

8.1 – Mensagem: Escreva uma função chamada `display_message()` que mostre uma frase informando a todos o que você está aprendendo neste capítulo. Chame a função e certifique-se de que a mensagem seja exibida corretamente.

8.2 – Livro favorito: Escreva uma função chamada `favorite_book()` que aceite um parâmetro `title`. A função deve exibir uma mensagem como **Um dos meus livros favoritos é Alice no país das maravilhas**. Chame a função e não se esqueça de incluir o título do livro como argumento na chamada da função.

8.3 – Camiseta: Escreva uma função chamada `make_shirt()` que aceite um tamanho e o texto de uma mensagem que deverá ser estampada na camiseta. A função deve exibir uma frase que mostre o tamanho da camiseta e a mensagem estampada.

Chame a função uma vez usando argumentos posicionais para criar uma camiseta. Chame a função uma segunda vez usando argumentos nomeados.

8.4 – Camisetas grandes: Modifique a função `make_shirt()` de modo que as camisetas sejam grandes por default, com uma mensagem *Eu amo Python*. Crie uma camiseta grande e outra média com a mensagem default, e uma camiseta de qualquer tamanho com uma mensagem diferente.

8.5 – Cidades: Escreva uma função chamada `describe_city()` que aceite o nome de uma cidade e seu país. A função deve exibir uma frase simples, como **Reykjavik está localizada na Islândia**. Forneça um valor default ao parâmetro que representa o país. Chame sua função para três cidades diferentes em que pelo menos uma delas não esteja no país default.

8.9 – Mágicos: Crie uma lista de nomes de mágicos. Passe a lista para uma função chamada `show_magicians()` que exiba o nome de cada mágico da lista.

8.10 – Grandes mágicos: Comece com uma cópia de seu programa do Exercício 8.9. Escreva uma função chamada `make_great()` que modifique a lista de mágicos acrescentando a expressão *o Grande* ao nome de cada mágico. Chame `show_magicians()` para ver se a lista foi realmente modificada.

8.11 – Mágicos inalterados: Comece com o trabalho feito no Exercício 8.10. Chame a função `make_great()` com uma cópia da lista de nomes de mágicos. Como a lista original não será alterada, devolva a nova lista e armazene-a em uma lista separada. Chame `show_magicians()` com cada lista para mostrar que você tem uma lista de nomes originais e uma lista com a expressão *o Grande* adicionada ao nome de cada mágico.

8.12 – Sanduíches: Escreva uma função que aceite uma lista de itens que uma pessoa quer em um sanduíche. A função deve ter um parâmetro que agrupe tantos itens quantos forem fornecidos pela chamada da função e deve apresentar um resumo do sanduíche pedido. Chame a função três vezes usando um número diferente de argumentos a cada vez.

8.13 – Perfil do usuário: Comece com uma cópia de *user_profile.py*, da página 210. Crie um perfil seu chamando **build_profile()**, usando seu primeiro nome e o sobrenome, além de três outros pares chave-valor que o descrevam.

8.14 – Carros: Escreva uma função que armazene informações sobre um carro em um dicionário. A função sempre deve receber o nome de um fabricante e um modelo. Um número arbitrário de argumentos nomeados então deverá ser aceito. Chame a função com as informações necessárias e dois outros pares nome-valor, por exemplo, uma cor ou um opcional. Sua função deve ser apropriada para uma chamada como esta:

```
car = make_car('subaru', 'outback', color='blue', tow_package=True)
```

Mostre o dicionário devolvido para garantir que todas as informações foram armazenadas corretamente.

8.6 – Nomes de cidade: Escreva uma função chamada **city_country()** que aceite o nome de uma cidade e seu país. A função deve devolver uma string formatada assim:

"Santiago, Chile"

Chame sua função com pelo menos três pares cidade-país e apresente o valor devolvido.

8.7 – Álbum: Escreva uma função chamada **make_album()** que construa um dicionário descrevendo um álbum musical. A função deve aceitar o nome de um artista e o título de um álbum e deve devolver um dicionário contendo essas duas informações. Use a função para criar três dicionários que representem álbuns diferentes. Apresente cada valor devolvido para mostrar que os dicionários estão armazenando as informações do álbum corretamente.

Acrescente um parâmetro opcional em **make_album()** que permita armazenar o número de faixas em um álbum. Se a linha que fizer a chamada incluir um valor para o número de faixas, acrescente esse valor ao dicionário do álbum. Faça pelo menos uma nova chamada da função incluindo o número de faixas em um álbum.

8.8 – Álbuns dos usuários: Comece com o seu programa do Exercício 8.7. Escreva um laço **while** que permita aos usuários fornecer o nome de um artista e o título de um álbum. Depois que tiver essas informações, chame **make_album()** com as entradas do usuário e apresente o dicionário criado. Lembre-se de incluir um valor de saída no laço **while**.