
Exercício 01

Nesta questão você deve identificar as partes problemáticas do código e reescrevê-lo utilizando tratamento de exceções.

Ou seja, devem ser identificadas todas as exceções que podem ser geradas e, para cada uma, deve ser dado o tratamento adequado que, nesse exercício, significa alertar o usuário quanto ao problema.

```
x = int(input('Primeiro valor:'))
y = int(input('Segundo valor:'))
z = x / y
print('O resultado da divisão é:', z)
```

Exercício 02

O código abaixo lança uma exceção e interrompe a execução do programa.

Utilizando tratamento de exceções, corrija o programa com o objetivo de alertar o usuário sobre o erro ocorrido, e impedir que o programa seja interrompido bruscamente.

```
def funcao_1():
    print('Início da função')
    lista = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
    for i in range(15):
        print(lista[i])
    print('Fim da função')

print('Início do programa')
funcao_1()
print('Fim do programa')
```

Exercício 03

Preencha uma lista com 5 nomes de pessoas, informados pelo usuário.

- Criar uma função que recebe como parâmetro de entrada a lista e uma posição (índice) dessa lista e retorna o nome que está nessa posição.
 - Essa função deve gerar e tratar uma exceção do tipo **IndexError** caso o índice não exista na lista.

Exercício 04

Crie um dicionário para armazenar uma listagem de alunos.

- Utilize como chave o RA do aluno e como valor o nome do aluno.
- Os dados devem ser informados pelo usuário
- O RA de cada aluno deve ser composto por um número inteiro de exatamente 7 dígitos.
 - Caso o RA informado não esteja no formato correto, deve ser gerada uma exceção do tipo **ValueError** (utilize a instrução `raise`).
 - Caso o RA informado já exista no dicionário, deve ser gerada uma exceção do tipo **TypeError** (utilize a instrução `raise`).

Observação: Você pode utilizar o código abaixo como exemplo e alterá-lo para gerar e tratar as exceções solicitadas.

```
alunos = {}
for i in range(5):
    ra = int(input('RA: '))
    nome = input('Nome: ')
    alunos[ra] = nome
print(alunos)
```

Exercício 05

Importe o módulo abaixo para um programa de teste e escreva testes unitários para as funções do módulo:

```
def converte_para_celsius(fahrenheit):
    celsius = (5.0/9.0) * (fahrenheit - 32)
    return celsius

def converte_para_fahrenheit(celsius):
    fahrenheit = 1.8 * celsius + 32
    return fahrenheit
```

Utilize os valores abaixo como parâmetros de entrada e saída das funções.

converte_para_fahrenheit(celsius)	
Entrada	Saída
0	32.0
27	80.6

converte_para_celsius(fahrenheit)	
Entrada	Saída
32	0
41	5.0