

## Exercícios Exceções:

Documentação para consultar exceptions do Python:

<https://docs.python.org/3/library/exceptions.html>

1. Faça um programa que calcule a raiz quadrada de um número  $n$  e trate os casos em que  $n < 0$ .  
**OBS:** Utilize o [módulo math](#) para calcular a raiz quadrada.
2. Faça um programa que calcule a divisão de dois números  $m$  e  $n$  e trate os casos em que  $n = 0$ .
3. Observe o programa abaixo:

```
number = int(input("Digite um número: "))  
print("O número digitado foi:", number)
```

Reescreva esse código de uma forma com que ele seja capaz de tratar a inserção de um caractere por parte do usuário.

4. Observe o seguinte programa:

```
number = int(input("Digite um número: "))  
print("O número digitado foi:", n)
```

Tendo em mente que ao executá-lo a exceção **NameError** é gerada, reescreva o código de forma com que tal exceção seja tratada.

5. Observe o seguinte programa:

```
import mathematics  
print(math.sqrt(25))
```

Qual exceção será gerada durante a execução?

6. Observe o seguinte programa:

```
try:
    number_1 = float(input("Insira um número: "))
    number_2 = float(input("Insira outro número: "))
    result = number_1 / number_2

    print("Resultado: {:.2f}".format(resultado))
except ValueError:
    print("Isso não é um número.")
except ZeroDivisionError:
    print("Divisão por 0 indeterminada.")
except:
    print("Algo deu errado.")
```

Escreva o que será impresso na tela caso o mesmo seja executado com as seguintes entradas (5, 3):

```
Insira um número: 5
Insira outro número: 3
```

7. Uma colega tentou executar o seguinte programa:

```
file = open("file.txt", "r")
file_lines = file.readline()
file.close()
```

E obteve a seguinte mensagem de erro:

```
Traceback (most recent call last):
  File "/home/michelly/Documents/Luiza Code/Códigos/exceptions_exercises.py", line 22, in <module>
    file = open("file.txt", "r")
FileNotFoundError: [Errno 2] No such file or directory: 'file.txt'
```

Reescreva o código para que esse erro seja exibido de forma mais clara e amigável.

8. Observe o seguinte programa:

```
try:
    file = open("file.txt", "r")
    file.write("Exemplo de texto.")
except IOError:
    print("Não foi possível escrever no arquivo.")
```

Um dos problemas do código acima é que o mesmo além de possuir um erro lógico (execução de um comando de escrita em um arquivo em modo de leitura), abre um arquivo e tem a sua execução encerrada sem realizar o **close**.

Incremente o código apresentado liberando da memória a referência do arquivo tanto nos casos de erro, quanto em execuções bem sucedidas (caso o mesmo fosse aberto em modo de escrita).

Leitura para ajudar na resolução do exercício 8:

<https://docs.python.org/3/tutorial/errors.html#defining-clean-up-actions>