

Exercício 1

Crie uma instância da classe GerenciadorSistema e:

Use o método `diretorio_atual()` para mostrar o diretório de trabalho atual.

Imprima o caminho retornado.

Exercício 2

Use `listar_conteudo()` sem argumentos para listar o conteúdo do diretório base.

Depois, liste o conteúdo de outro diretório qualquer, passando o caminho como argumento.

Exercício 3

Use o método `mudar_diretorio()` para alterar o diretório de trabalho para outro caminho válido (por exemplo, `".."` para subir um nível).

Após mudar, use `diretorio_atual()` novamente para verificar a alteração.

Retorne ao diretório original usando `mudar_diretorio()`.

Exercício 4

Crie um novo diretório chamado teste usando `criar_diretorio`.

Depois, crie uma estrutura mais profunda com `criar_diretorio sub_diretorio`.

Exercício 5

Use o método `existe()` para verificar se o diretório teste existe.

Verifique se um arquivo inexistente.

Exercício 6

Use `remover_diretorio` para remover um subdiretório.

Em seguida, remova o diretório teste com `remover_diretorio`.

Tente remover um diretório que não esteja vazio e observe o comportamento.

Exercício 7

Crie um arquivo manualmente usando `open()` com o caminho gerado por `juntar_caminho()`.

Escreva algo dentro do arquivo.

Depois, use `remover_arquivo()` para excluí-lo.

Verifique com `existe()` se ele foi realmente removido.

Exercício 8

Crie um arquivo chamado `exemplo.txt`.

Use `renomear` para renomeá-lo.

Verifique se o novo nome existe usando `existe()`.

Exercício 9

Use `juntar_caminho('meus_arquivos', 'dados', 'arquivo.txt')`.

Imprima o resultado para ver como o caminho fica formatado no seu sistema operacional.

Exercício 10

Use `pegar_variavel_ambiente` para obter o valor da variável `PATH`.

Tente acessar outras variáveis, como `PYTHONPATH`, `USER`, `TEMP`.

Imprima os valores retornados.