco: 9.99  
};  
  
console.log(typeof produto);  
  
console.log(produto['nome']);  
  
console.log(produto.nome);  
console.log(produto.preco);
```

Definição de métodos

```
const produto = {  
  nome: 'Caneta',  
  preco: 9.99,  
  desconto: 0.2,  
  precoComDesconto: function() {  
    return preco * (1 - desconto);  
  }  
};  
  
console.log(produto.nome);  
console.log(preco); //erro porque preco não existe  
console.log(produto.preco);  
  
console.log(produto.precoComDesconto()); //erro!!
```

Definição de métodos

```
const produto = {  
  nome: 'Caneta',  
  preco: 9.99,  
  desconto: 0.2,  
  precoComDesconto: function() {  
    return preco * (1 - desconto);  
  }  
};  
  
console.log(produto.nome);  
console.log(preco); //erro porque preco não existe  
console.log(produto.preco);  
  
console.log(produto.precoComDesconto()); //erro!!
```

O this

```
const produto = {  
  nome: 'Caneta',  
  preco: 9.99,  
  desconto: 0.2,  
  precoComDesconto: function() {  
    return this.preco * (1 - this.desconto);  
  }  
};  
  
console.log(produto.nome);  
  
console.log(produto.preco);  
  
console.log(produto.precoComDesconto());
```

Modificar propiedades (atributos)

```
const data = {  
  dia : 14,  
  mes : 12,  
  ano : 2021,  
};  
  
console.log(`${data.dia}/${data.mes}/${data.ano}`);  
  
data.dia = 20;  
data.mes = 12;  
  
console.log(`${data.dia}/${data.mes}/${data.ano}`);
```


Adicionar propriedades (atributos)

```
const utilizador = {  
  email: "it.fernandolira@gmail.com",  
};  
  
console.log(utilizador.email);  
console.log(utilizador.nome); //undefined  
  
utilizador.nome = "Fernando";  
  
console.log(utilizador.nome);  
console.log(utilizador);
```

Objetos complexos

```
const cliente = {
  codigo: 456,
  nome: "Ana Silva",
  vip: true,
  endereco: {
    rua: "Avenida principal",
    localidade: "Porto",
    numero: 45,
    pontosReferencia: [
      {referencia: "Hospital", distancia: 0.4 },
      {referencia: "Estação", distancia: 0.1 }
    ],
  },
  filhos: ["Ana", "Carlos"],
}

console.log(cliente.nome);
console.log(cliente.endereco);
console.log(cliente.endereco.rua);
console.log(cliente.endereco.pontosReferencia[0]);
console.log(cliente.endereco.pontosReferencia);
console.log(cliente.filhos[1]);
```

Problema: duplicação de código

```
const d1 = {  
  dia : 10,  
  mes : 12,  
  ano : 2021,  
  exibir : function () {  
    console.log(`${this.dia}/${this.mes}/${this.ano}`);  
  }  
}  
  
const d2 = {  
  dia : 2,  
  mes : 2,  
  ano : 2022,  
  exibir : function () {  
    console.log(`${this.dia}/${this.mes}/${this.ano}`);  
  }  
}  
  
d1.exibir();  
  
d2.exibir();
```

Solução

```
function criarData(dia, mes, ano) {  
  return {  
    dia: dia,  
    mes: mes,  
    ano: ano,  
    exibir : function () {  
      console.log(`${this.dia}/${this.mes}/${this.ano}`);  
    }  
  };  
}  
  
const d1 = criarData(10,12,2021);  
const d2 = criarData(2,2,2022);  
  
d1.exibir();  
  
d2.exibir();
```

Solução simplificada

```
function criarData(dia, mes, ano) {  
  return {  
    dia,  
    mes,  
    ano,  
    exibir() {  
      console.log(`${this.dia}/${this.mes}/${this.ano}`);  
    }  
  };  
}  
  
const d1 = criarData(10,12,2021);  
const d2 = criarData(2,2,2022);  
  
d1.exibir();  
  
d2.exibir();
```

Class

```
class Data {  
  
    constructor(dia = 1, mes = 1, ano = 2021) {  
        this.dia = dia;  
        this.mes = mes;  
        this.ano = ano;  
    }  
  
    exibir() {  
        console.log(`${this.dia}/${this.mes}/${this.ano}`);  
    }  
}  
  
const d1 = new Data();  
const d2 = new Data(15,2,2023);  
  
d1.exibir();  
d2.exibir();
```

EXERCÍCIOS

Desafios

Escreva uma classe cujos objetos representam alunos matriculados em uma disciplina. Cada objeto dessa classe deve guardar os seguintes dados do aluno: nome, 2 notas de prova e 1 nota de trabalho. Escreva o seguinte métodos para esta classe e implemente o interface.

- **media** calcula a média final do aluno (cada prova tem peso 2,5 e o trabalho tem peso 2)

Se o aluno matriculado tiver o estatuto de atleta de alta competição só é tido em consideração a prova mais alta com um peso de 4 e o trabalho com um peso de 2

Desafios

Escreva uma classe cujos objetos representam contas bancárias.

Cada objeto dessa classe devem guardar os seguintes dados: NIB, Nome, Saldo (opcional = 20) e lista de movimentos.

Desenvolve a classe com o método Movimento que recebe os seguintes argumentos (Tipo de movimento (D-Débito ou C-Crédito), Descrição do movimento, Valor).

No final do método Movimento ser invocado, deve ser apresentado o NIB, Nome e Saldo atual do objeto.

Criar a classe listarMovimentos para listar todos os movimentos efetuados.