

FAÇA VOCÊ MESMO

4.1 – 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!*

4.2 – Animais: Pense em pelo menos três animais diferentes que tenham uma característica em comum. Armazene os nomes desses animais em uma lista e, então, utilize um laço **for** para exibir o nome de cada animal.

- Modifique seu programa para exibir uma frase sobre cada animal, por exemplo, *Um cachorro seria um ótimo animal de estimação*.
- Acrescente uma linha no final de seu programa informando o que esses animais têm em comum. Você poderia exibir uma frase como *Qualquer um desses animais seria um ótimo animal de estimação!*

4.3 – Contando até vinte: Use um laço **for** para exibir os números de 1 a 20, incluindo-os.

4.4 – 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.)

4.5 – 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.

4.6 – 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.

4.7– Três: Crie uma lista de múltiplos de 3, de 3 a 30. Use um laço **for** para exibir os números de sua lista.

4.8 – 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.

4.9 – Comprehension de cubos: Use uma list comprehension para gerar uma lista dos dez primeiros cubos.

FAÇA VOCÊ MESMO

4.10 – Fatias: Usando um dos programas que você escreveu neste capítulo, acrescente várias linhas no final do programa que façam o seguinte:

- Exiba a mensagem *Os três primeiros itens da lista são:* Em seguida, use uma fatia para exibir os três primeiros itens da lista desse programa.
- Exiba a mensagem *Três itens do meio da lista são:*. Use uma fatia para exibir três itens do meio da lista.
- Exiba a mensagem *Os três últimos itens da lista são:*. Use uma fatia para exibir os três últimos itens da lista.

4.11 – Minhas pizzas, suas pizzas: Comece com seu programa do Exercício 4.1 (página 97). Faça uma cópia da lista de pizzas e chame-a de `friend_pizzas`. Então faça o seguinte:

- Adicione uma nova pizza à lista original.
- Adicione uma pizza diferente à lista `friend_pizzas`.
- 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.

4.12 – Mais laços: Todas as versões de `foods.py` nesta seção evitaram usar laços `for` para fazer exibições a fim de economizar espaço. Escolha uma versão de `foods.py` e escreva dois laços `for` para exibir cada lista de comidas.

FAÇA VOCÊ MESMO

4.13 – 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.