

Laboratório 03: Listas

Questões do capítulo 03 e 04 do livro **Curso Intensivo de Python**, de *Eric Matthes*.

Crie um arquivo .py específico para cada questão.

1. **Nomes:** Armazene os nomes de alguns de seus amigos em uma lista chamada `names`. Exiba o nome de cada pessoa acessando cada elemento da lista, um de cada vez.
2. **Saudações:** Comece com a lista usada no exercício anterior, mas em vez de simplesmente exibir o nome de cada pessoa, apresente uma mensagem a elas. O texto de cada mensagem deve ser o mesmo, porém cada mensagem deve estar personalizada com o nome da pessoa.
3. **Lista de convidados:** Se pudesse convidar alguém, vivo ou morto, para o jantar, quem você convidaria? Crie uma lista que inclua pelo menos três pessoas que você gostaria de convidar para jantar. Em seguida, utilize sua lista para exibir uma mensagem para cada pessoa, convidando-a para jantar.
4. **Alterando a lista de convidados:** Você acabou de saber que um de seus convidados não poderá comparecer ao jantar, portanto será necessário enviar um novo conjunto de convites. Você deverá pensar em outra pessoa para convidar.
 - Comece com seu programa do exercício anterior. Acrescente uma instrução `print` no final de seu programa, especificando o nome do convidado que não poderá comparecer.
 - Modifique sua lista, substituindo o nome do convidado que não poderá comparecer pelo nome da nova pessoa que você está convidando.
 - Exiba um segundo conjunto de mensagens com o convite, uma para cada pessoa que continua presente em sua lista.
5. **Mais convidados:** Você acabou de encontrar uma mesa de jantar maior, portanto agora tem mais espaço disponível. Pense em mais três convidados para o jantar.
 - Comece com seu programa do Exercício 3 ou 4. Acrescente uma instrução `print` no final de seu programa informando às pessoas que você encontrou uma mesa de jantar maior.
 - Utilize `insert()` para adicionar um novo convidado no início de sua lista.
 - Utilize `insert()` para adicionar um novo convidado no meio de sua lista.
 - Utilize `append()` para adicionar um novo convidado no final de sua lista.

Exiba um novo conjunto de mensagens de convite, uma para cada pessoa que está em sua lista.

6. **Reduzindo a lista de convidados:** Você acabou de descobrir que sua nova mesa de jantar não chegará a tempo para o jantar e tem espaço para somente dois convidados.
 - Comece com seu programa do exercício anterior. Acrescente uma nova linha que mostre uma mensagem informando que você pode convidar apenas duas pessoas para o jantar.
 - Utilize `pop()` para remover os convidados de sua lista, um de cada vez, até que apenas dois nomes permaneçam em sua lista. Sempre que remover um nome de sua lista, mostre uma mensagem a essa pessoa, permitindo que ela saiba que você sente muito por não poder convidá-la para o jantar.
 - Apresente uma mensagem para cada uma das duas pessoas que continuam na lista, permitindo que elas saibam que ainda estão convidadas.
 - Utilize `del` para remover os dois últimos nomes de sua lista, de modo que você tenha uma lista vazia. Mostre sua lista para garantir que você realmente tem uma lista vazia no final de seu programa.
7. **Conhecendo o mundo:** Pense em pelo menos cinco lugares do mundo que você gostaria de visitar.
 - Armazene as localidades em uma lista. Certifique-se de que a lista não esteja em ordem alfabética.
 - Exiba sua lista na ordem original. Não se preocupe em exibir a lista de forma elegante; basta exibi-la como uma lista Python pura.
 - Utilize `sorted()` para exibir sua lista em ordem alfabética, sem modificar a lista propriamente dita.
 - Mostre que sua lista manteve sua ordem original exibindo-a.
 - Utilize `sorted()` para exibir sua lista em ordem alfabética inversa sem alterar a ordem da lista original.
 - Mostre que sua lista manteve sua ordem original exibindo-a novamente.
 - Utilize `reverse()` para mudar a ordem de sua lista. Exiba a lista para mostrar que sua ordem mudou.
 - Utilize `reverse()` para mudar a ordem de sua lista novamente. Exiba a lista para mostrar que ela voltou à sua ordem original.
 - Utilize `sort()` para mudar sua lista de modo que ela seja armazenada em ordem alfabética. Exiba a lista para mostrar que sua ordem mudou.
 - Utilize `sort()` para mudar sua lista de modo que ela seja armazenada em ordem alfabética inversa. Exiba a lista para mostrar que sua ordem mudou.
8. **Convidados para o jantar:** Reescreva o programa dos exercícios 5 ou 6 de modo a usar `len()` para exibir uma mensagem informando o número

de pessoas que você está convidando para o jantar.

9. **Erro proposital:** Se você ainda não recebeu um erro de índice em um de seus programas, tente fazer um erro desse tipo acontecer. Altere um índice em um de seus programas de modo a gerar um erro de índice. Não se esqueça de corrigir o erro antes de fechar o programa.
10. **Pizzas:** Pense em pelo menos três tipos de pizzas favoritas. Armazene os nomes dessas pizzas e, então, utilize um laço **for** para exibir o nome de cada pizza.
 - Modifique seu laço **for** para mostrar uma frase usando o nome da pizza em vez de exibir apenas o nome dela. Para cada pizza, você deve ter uma linha na saída contendo uma frase simples como *Gosto de pizza de pepperoni*.
 - Acrescente uma linha no final de seu programa, fora do laço **for**, que informe quanto você gosta de pizza. A saída deve ser constituída de três ou mais linhas sobre os tipos de pizza que você gosta e de uma frase adicional, por exemplo, *Eu realmente adoro pizza!*
11. **Contando até vinte:** Use um laço **for** para exibir os números de 1 a 20, incluindo-os.
12. **Um milhão:** Crie uma lista de números de um a um milhão e, então, use um laço **for** para exibir os números. (Se a saída estiver demorando demais, interrompa pressionando **CTRL-C** ou feche a janela de saída.)
13. **Somando um milhão:** Crie uma lista de números de um a um milhão e, em seguida, use **min()** e **max()** para garantir que sua lista realmente começa em um e termina em um milhão. Além disso, utilize a função **sum()** para ver a rapidez com que Python é capaz de somar um milhão de números.
14. **Números ímpares:** Use o terceiro argumento da função **range()** para criar uma lista de números ímpares de 1 a 20. Utilize um laço **for** para exibir todos os números.
15. **Três:** Crie uma lista de múltiplos de 3, de 3 a 30. Use um laço **for** para exibir os números de sua lista.
16. **Cubos:** Um número elevado à terceira potência é chamado de cubo. Por exemplo, o cubo de 2 é escrito como **2**3** em Python. Crie uma lista dos dez primeiros cubos (isto é, o cubo de cada inteiro de 1 a 10), e utilize um laço **for** para exibir o valor de cada cubo.
17. **Comprehension de cubos:** Use uma list comprehension para gerar uma lista dos dez primeiros cubos.
18. **Fatias:** Escolha um dos programas que você desenvolveu nos exercícios anteriores e acrescente algumas linhas no final dele para fazer o seguinte:

- Exibir a mensagem *Os três primeiros itens da lista são:*. Em seguida, use uma fatia para exibir os três primeiros itens da lista desse programa.
 - Exibir a mensagem *Três itens do meio da lista são:*. Use uma fatia para exibir três itens do meio da lista.
 - Exibir a mensagem *Os três últimos itens da lista são:*. Use uma fatia para exibir os três últimos itens da lista.
19. **Minhas pizzas, suas pizzas:** Comece com seu programa do Exercício 10. Faça uma cópia da lista de pizzas e chame-a de `pizzas_amigo`. Então faça o seguinte:
- Adicione uma nova pizza à lista original.
 - Adicione uma pizza diferente à lista `pizzas_amigo`.
 - Prove que você tem duas listas diferentes. Exiba a mensagem *Minhas pizzas favoritas são:*; em seguida, utilize um laço `for` para exibir a primeira lista. Exiba a mensagem *As pizzas favoritas de meu amigo são:*; em seguida, utilize um laço `for` para exibir a segunda lista. Certifique-se de que cada pizza nova esteja armazenada na lista apropriada.
20. **Buffet:** Um restaurante do tipo buffet oferece apenas cinco tipos básicos de comida. Pense em cinco pratos simples e armazene-os em uma tupla.
- Use um laço `for` para exibir cada prato oferecido pelo restaurante.
 - Tente modificar um dos itens e certifique-se de que Python rejeita a mudança.
 - O restaurante muda seu cardápio, substituindo dois dos itens com pratos diferentes. Acrescente um bloco de código que reescreva a tupla e, em seguida, use um laço `for` para exibir cada um dos itens do cardápio revisado.
21. **PEP 8:** Observe o guia de estilo da PEP 8 original em <https://python.org/dev/peps/pep-0008/>. Você não usará boa parte dele agora, mas pode ser interessante dar uma olhada.
22. **Revisão de código:** Escolha três programas que você escreveu neste capítulo e modifique-os para que estejam de acordo com a PEP 8:
- Use quatro espaços para cada nível de indentação. Configure seu editor de texto para inserir quatro espaços sempre que a tecla `TAB` for usada, caso ainda não tenha feito isso (consulte o Apêndice B do livro Curso Intensivo de Python para ver instruções sobre como fazê-lo).
 - Use menos de 80 caracteres em cada linha e configure seu editor para que mostre uma linha vertical na posição do caractere de número 80.
 - Não use linhas em branco em demasia em seus arquivos de programa.