



escola  
britânica de  
artes criativas  
& tecnologia

## Módulo | Python: Tratamento de Erros

Caderno de **Aula**

Professor [André Perez](#)

### Tópicos

1. Tipos de erros;
2. Erros de sintaxe;
3. Erros em tempo de execução.

### Aulas

#### 1. Tipos de erros

##### 1.1. Definição

- **Erros de sintaxe:** erros que ocorrem durante a escrita do código.

```
In [ ]: pessoa = {'nome': 'Andre Perez', 'idade': 19}

if pessoa['idade'] > 18:
    print(True)
```

- **Erros em tempo de execução:** erros que ocorrem durante a execução do código.

Erros de uso incorreto de tipos de dados.

```
In [ ]: print(1 + 'Andre')
```

Erros de lógica.

```
In [ ]: i = 0
while True:
    ... # bloco de código
```

```
i = 1
i + 1
if i > 10:
    break

print(i)
```

## 2. Erros de sintaxe

### 2.1. Definição

São erros que ocorrem durante a escrita do código. O trecho do código **não é** executado.

```
In [ ]: carrinho_compras = [
        {'id': 3184, 'preco': 37.65, 'qtd': 10},
        {'id': 1203, 'preco': 81.20, 'qtd': 2},
        {'id': 8921, 'preco': 15.90, 'qtd': 2}
    ]
```

**Exemplo:** Esquecer o 'dois pontos' : no final de estruturas de condição, repetição, etc.

```
In [ ]: for produto in carrinho_compras:
        ...
```

**Exemplo:** Condição lógica no else da estrutura de decisão if-elif-else .

```
In [ ]: for produto in carrinho_compras:
        if produto['id'] == 3184:
            ...
        elif produto['id'] == 1203:
            ...
```

### 2.2. Manipulação

## 3. Erros em tempo de execução

### 3.1. Motivação

Você trabalha como analista de dados em uma empresa de telecomunicações. Você precisa fazer uma análise para o time de vendas do quanto a empresa vai receber este mês.

Diariamente você recebe do time de engenharia os dados.

```
In [ ]: %%writefile telecom.csv
customerID,PaymentMethod,MonthlyCharges,TotalCharges,Churn
7010-BRBUU,Credit card (automatic),24.1,1734.65,No
9688-YGXVR,Credit card (automatic),88.15,3973.2,No
9286-DOJGF,Bank transfer (automatic),74.95,2869.85,Yes
6994-KERXL,Electronic check,55.9,238.5,No
2181-UAESM,Electronic check,53.45,119.5,No
4312-GVYNH,Bank transfer (automatic),49.85,3370.2,No
2495-KZNFB,Electronic check,90.65,2989.6,No
4367-NHWM, Mailed check,24.9,24.9,No
8898-KASCD, Mailed check,35.55,1309.15,No
```

```
In [ ]:
```

```

from functools import reduce

def processar_faturas(nome_arquivo: str) -> float:

    faturas = []

    with open(file=nome_arquivo, mode='r', encoding='utf8') as arquivo:
        linha = arquivo.readline()
        linha = arquivo.readline()
        while linha:
            fatura = float(linha.strip().split(sep=',')[-3])
            faturas.append(fatura)
            linha = arquivo.readline()

    total_a_pagar = reduce(lambda x, y: x + y, faturas)
    total_a_pagar = round(total_a_pagar, 2)

    return total_a_pagar

```

```

In [ ]: total_a_pagar = processar_faturas(nome_arquivo='./telecom.csv')
        print(total_a_pagar)

```

Em um certo dia, você recebe uma base de dados com a coluna de faturas trocada pela de meios de pagamento.

```

In [ ]: %%writefile telecom.csv
        customerID,MonthlyCharges,PaymentMethod>TotalCharges,Churn
        7010-BRBUU,24.1,Credit card (automatic),1734.65,No
        9688-YGXVR,88.15,Credit card (automatic),3973.2,No
        9286-DOJGF,74.95,Bank transfer (automatic),2869.85,Yes
        6994-KERXL,55.9,Electronic check,238.5,No
        2181-UAESM,53.45,Electronic check,119.5,No
        4312-GVYNH,49.85,Bank transfer (automatic),3370.2,No
        2495-KZNFB,90.65,Electronic check,2989.6,No
        4367-NHWM,24.9,Mailed check,24.9,No
        8898-KASCD,35.55,Mailed check,1309.15,No

```

## 3.2. Definição

São erros que ocorrem durante a execução do código. O trecho do código é executado até o erro 'estourar'.

Erros por uso incorreto de tipos de dados. 'Estoura' exceção. Podem ser manipulados ou passados para frente ( `raise` ).

- **Exemplo:** Erro de operações numéricas impossíveis

```

In [ ]: preco = 132.85
        pessoas = 0

```

```

In [ ]: valor_por_pessoa = preco / pessoas

```

- **Exemplo:** Erro por combinações de tipos diferentes

```

In [ ]: nome = 'Andre Perez'
        idade = 20

```

```
In [ ]: apresentacao = 'Fala pessoal, meu nome é ' + nome + \
                ' e eu tenho ' + idade + ' anos'
```

- **Exemplo:** Erro de indexação de estrutura de dados

```
In [ ]: anos = [2019, 2020, 2021]
```

```
In [ ]: ano_atual = anos[3]
print(ano_atual)
```

```
In [ ]: cursos = {
    'python': {
        'nome': 'Python para Análise de Dados', 'duracao': 2.5
    },
    'sql': {
        'nome': 'SQL para Análise de Dados', 'duracao': 2
    }
}
```

```
In [ ]: curso_atual = cursos['python']
print(curso_atual)
```

```
In [ ]: curso_atual = cursos['sql']
print(curso_atual)
```

```
In [ ]: curso_atual = cursos['analista']
print(curso_atual)
```

**Erros de lógica.** Não 'estoura' exceção. A melhor forma de análise é usar a função `print` para verificar resultados intermediários.

- **Exemplo:** Loops infinitos.

```
In [ ]: controle = 0
while True:
    ...
    controle += 1
    if controle > 10:
        break
```

```
In [ ]: s = []
while True:
    s = s + \
        (
            ['CURSO DE PYTHON PARA ANALISE DE DADOS DA EBAC'] * \
            (1000 * 1000 * 1000)
        )
```

- **Exemplo:** Limites de coleções.

```
In [ ]: carrinho_compras = [
        {'id': 3184, 'preco': 37.65, 'qtd': 10},
        {'id': 1203, 'preco': 81.20, 'qtd': 2},
        {'id': 8921, 'preco': 15.90, 'qtd': 2}
    ]
```

```
In [ ]: valor_total = 0
for indice in range(1, len(carrinho_compras)):
    valor_total += carrinho_compras[indice]['preco'] * \
        carrinho_compras[indice]['qtd']

valor_total = round(valor_total, 2)
print(valor_total)
```

```
In [ ]: valor_total = 0
for produto in carrinho_compras:
    valor_total += produto['preco'] * produto['qtd']

valor_total = round(valor_total, 2)
print(valor_total)
```

```
In [ ]: valor_total = 0
for indice in range(0, len(carrinho_compras)):
    print(carrinho_compras[indice])
    valor_total += carrinho_compras[indice]['preco'] * \
        carrinho_compras[indice]['qtd']

valor_total = round(valor_total, 2)
print(valor_total)
```

### 3.3. Manipulação

- Manipular o erro com a estrutura `try-catch-finally-else`.

```
In [ ]: # anos = [2019, 2020, 2021]
anos = {2019, 2020, 2021}

try:
    ano_atual = anos[3]
    print(ano_atual)
except IndexError:
    print('Lista de anos é menor que o valor escolhido. ' + \
        'Espera-se um valor entre 0 e ' + \
        str(len(anos) - 1)
    )
except Exception as exc:
    print(exc)
finally:
    ...
```

- Passar o erro para frente com a estrutura `raise`.

```
In [ ]: anos = [2019, 2020, 2021]
# anos = {2019, 2020, 2021}
```

```

try:
    ano_atual = anos[3]
    print(ano_atual)
except IndexError as exc:
    raise Exception('Lista de anos é menor que o valor escolhido. ' + \
                    'Espera-se um valor entre 0 e ' + \
                    str(len(anos) - 1)
                    )
except Exception as exc:
    raise exc

```

### 3.4. Revisitando a motivação

```

In [ ]: faturas = []

with open(file='./telecom.csv', mode='r', encoding='utf8') as arquivo:
    linha = arquivo.readline()
    linha = arquivo.readline()
    while linha:
        try:
            fatura = float(linha.strip().split(sep=',')[-3])
        except ValueError:
            print('Falha ao processar as faturas! Abortando o processamento.')
            break
        else:
            faturas.append(fatura)
            linha = arquivo.readline()

print(faturas)

```

```

In [ ]: from functools import reduce

def processar_faturas(nome_arquivo: str):

    faturas = []

    with open(file=nome_arquivo, mode='r', encoding='utf8') as arquivo:
        linha = arquivo.readline()
        linha = arquivo.readline()
        while linha:
            try:
                fatura = float(linha.strip().split(sep=',')[-3])
            except ValueError as exc:
                raise ValueError(f'Falha ao processar as faturas ' + \
                                f'devido ao seguinte erro: "{exc}"'
                                )
            else:
                faturas.append(fatura)
                linha = arquivo.readline()

    total_a_pagar = reduce(lambda x, y: x + y, faturas)
    total_a_pagar = round(total_a_pagar, 2)

    return total_a_pagar

```

```

In [ ]: try:
    total_a_pagar = processar_faturas(nome_arquivo='./telecom.csv')
except Exception as exc:
    print(exc)

```

```
else:  
    print(total_a_pagar)
```