

Lista de Exercícios 01 - NODE

- Exercício pode ser feito individual ou em dupla
 - Códigos desenvolvidos por Inteligência Artificial (IA) ou plágios acarretam nota zero.
1. Utilizando o CJS, implemente um módulo de manipulação de datas e em seguida uma aplicação que utilize esse módulo para calcular a idade com base na data de nascimento:
 - **caculateAge(dt)** – Esta função deve aceitar uma data de nascimento como entrada e calcular a idade com base na data atual.
 2. Utilizando o ESM, desenvolva um módulo para calcular média das notas, menor e maior nota. Em seguida, desenvolva uma aplicação que utilize esse módulo passando como parâmetro um conjunto de notas de uma turma.
 - **media(vetor)** – retornar a média dos valores contidos em um vetor passado como parâmetro.
 - **menor(vetor)** – retornar o menor valor contido no vetor passado como parâmetro.
 - **maior(vetor)** – retornar o maior valor contido em um vetor passado como parâmetro.
 3. Criar uma aplicação para manipulação de Arrays com a Biblioteca "lodash".

Parte 1: Configuração Inicial

1. Certifique-se de que o Node.js esteja instalado em sua máquina.
2. Crie uma nova pasta para esta atividade e navegue até ela no terminal.
3. Inicialize um projeto Node.js no terminal

Parte 2: Instalação da Biblioteca "lodash"

4. Instale a biblioteca "lodash" usando o seguinte comando:
npm install lodash

Parte 3: Criando a Aplicação

5. Crie um arquivo JavaScript chamado "app.js".
6. Dentro de "app.js", importe a biblioteca "lodash"

Parte 4: Usando a Biblioteca "lodash"

1. Crie um array de números
2. Use a biblioteca "lodash" para realizar as seguintes operações de manipulação de arrays e imprima os resultados no console:
 - a. Ordene o array em ordem crescente.
 - b. Remova os elementos duplicados do array.
 - c. Filtrar e crie um novo array com apenas os números pares.
 - d. Mapeie o array para criar um novo array com o dobro de cada número.
 - e. Calcule a soma de todos os elementos do array.

Parte 5: Executando a Aplicação

1. No terminal, execute o arquivo "app.js" usando o Node.js
 2. Verifique a saída no console para ver os resultados das operações de manipulação de arrays usando a biblioteca "lodash".
-
4. Criar uma aplicação para manipulação de arquivos com tratamento de erro, utilizando a biblioteca "file system" (biblioteca interna do Node) e cores com o "chalk".

Parte 1: Configurar o Projeto

1. Inicie um projeto Node.js e crie um arquivo package.json.
2. Instale a biblioteca Chalk.

Parte 2: Importar as Bibliotecas

- No arquivo principal do projeto, importe as bibliotecas fs (File System) e chalk.

Parte 3: Ler o Arquivo Markdown

- Utilize a biblioteca `fs` para ler o conteúdo de um arquivo Markdown, garantindo que o conteúdo seja lido em formato de texto.

Parte 4: Encontrar Links no Texto (opcional)

- Defina uma expressão regular que identifique links no formato Markdown, ou seja, no padrão `[texto](url)`.

Parte 5: Colorir Links com Chalk

- Use a biblioteca `Chalk` para substituir os links detectados e aplicar cores distintas no texto do link e no próprio URL, destacando-os sem modificar o restante do texto.

Parte 6: Exibir o Texto Colorido no Console

- Por fim, imprima o conteúdo no console para visualizar os links destacados em cores, o que ajudará a confirmar a formatação correta.

5. Uma empresa de desenvolvimento de software está criando um sistema de gerenciamento de arquivos para uma equipe de marketing. O sistema deverá facilitar o armazenamento e a manipulação de arquivos JSON que contêm informações de clientes e campanhas. Sua tarefa é criar um módulo em Node.js usando o formato CommonJS, responsável por funções como leitura, escrita, e atualização desses arquivos JSON.

Objetivo da Atividade:

- Desenvolver um módulo chamado `fileManager.js` que ofereça métodos para:
 1. Ler arquivos JSON e retornar os dados em formato de objeto.
 2. Escrever novos dados em um arquivo JSON, caso ele já exista, sobrescrevendo o conteúdo antigo.

Requisitos

1. **Ler Arquivo JSON:** Desenvolva uma função `readJsonFile(filePath)` que leia e retorne o conteúdo de um arquivo JSON como um objeto JavaScript.
2. **Escrever Arquivo JSON:** Desenvolva uma função `writeJsonFile(filePath, data)` que receba um objeto `data` e sobrescreva o conteúdo do arquivo especificado.