

Aplicando

October 13, 2023

1 Exception handling - Aula 8

1.1 Aplicando

Desenvolva um programa em Python que peça para o usuário entrar com dois números inteiros.

Calcule e mostre o resultado da divisão do primeiro pelo segundo, caso não haja erro. O programa não deve continuar se houver algum erro.

Atenção, use o controle de exceção (`try-except` nas leituras – `ValueError` – e controle da divisão por zero – `ZeroDivisionError`).

1.2 Resposta

```
[10]: from numbers import Number

def read_numbers() -> (Number, Number):
    num1 = int(input("Digite o primeiro número: "))
    num2 = int(input("Digite o segundo número: "))
    return num1, num2

def divide(a: Number, b: Number) -> Number:
    result = a / b
    return result

def read_and_divide() -> Number:
    try:
        a, b = read_numbers()
        result = divide(a, b)
        return result

    except ValueError:
        print("Erro: Entre números inteiros válidos.")
        return None

    except ZeroDivisionError:
        print("Erro: O segundo número não pode ser zero.")
```

```
return None
```

1.3 Testes

```
[11]: # Entra com 10 e 10  
read_and_divide()
```

```
[11]: 1.0
```

```
[12]: # Entra com 1.1 e 0.5  
read_and_divide()
```

Erro: Entre números inteiros válidos.

```
[13]: # Entra com 10 e 0  
read_and_divide()
```

Erro: O segundo número não pode ser zero.