

Excepciones

1. Crea una función que intente dividir dos números proporcionados por el usuario. Usa try-except para capturar cualquier error de división (por ejemplo, división por cero).
2. Crea una función que tome una cadena e intente convertirla en un número entero. Usa try-except para capturar cualquier error en la conversión.
3. Crea una función que abra un archivo, lea su contenido y maneje posibles errores (por ejemplo, archivo no encontrado). Usa try-except para gestionar las operaciones de archivos de forma segura.
4. Crea una función que realice múltiples operaciones (suma, resta, división, multiplicación) con dos números. Usa try-except-else-finally para manejar errores y asegurar que se imprima un mensaje final, independientemente de los errores.
5. Crea una función que le pida al usuario su edad y lance un ValueError si la entrada no es un número entero positivo. Usa el manejo de excepciones para gestionar la entrada y lanzar excepciones personalizadas cuando sea necesario.
6. Crea una función que intente acceder a un elemento de una lista por índice. Usa try-except para manejar el caso donde el índice esté fuera de rango.
7. Crea una función que use try-except para manejar múltiples excepciones: ZeroDivisionError, ValueError y TypeError.
8. Crea una función que simule una transacción. Lanza una excepción personalizada llamada InsufficientFundsError si el saldo es menor que la cantidad a retirar.
9. Crea una función que intente convertir una lista de cadenas en enteros. Maneja cualquier error que surja cuando una cadena no pueda convertirse.
10. Crea una función que calcule la raíz cuadrada de un número. Lanza un ValueError si el número es negativo.