



escola
britânica de
artes criativas
& tecnologia

Módulo 08 | Python: Tratamento de Erros

Caderno de **Exercícios**

Professor [André Perez](#)

▼ Tópicos

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

▼ Exercícios

▼ 1. Erros de sintaxe

Identifique o erro de sintaxe nos trechos de código abaixo e corrija-os para que o trecho de código funcione.

- Laços de repetição.

```
credito = {'123': 750, '456': 812, '789': 980}
```

```
for chave, valor in credito.items():  
    print(f'Para o documento {chave}, o valor do escore de crédito é {valor}.")
```

- Funções

```
def pi() --> float:
    return 3.14159265359

pi = pi()
print(pi)
```

- Programação Funcional

```
emails = ['andre.perez@gmail.com', 'andre.perez@live.com', 'andre.perez@yahoo.com']
provedor_da_google = lambda email: 'gmail' in email

emails_google = filter(provedor_da_google, emails)
print(list(emails_google))
```

- Programação orientação a objetos

```
class Pessoa(object):

    def __init__(self nome: str, idade: int, documento: str):
        self.nome = nome
        self.idade = idade
        self.documento = documento

andre = Pessoa(nome="Andre", idade=30, document="123")
```

▼ 2. Erros em tempo de execução

Neste exercício vamos trabalhar com o arquivo csv com dados de crédito, definido abaixo. Execute cada uma das células de código para escrever os arquivos na sua máquina virtual.

```
%writefile credito.csv
id_vendedor,valor_emprestimos,quantidade_emprestimos,data
104271,448.0,1,20161208
21476,826.7,3,20161208
87440,313.6,3,20161208
15980,808.0,6,20161208
215906,2212.0,5,20161208
33696,2771.3,2,20161208
33893,2240.0,3,20161208
214946,"4151.0",18,20161208
123974,2021.95,2,20161208
225870,4039.0,2,20161208
```

O código abaixo deve calcular o total emprestado por cada vendedor mas está "estourando" a exceção `ValueError` devido a um erro no conjunto de dados. Utilize a estrutura `try-catch` para garantir que o código seja executado com sucesso.

Atenção: Você não deve alterar o arquivo de dados.

Dicas:

1. Identifique o bloco de código que pode gerar a exceção e utilize `try` e `except` de modo que a operação que pode causar o problema seja colocada dentro do bloco `try`, e o código que trata a exceção seja escrito dentro do bloco `except`.
2. Tratar a exceção no `except`: utilize o método `replace()` para remover as aspas do conjunto de dados `'linha_elementos[1]'`.

Resumo sobre o método `replace()`:

O método `replace()` é usado para substituir determinado conteúdo de uma string. Esse método recebe 2 argumentos obrigatórios: o 1º corresponde ao valor original que será substituído e o 2º corresponde ao novo valor inserido.

Na prática, o interpretador Python vai percorrer a string e, assim que encontrar o valor correspondente ao 1º argumento vai substituir o conteúdo do 1º argumento pelo conteúdo do 2º argumento.

Sintaxe:

```
replace('valor que será substituído', 'novo valor inserido')
```

Exemplo - aplicação método `replace()`:

```
frase2 = 'Módulos finalizados: 1, 2, 3, *4*, *5*, *6* e 7'
```

```
# Usar o replace() para trocar o caractere asterisco pelo caractere aspas na string "fr
print(frase2.replace('*', ''))
```

```
Módulos finalizados: 1, 2, 3, "4", "5", "6" e 7
```

Obs: Através do `replace()` para remover um caractere, o método vai substituir cada caractere por vazio.

```
def valor_total_emprestimo(valor: float, quantidade: int) -> float:
    return valor * quantidade
```

```
        valor, quantidade
```

```
emprestimos = []
```

```
with open(file='./credito.csv', mode='r', encoding='utf8') as fp:
```

```
    fp.readline() # cabeçalho
```

```
    linha = fp.readline()
```

```
    while linha:
```

```
        linha_emprestimo = {}
```

```
        linha_elementos = linha.strip().split(sep=',')
```

```
        linha_emprestimo['id_vendedor'] = linha_elementos[0]
```

```
        linha_emprestimo['valor_emprestimos'] = float(linha_elementos[1])
```

```
        linha_emprestimo['quantidade_emprestimos'] = int(linha_elementos[2])
```

```
        linha_emprestimo['data'] = linha_elementos[3]
```

```
        emprestimos.append(linha_emprestimo)
```

```
        linha = fp.readline()
```

```
emprestimos_total = []
```

```
for emprestimo in emprestimos:
```

```
    valor_total = valor_total_emprestimo(valor=emprestimo['valor_emprestimos'], quantidade
```

```
    emprestimos_total.append({emprestimo['id_vendedor']: valor_total})
```

```
for emprestimo_total in emprestimos_total:
```

```
    print(emprestimo_total)
```

O resultado final deve ser a impressão da seguinte lista:

```
{'104271': 448.0}
```

```
{'21476': 2480.1000000000004}
```

```
{'87440': 940.8000000000001}
```

```
{'15980': 4848.0}
```

```
{'215906': 11060.0}
```

```
{'33696': 5542.6}
```

```
{'33893': 6720.0}
```

```
{'214946': 74718.0}
```

```
{'123974': 4043.9}
```

```
{'225870': 8078.0}
```
