

FAÇA VOCÊ MESMO

5.1 – Testes condicionais: Escreva uma série de testes condicionais. Exiba uma frase que descreva o teste e o resultado previsto para cada um. Seu código deverá ser semelhante a:

```
car = 'subaru'
print("Is car == 'subaru'? I predict True.")
print(car == 'subaru')

print("\nIs car == 'audi'? I predict False.")
print(car == 'audi')
```

- Observe atentamente seus resultados e certifique-se de que compreende por que cada linha é avaliada como **True** ou **False**.
- Crie pelo menos dez testes. Tenha no mínimo cinco testes avaliados como **True** e outros cinco avaliados como **False**.

5.2 – Mais testes condicionais: Você não precisa limitar o número de testes que criar em dez. Se quiser testar mais comparações, escreva outros testes e acrescente-os em *conditional_tests.py*. Tenha pelo menos um resultado **True** e um **False** para cada um dos casos a seguir:

- testes de igualdade e de não igualdade com strings;
- testes usando a função **lower()**;
- testes numéricos que envolvam igualdade e não igualdade, maior e menor que, maior ou igual a e menor ou igual a;
- testes usando as palavras reservadas **and** e **or**;
- testes para verificar se um item está em uma lista;
- testes para verificar se um item não está em uma lista.

dizendo que ela é um(a) garoto(a).

- Se a pessoa tiver pelo menos 13 anos, mas menos de 20, mostre uma mensagem dizendo que ela é um(a) adolescente.
- Se a pessoa tiver pelo menos 20 anos, mas menos de 65, mostre uma mensagem dizendo que ela é adulto.
- Se a pessoa tiver 65 anos ou mais, mostre uma mensagem dizendo que essa pessoa é idoso.

5.7 – Fruta favorita: Faça uma lista de suas frutas favoritas e, então, escreva uma série de instruções **if** independentes que verifiquem se determinadas frutas estão em sua lista.

- Crie uma lista com suas três frutas favoritas e chame-a de **favorite_fruits**.
- Escreva cinco instruções **if**. Cada instrução deve verificar se uma determinada fruta está em sua lista. Se estiver, o bloco **if** deverá exibir uma frase, por exemplo, *Você realmente gosta de bananas!*

5.8 – Olá admin: Crie uma lista com cinco ou mais nomes de usuários, incluindo o nome **'admin'**. Suponha que você esteja escrevendo um código que exibirá uma saudação a cada usuário depois que eles fizerem login em um site. Percorra a lista com um laço e mostre uma saudação para cada usuário:

- Se o nome do usuário for **'admin'**, mostre uma saudação especial, por exemplo, *Olá admin, gostaria de ver um relatório de status?*
- Caso contrário, mostre uma saudação genérica, como *Olá Eric, obrigado por fazer login novamente.*

5.9 – Sem usuários: Acrescente um teste **if** em **hello_admin.py** para garantir que a lista de usuários não esteja vazia.

- Se a lista estiver vazia, mostre a mensagem *Precisamos encontrar alguns usuários!*
- Remova todos os nomes de usuário de sua lista e certifique-se de que a mensagem correta seja exibida.

5.10 – Verificando nomes de usuários: Faça o seguinte para criar um programa que simule o modo como os sites garantem que todos tenham um nome de usuário único.

- Crie uma lista chamada **current_users** com cinco ou mais nomes de usuários.

- Crie outra lista chamada **new_users** com cinco nomes de usuários. Garanta que um ou dois dos novos usuários também estejam na lista **current_users**.
- Percorra a lista **new_users** com um laço para ver se cada novo nome de usuário já foi usado. Em caso afirmativo, mostre uma mensagem informando que a pessoa deverá fornecer um novo nome. Se um nome de usuário não foi usado, apresente uma mensagem dizendo que o nome do usuário está disponível.
- Certifique-se de que sua comparação não levará em conta as diferenças entre letras maiúsculas e minúsculas. Se 'John' foi usado, 'JOHN' não deverá ser aceito.

5.11 – Números ordinais: Números ordinais indicam sua posição em uma lista, por exemplo, *1st* ou *2nd*, em inglês. A maioria dos números ordinais nessa língua termina com *th*, exceto 1, 2 e 3.

- Armazene os números de 1 a 9 em uma lista.
- Percorra a lista com um laço.
- Use uma cadeia **if-elif-else** no laço para exibir a terminação apropriada para cada número ordinal. Sua saída deverá conter "**1st 2nd 3rd 4th 5th 6th 7th 8th 9th**", e cada resultado deve estar em uma linha separada.

5.12 – Estilizando instruções **if:** Revise os programas que você escreveu neste capítulo e certifique-se de que os testes condicionais foram estilizados de forma apropriada.

5.13 – Suas ideias: A essa altura, você é um programador mais capacitado do que era quando começou a ler este livro. Agora que você tem melhor noção de como situações do mundo real são modeladas em programas, talvez esteja pensando em alguns problemas que poderia resolver com seus próprios programas. Registre qualquer ideia nova que tiver sobre problemas que você queira resolver à medida que suas habilidades em programação continuam a melhorar. Considere jogos que você queira escrever, conjuntos de dados que possa querer explorar e aplicações web que gostaria de criar.