

5a3c8074-0943-4c32-8d67-0dbd0afc55c3

January 5, 2025

1 Revisão de código

Olá!

Meu nome é Suelen. Estou feliz em revisar seu projeto hoje!

Quando vejo um erro pela primeira vez, apenas aponto. Deixarei você encontrá-lo e corrigi-lo sozinho. Além disso, ao longo do texto, farei algumas observações sobre melhorias no código e também farei comentários sobre suas percepções sobre o assunto. Mas se você ainda não consegue lidar com essa tarefa, darei uma dica mais precisa na próxima iteração e também alguns exemplos práticos. Estarei aberta a feedbacks e discussões sobre o tema.

Você pode encontrar meus comentários em caixas verdes, amarelas ou vermelhas como estas:

Comentário:

Sucesso. Tudo está correto.

Comentário:

Observações. Algumas recomendações.

Comentário:

O bloco requer algumas correções. O trabalho não pode ser aceito com os comentários vermelhos.

Você pode me responder usando isto:

Resposta do aluno

[]:

Comentário geral:

Parabéns pela conclusão do projeto, Vitor!

O seu trabalho apresenta uma qualidade excelente e **todas as questões foram respondidas corretamente!** Como pontos positivos, destaco:

- Bom uso dos recursos das lições
- Comentários pertinentes à análise
- Projeto bem estruturado em sessões

O projeto está aprovado!

Uma empresa de comércio eletrônico, Store 1, começou recentemente a coletar dados sobre seus clientes. O objetivo da Store 1 é entender melhor o comportamento dos clientes e tomar decisões baseadas em dados para melhorar experiência online deles.

Como parte da equipe analítica, sua primeira tarefa é avaliar a qualidade de uma amostra de dados coletados e preparar elas para análises futuras.

2 Quiz

A Store 1 visa garantir a consistência na coleta de dados. Como parte desse esforço, a qualidade dos dados coletados sobre os usuários precisa ser avaliada. Foi pedido que você revise os dados coletados e proponha alterações. Abaixo, você verá dados sobre um determinado usuário. Revise os dados e identifique possíveis problemas.

```
[ ]: user_id = '32415'
      user_name = ' mike_reed '
      user_age = 32.0
      fav_categories = ['ELECTRONICS', 'SPORT', 'BOOKS']
```

Opções:

1. O tipo de dados de `user_id` deve ser alterado de string para número inteiro (integer).
2. A variável `user_name` contém uma string com espaçamento desnecessário e um sublinhado entre o nome e o sobrenome.
3. O tipo de dados de `user_age` está incorreto.
4. A lista `fav_categories` contém strings em letras maiúsculas. Em vez disso, devemos converter os valores da lista para letras minúsculas.

Escreva na célula Markdown abaixo o número de opções que você identificou como problemas. Se você identificou vários problemas, separe o número por vírgulas. Por exemplo, se você acha que os números 1 e 3 estão incorretos, escreva 1, 3, e explique o motivo.

Escreva sua resposta e explique seu raciocínio:

1. Falsa

É falsa pois o `user_id` deve ser escrito como uma string, pois não usamos aquele número em operações aritméticas.

2. Verdadeira

É verdadeira porque quando for printado (`print`) o código, terá um espaçamento desnecessário na resolução e o sublinhado devemos retirar para ficar uma apresentação mais justa e formal ao cliente.

3. Verdadeira

É verdadeira pois a idade de uma pessoa não é um número decimal (float) e sim um número inteiro (int).

4. Verdadeira

É verdadeira pois as categorias deveriam ser escritas em letras minúsculas. #porém fiquei em dúvida se realmente é necessário escrever em letra minúscula ou se poderia deixar do jeito que está...

Comentário:

Concordo com as observações do aluno.

3 Tarefa 1

Vamos implementar as mudanças que identificamos. Primeiro, queremos corrigir os problemas com a variável `user_name`. Como verificamos, ela possui espaços desnecessários e um sublinhado como separador entre o nome e o sobrenome. Seu objetivo é remover os espaços e depois substituir o sublinhado por espaço.

```
[12]: user_name = ' mike_reed '  
user_name = user_name.strip()# remova os espaços na string original  
user_name = user_name.replace('_', ' ') # substitua o sublinhado por espaço  
  
print(user_name)
```

mike reed

4 Tarefa 2

Em seguida, precisamos dividir o `user_name` atualizado em duas substrings para obter uma lista que contém dois valores: a string para o nome e a string para o sobrenome.

```
[13]: user_name = 'mike reed'  
name_split = user_name.split()# divida a string user_name aqui  
  
print(name_split)
```

['mike', 'reed']

5 Tarefa 3

Ótimo! Agora queremos trabalhar com a variável `user_age`. Como mencionamos antes, ela possui um tipo de dados incorretos. Vamos corrigir esse problema transformando o tipo de dados e imprimindo o resultado final.

```
[17]: user_age = 32.0  
user_age = int(user_age)# altere o tipo de dados da idade de um usuário  
  
print(user_age)
```

32

6 Tarefa 4

Como sabemos, os dados nem sempre são perfeitos. Temos que considerar cenários em que o valor de `user_age` não pode ser convertido em um número inteiro. Para evitar que nosso sistema falhe, devemos tomar medidas com antecedência.

Escreva um código que tenta converter a variável `user_age` em um número inteiro e atribua o valor transformado a `user_age_int`. Se a tentativa falhar, vamos exibir uma mensagem solicitando que o usuário forneça sua idade como um valor numérico com a mensagem: **Forneça sua idade como um valor numérico.**

```
[18]: user_age = 'thirty two' # esta é a variável que armazena a idade como uma
      ↪ string.

      try:
          print(int(user_age))
      except:
          print('Forneça sua idade como um valor numérico.')
```

escreva um código que vai tentar transformar user_age em um número inteiro e, se falhar, vai imprimir a mensagem especificada

Forneça sua idade como um valor numérico.

Comentário:

Bom uso dos comandos `try` e `except`

7 Tarefa 5

Por fim, observe que todas as categorias de favoritos são armazenadas em letras maiúsculas. Para preencher uma nova lista chamada `fav_categories_low` com as mesmas categorias, mas em letras minúsculas, repita os valores na lista `fav_categories`, os modifique e anexe os novos valores à lista `fav_categories_low`. Como sempre, imprima o resultado final.

```
[19]: fav_categories = ['ELECTRONICS', 'SPORT', 'BOOKS']
      fav_categories_low = []

      fav_categories_low = [category.lower() for category in fav_categories]
```

escreva seu código aqui

não remova a instrução de impressão abaixo

```
print(fav_categories_low)
```

```
['electronics', 'sport', 'books']
```

8 Tarefa 6

Conseguimos informações adicionais sobre os hábitos de consumo de nossos usuários, incluindo o valor gasto em cada uma de suas categorias favoritas. A administração está interessada nas

seguintes métricas:

- Valor total gasto pelo usuário
- Valor mínimo gasto
- Valor máximo gasto

Vamos calcular e imprimir esses valores:

```
[16]: fav_categories_low = ['electronics', 'sport', 'books']
      spendings_per_category = [894, 213, 173]

      total_amount = sum(spendings_per_category) # escreva seu código aqui
      max_amount = max(spendings_per_category) # escreva seu código aqui
      min_amount = min(spendings_per_category) # escreva seu código aqui

      # não remova a instrução de impressão abaixo
      print(total_amount)
      print(max_amount)
      print(min_amount)
```

1280

894

173

9 Tarefa 7

A empresa quer oferecer descontos aos seus clientes fiéis. Clientes que fizerem compras totalizando mais de \$1.500 são considerados fiéis e vão receber um desconto.

Nosso objetivo é criar um ciclo `while` que verifique o valor total gasto e pare quando ele for atingido. Para simular novas compras, a variável `new_purchase` gera um número entre 30 e 80 em cada ciclo. Isso representa a quantidade de dinheiro gasto em uma nova compra, e é o que você precisa adicionar ao total.

Assim que o valor alvo for atingido e o ciclo `while` for encerrado, o valor final será impresso.

```
[25]: from random import randint

      total_amount_spent = 1280
      target_amount = 1500

      while total_amount_spent < target_amount: # escreva seu código aqui
          new_purchase = randint(30, 80) # geramos um número aleatório de 30 a 80
          total_amount_spent += new_purchase # escreva seu código aqui

      print(total_amount_spent)
```

1525

10 Tarefa 8

Agora temos todas as informações sobre um cliente da maneira que queremos. A administração de uma empresa nos pediu para encontrar uma maneira de resumir toda a informação sobre um usuário. Seu objetivo é criar uma string formatada que usa informações das variáveis `user_id`, `user_name` e `user_age`.

Aqui está a string final que queremos criar: Usuário 32415 chama-se mike e tem 32 anos.

```
[2]: user_id = '32415'
user_name = ['mike', 'reed']
user_age = 32

user_info = 'Usuário {} chama-se {} e tem {} anos'.
↳ format(user_id, user_name[0], user_age) # escreva seu código aqui

# não remova a instrução de impressão abaixo
print(user_info)
```

Usuário 32415 chama-se mike e tem 32 anos

Como você já deve saber, as empresas coletam e armazenam dados de uma maneira específica. A Store 1 deseja armazenar todas as informações sobre seus clientes em uma tabela.

user_id	user_name	user_age	purchase_category	spending_per_category
'32415'	'mike', 'reed'	32	'electronics', 'sport', 'books'	894, 213, 173
'31980'	'kate', 'morgan'	24	'clothes', 'shoes'	439, 390

Em termos técnicos, uma tabela é simplesmente uma lista aninhada que possui uma sublista para cada usuário.

A Store 1 criou essa tabela para seus usuários. Ela está armazenada na variável `users`. Cada sublista contém o ID do usuário, nome e sobrenome, idade, categorias favoritas e o valor gasto em cada categoria.

11 Tarefa 9

Para calcular a receita da empresa, siga estas etapas:

1. Use um ciclo `for` para iterar na lista `users`.
2. Extraia a lista de gastos de cada usuário e some os valores.
3. Atualize o valor da receita com o total de cada usuário.

Isso vai fornecer a receita total da empresa, que você vai imprimir no final.

```
[32]: users = [
    # este é o começo da primeira sublista
    ['32415', ['mike', 'reed'], 32, ['electronics', 'sport', 'books'],
```

```

        [894, 213, 173]
    ], # este é o fim da primeira sublista

    # este é o começo da segunda sublista
    ['31980', ['kate', 'morgan'], 24, ['clothes', 'shoes'],
        [439, 390]
    ] # este é o fim da segunda sublista
]

revenue = 0

for user in users:
    spendings_list = user[4] # extraia a lista de gastos para cada usuário e
    ↪ some os valores
    total_spending = sum(spendings_list) # some os gastos em todas as
    ↪ categorias para obter um total para um usuário específico
    revenue += total_spending # atualize a receita

# não remova a instrução de impressão abaixo
print(revenue)

```

2109

12 Tarefa 10

Use um ciclo for para percorrer a lista de usuários que fornecemos e imprima os nomes dos clientes com menos de 30 anos.

```

[77]: users = [
    ['32415', ['mike', 'reed'], 32, ['electronics', 'sport', 'books'],
        [894, 213, 173]],
    ['31980', ['kate', 'morgan'], 24, ['clothes', 'books'], [439,
        390]],
    ['32156', ['john', 'doe'], 37, ['electronics', 'home', 'food'],
        [459, 120, 99]],
    ['32761', ['samantha', 'smith'], 29, ['clothes', 'electronics',
        'beauty'], [299, 679, 85]],
    ['32984', ['david', 'white'], 41, ['books', 'home', 'sport'], [234,
        329, 243]],
    ['33001', ['emily', 'brown'], 26, ['beauty', 'home', 'food'], [213,
        659, 79]],
    ['33767', ['maria', 'garcia'], 33, ['clothes', 'food', 'beauty'],
        [499, 189, 63]],
    ['33912', ['jose', 'martinez'], 22, ['sport', 'electronics', 'home'
        ], [259, 549, 109]],
    ['34009', ['lisa', 'wilson'], 35, ['home', 'books', 'clothes'],
        [329, 189, 329]],

```

```

    ['34278', ['james', 'lee'], 28, ['beauty', 'clothes', 'electronics'
    ], [189, 299, 579]],
]

client_filtered = []

for user in users:
    if user[2] < 30:
        client_filtered.append(user)

for user in client_filtered:
    name = " ".join(user[1])
    print(name)

```

```

kate morgan
samantha smith
emily brown
jose martinez
james lee

```

13 Tarefa 11

Vamos juntar as tarefas 9 e 10 e imprimir os nomes de usuários com menos de 30 anos com gastos totais acima de 1.000 dólares.

```

[78]: users = [
    ['32415', ['mike', 'reed'], 32, ['electronics', 'sport', 'books'],
    [894, 213, 173]],
    ['31980', ['kate', 'morgan'], 24, ['clothes', 'books'], [439,
    390]],
    ['32156', ['john', 'doe'], 37, ['electronics', 'home', 'food'],
    [459, 120, 99]],
    ['32761', ['samantha', 'smith'], 29, ['clothes', 'electronics',
    'beauty'], [299, 679, 85]],
    ['32984', ['david', 'white'], 41, ['books', 'home', 'sport'], [234,
    329, 243]],
    ['33001', ['emily', 'brown'], 26, ['beauty', 'home', 'food'], [213,
    659, 79]],
    ['33767', ['maria', 'garcia'], 33, ['clothes', 'food', 'beauty'],
    [499, 189, 63]],
    ['33912', ['jose', 'martinez'], 22, ['sport', 'electronics', 'home'
    ], [259, 549, 109]],
    ['34009', ['lisa', 'wilson'], 35, ['home', 'books', 'clothes'],
    [329, 189, 329]],
    ['34278', ['james', 'lee'], 28, ['beauty', 'clothes', 'electronics'
    ], [189, 299, 579]],

```



```

    ]

client_filtered = []

for user in users:
    if user[2] < 30 and sum(user[4]) > 1000:
        client_filtered.append(user)

for user in client_filtered:
    name = ' '.join(user[1])
    print(name)

```

```

samantha smith
james lee

```

14 Tarefa 12

Agora vamos imprimir o nome e a idade de todos os usuários que compraram roupas (clothes). Imprima o nome e a idade na mesma instrução de impressão.

```

[83]: users = [
    ['32415', ['mike', 'reed'], 32, ['electronics', 'sport', 'books'],
    [894, 213, 173]],
    ['31980', ['kate', 'morgan'], 24, ['clothes', 'books'], [439,
    390]],
    ['32156', ['john', 'doe'], 37, ['electronics', 'home', 'food'],
    [459, 120, 99]],
    ['32761', ['samantha', 'smith'], 29, ['clothes', 'electronics',
    'beauty'], [299, 679, 85]],
    ['32984', ['david', 'white'], 41, ['books', 'home', 'sport'], [234,
    329, 243]],
    ['33001', ['emily', 'brown'], 26, ['beauty', 'home', 'food'], [213,
    659, 79]],
    ['33767', ['maria', 'garcia'], 33, ['clothes', 'food', 'beauty'],
    [499, 189, 63]],
    ['33912', ['jose', 'martinez'], 22, ['sport', 'electronics', 'home'
    ], [259, 549, 109]],
    ['34009', ['lisa', 'wilson'], 35, ['home', 'books', 'clothes'],
    [329, 189, 329]],
    ['34278', ['james', 'lee'], 28, ['beauty', 'clothes', 'electronics'
    ], [189, 299, 579]],
    ]

client = []

```

```
for user in users:
    categories = user[3]
    if 'clothes' in categories:
        client.append(user)

for user in client:
    name = ' '.join(user[1])
    age = user[2]
    print(name, age)
```

```
kate morgan 24
samantha smith 29
maria garcia 33
lisa wilson 35
james lee 28
```

Comentário:

Bom trabalho usando comandos `for` `if`

`#`Escreva quaisquer comentários ou ideias finais aqui