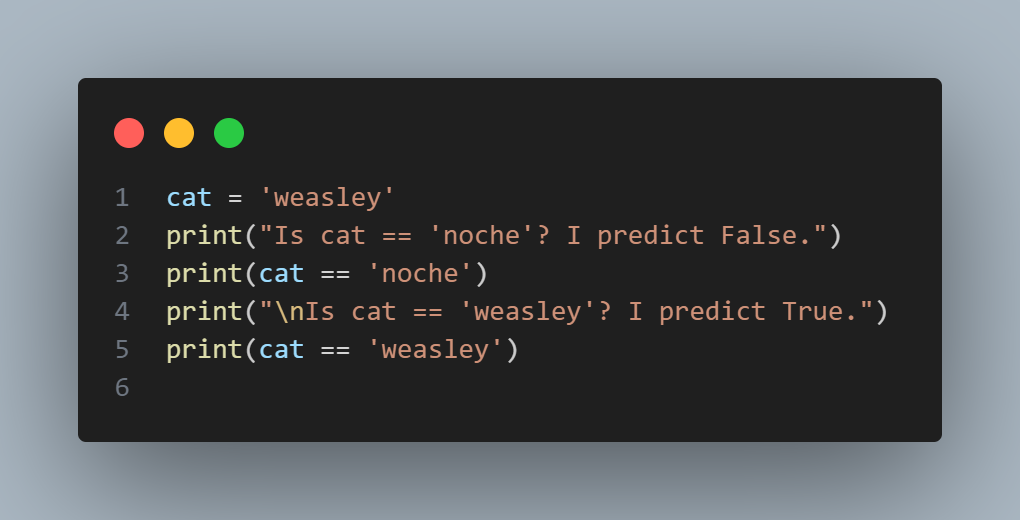
**Exercise 5: If Statements Solutions**

1. Write a series of conditional tests. Print a statement describing each test and your prediction for the results of each test. Your code should look something like this:



• Look closely at your results, and make sure you understand why each line evaluates to True or False.

• Create at least 10 tests. Have at least 5 tests evaluate to True and another 5 tests evaluate to False.

1. Write an if-elif-else chain that determines a person’s stage of life. Set a value for the variable age, and then:

• If the person is less than 2 years old, print a message that the person is a baby. •If the person is at least 2 years old but less than 4, print a message that the person is a toddler.

• If the person is at least 4 years old but less than 13, print a message that the person is a kid.

• If the person is at least 13 years old but less than 20, print a message that the person is a teenager.

• If the person is at least 20 years old but less than 65, print a message that the person is an adult.

• If the person is age 65 or older, print a message that the person is an elder.

1. Make a list of your favorite fruits, and then write a series of independent if statements that check for certain fruits in your list.

• Make a list of your three favorite fruits and call it favorite\_fruits.

• Write five if statements. Each should check whether a certain kind of fruit is in your list. If the fruit is in your list, the if block should print a statement, such as You really like bananas!

1. Make a list of five or more usernames, including the name 'admin'. Imagine you are writing code that will print a greeting to each user after they log in to a website. Loop through the list, and print a greeting to each user:

• If the username is 'admin', print a special greeting, such as Hello admin, would you like to see a status report?

• Otherwise, print a generic greeting, such as Hello Eric, thank you for logging in again

**Stretch and Challenge:** add on an if test to make sure the list of users is not empty.

•If the list is empty, print the message We need to find some users!

• Remove all of the usernames from your list, and make sure the correct message is printed

1. Do the following to create a program that simulates how websites ensure that everyone has a unique username.

• Make a list of five or more usernames called current\_users.

• Make another list of five usernames called new\_users. Make sure one or two of the new usernames are also in the current\_users list.

• Loop through the new\_users list to see if each new username has already been used. If it has, print a message that the person will need to enter a new username. If a username has not been used, print a message saying that the username is available.

• Make sure your comparison is case insensitive. If 'John' has been used, 'JOHN' should not be accepted.

**Stretch and Challenge**

* Ordinal numbers indicate their position in a list, such as 1st or 2nd. Most ordinal numbers end in th, except 1, 2, and 3.

• Store the numbers 1 through 9 in a list.

• Loop through the list.

• Use an if-elif-else chain inside the loop to print the proper ordinal ending for each number. Your output should read "1st 2nd 3rd 4th 5th 6th 7th 8th 9th", and each result should be on a separate line.

* At this point, you’re a more capable programmer than you were when you started this course. Now that you have a better sense of how real-world situations are modeled in programs, you might be thinking of some problems you could solve with your own programs. Record below any new ideas you have about problems you might want to solve as your programming skills continue to improve. Consider games you might want to write, data sets you might want to explore, and web applications you’d like to create. One of your ideas could be a project you can work on once you are in the post-programme phase!