

Módulo 17

Exercícios - modularização

Lembre-se de que os exercícios podem ser resolvidos com diferentes abordagens, e nosso gabarito é apenas uma forma de implementação.

Nesta etapa do curso, você encontrará uma lista de desafios que foram projetados para estimular sua curiosidade e promover uma abordagem prática no aprendizado de programação. Embora alguns dos conceitos abordados nesses exercícios ainda não tenham sido explicados detalhadamente, essa abordagem faz parte da nossa metodologia de ensino.

Objetivo dos Desafios:

- **Simular a Vida de Programador:** Ao se deparar com problemas que envolvem conceitos ainda não estudados, você terá a oportunidade de desenvolver habilidades de pesquisa e resolução de problemas, semelhantes às que um programador enfrenta no seu dia a dia.
- **Incentivar a Pesquisa:** A ideia é que você procure soluções, entenda novos conceitos e aprenda a aplicar técnicas que ainda serão discutidas nas aulas futuras. Isso ajudará a solidificar seu entendimento e prepará-lo para os tópicos que serão abordados em breve.
- **Preparação para o Futuro:** Ao resolver os desafios e pesquisar os conceitos, você estará se preparando para as explicações detalhadas que virão posteriormente no curso. Cada exercício é uma oportunidade para explorar e aprender de forma independente.

Como Abordar os Exercícios:

1. **Pesquise e Experimente:** Antes de procurar a solução completa, tente entender o problema e busque informações sobre os conceitos que estão relacionados. Use recursos como documentação, fóruns e tutoriais.
2. **Resolva com Criatividade:** A abordagem não precisa ser perfeita. O objetivo é experimentar e ver o que funciona, e o que não funciona.
3. **Refleta e Revise:** Após encontrar e entender a solução, revise o exercício e compare com o que foi discutido nas aulas subsequentes. Isso ajudará a reforçar seu aprendizado.

O que Esperar Após os Desafios:

- **Explicação Detalhada:** Após concluir os desafios, você receberá uma explicação detalhada dos conceitos envolvidos e das melhores práticas para resolver os problemas. Isso garantirá que você compreenda completamente o que foi proposto e como aplicar o conhecimento em situações futuras.

Prepare-se para um desafio enriquecedor e lembre-se: a pesquisa e a prática são ferramentas essenciais para se tornar um programador habilidoso. Boa sorte e divirta-se explorando!

Exercício 1: Crie dois arquivos JavaScript (`mathOperations.js` e `index.js`). No `mathOperations.js`, exporte duas funções utilizando CommonJS: `add(a, b)` e `subtract(a, b)`, que realizam as operações básicas de adição e subtração. No `index.js`, importe essas funções e utilize-as para calcular a soma e a subtração de dois números fornecidos pelo usuário.

Desafio Extra: Modifique o `mathOperations.js` para incluir uma função `multiply(a, b)` e a exporte de forma condicional (apenas se o ambiente for de desenvolvimento).

Exercício 2: Converta o módulo `mathOperations.js` do exercício anterior para utilizar ES Modules em vez de CommonJS. Implemente a exportação das funções (`add`, `subtract`, `multiply`) em um único objeto e faça as correções necessárias para que seu programa funcione corretamente. No `index.js`, corrija a importação para ES Modules. A função `multiply` pode ser executada independentemente do ambiente de execução.

Desafio Extra: Observe os erros que podem ocorrer com a conversão e aplique os conceitos do módulo para corrigi-los.

Exercício 3: Refatore o código do exercício anterior para usar a exportação nomeada (named exports) em vez de exportar um objeto. E no `index.js`, importe apenas a função `add` e a utilize para somar dois números.

Para o exercício 4 - caso você não queira criar os arquivos `html` e `css`, você pode fazer o download da pasta `Projeto_modular` e utilizar os arquivos.

Exercício 4: Desenvolva uma pequena aplicação de gerenciamento de usuários com HTML, CSS e JavaScript, seguindo os princípios de módulos ES e interatividade na página. Crie um módulo que exporte um objeto com várias funções relacionadas ao gerenciamento de usuários (e.g., `addUser`, `deleteUser`, `updateUser`). Utilize exportação padrão para exportar o objeto completo. No arquivo principal (`index.js`), importe o objeto completo e utilize pelo menos duas funções.

- Crie o arquivo `userManager.js` para definir as funções de gerenciamento de usuários.
- Crie o arquivo `index.js` para importar o módulo e adicionar interatividade.