

# Classes e JSON

Classes oferecem uma maneira organizada de modelar objetos e seu comportamento. Ao compreender classes, você poderá criar código mais limpo e reutilizável em seus projetos.

JSON é um formato de dados leve e amplamente utilizado para facilitar a troca de informações. Caso queira enviar um objeto para o back-end, ou receber um objeto no front.

---

## Questão 1

Responda as perguntas abaixo:

### 1. O que são objetos em JavaScript?

Objetos são estruturas de dados que representam entidades do mundo real. Podem ser instâncias de classes ou coleções de pares chave-valor, onde as chaves são strings e os valores podem ser de qualquer tipo de dado.

Os objetos em JavaScript são flexíveis e dinâmicos, permitindo a adição e remoção de propriedades dinamicamente.

### 2. O que são atributos em JavaScript?

Atributos são as características ou propriedades de um objeto, que representam dados associados a esse objeto.

Podem ser acessados e manipulados indicando **objeto.atributo** ou como um item do array de itens do objeto **objeto['atributo']**.

### 3. O que são métodos de classe em JavaScript?

Métodos de classe são funções associadas a uma classe. São definidos no corpo da classe e operam nos dados da instância da classe. São funções que podem ser chamadas em instâncias da classe.

### 4. O que é uma classe em JavaScript?

Classe é uma definição padrão de objetos que o JavaScript incorporou a partir de 2015. Antes não tinha uma sintaxe de classe como em outras linguagens de programação orientada a objetos.

A sintaxe de classe para criar objetos e construtores o JavaScript ficou parecido com outras linguagens orientadas a objetos.

### 5. O que é o construtor de uma classe em JavaScript?

O construtor é um método especial (função) usado para criar um objeto a partir da definição da classe. No construtor as propriedades do objeto são inicializadas e por ainda executar outro comando necessário para criar um objeto da classe. É definido usando o nome especial `constructor` dentro da classe.

6. Explique o conceito de herança de classes em JavaScript. De um exemplo de uma herança em JavaScript.

Herança é uma característica que permite que uma classe (subclasse ou derivada) possa ser criada usando as propriedades e métodos de outra classe (superclasse ou pai).

Assim pode-se reutilizar o código da superclasse e a subclasse pode estender ou modificar o comportamento da classe superclasse.

---

## Questão 2

- Crie a classe Produto com os atributos nome e preço
- Crie a classe Cliente com os atributos nome e cpf
- Crie a classe Venda com atributo cliente e lista de produtos, e com o método total que calcula o valor total de todos os produtos da venda
- Crie uma lista de clientes a partir da seguinte string em formato json

```
const clientsStringJson = '[{"name":"Client 1", "cpf":"12345678901"},'+ '{"name":"Client 2", "cpf":"10987654321"},'+ '{"name":"Client 3", "cpf":"12345098765"},'+ '{"name":"Client 4", "cpf":"12345098765"},'+ '{"name":"Client 5", "cpf":"12345098765"},'+ '{"name":"Client 6", "cpf":"12345098765"},'+ '{"name":"Client 7", "cpf":"12345098765"},'+ '{"name":"Client 8", "cpf":"12345098765"},'+ '{"name":"Client 9", "cpf":"12345098765"},'+ '{"name":"Client 10", "cpf":"12345098765"}'];
```

- Crie uma lista de produtos a partir da seguinte string em formato json

```
const productsStringJson = '[{"name":"Product 1", "price":1.00},'+ '{"name":"Product 2", "price":2.00},'+ '{"name":"Product 3", "price":3.00},'+ '{"name":"Product 4", "price":4.00},'+ '{"name":"Product 5", "price":5.00},'+ '{"name":"Product 6", "price":6.00},'+ '{"name":"Product 7", "price":7.00},'+ '{"name":"Product 8", "price":8.00},'+ '{"name":"Product 9", "price":9.00},'+ '{"name":"Product 10", "price":10.00}'];
```

- Crie o objeto vendao cliente 1 e todos os produtos, depois imprimna o valor total da venda

---

## Questão Extra

- Adapte o codigo da questão 2, criando um menu que permite ao usuario:
  - inserir clientes
  - inserir produtos
  - realizar venda

OBS: ao realizar venda, o usuario deve selecionar quantos produtos desejar

---