

Exercício Prático: Criando Classes no JavaScript

Objetivo: O objetivo deste exercício é treinar a criação de classes no JavaScript e trabalhar com atributos e métodos. Você irá criar duas classes diferentes: `Member` e `Item`, e implementará funcionalidades específicas para cada uma delas.

1. Classe `Member`

A classe `Member` deve ter os seguintes atributos:

- `fullname`: O nome completo do membro (atributo atribuível na instanciação).
- `email`: O e-mail do membro (atributo atribuível na instanciação).
- `password`: A senha do membro (atributo atribuível na instanciação).

A classe também deve ter o seguinte método:

- `authenticate(email, password)`: Este método deve receber dois parâmetros, o `email` e a `password`, e verificar se eles coincidem com os atributos `email` e `password` do membro. Se a comparação for bem-sucedida, o login deve ser considerado realizado com sucesso, e uma mensagem deve ser exibida no console. Caso contrário, deve ser exibida uma mensagem de falha.

2. Classe `Item`

A classe `Item` deve ter os seguintes atributos:

- `name`: O nome do item (atributo atribuível na instanciação).
- `description`: A descrição do item (atributo atribuível na instanciação).
- `price`: O preço do item (atributo atribuível na instanciação).
- `quantityAvailable`: A quantidade do item disponível, que deve ser inicializada sempre como 0.

A classe também deve ter os seguintes métodos:

- `addToInventory(quantity)`: Este método deve receber a quantidade a ser adicionada ao inventário e somá-la ao valor atual de `quantityAvailable`.
- `applyOffer(discount)`: Este método deve calcular o preço do item após a aplicação de um desconto. O parâmetro `discount` representa a porcentagem de desconto e o método deve retornar o valor final do item com o desconto.

Instruções para o Exercício:

1. Classe `Member`:
 - Crie a classe `Member` conforme descrito.
 - Implemente o método `authenticate`, verificando se o e-mail e a senha fornecidos no método coincidem com os dados do membro.
2. Classe `Item`:
 - Crie a classe `Item` conforme descrito.
 - Implemente os métodos `addToInventory` e `applyOffer`.
3. Testando as Classes:
 - Crie pelo menos um membro e um item, instanciando-os e utilizando os métodos `authenticate`, `addToInventory` e `applyOffer` para verificar se tudo funciona corretamente.

Exemplo Esperado de Execução:

```
const alice = new Member("Alice Johnson", "alice@email.com",  
"password123");  
  
alice.authenticate("alice@email.com", "password123"); // Login  
bem-sucedido.  
  
alice.authenticate("alice@email.com", "wrongpassword"); // Falha no  
login.
```

Testando a Classe Item

```
const chair = new Item("Cadeira de Escritório", "Cadeira confortável  
com ajuste de altura", 150);  
  
chair.addToInventory(50);  
  
console.log(chair.quantityAvailable); // 50  
  
console.log(chair.applyOffer(20)); // 120  
  
console.log(chair.applyOffer(10)); // 135
```

Agora, crie o seu código seguindo os passos acima e depois verifique a resolução do exercício na próxima aula e no arquivo de gabarito!