C5

if Statements

CHAPTER ASSIGNMENT

Name _____ Class _____ No. _____ Date _____

Answer every question in the space given. Show your working. You may use the chapter page to help you.

Questions from *Python Crash Course*, 2nd edition, Eric Matthes (No Starch Press), for classroom use.

Try It Yourself

5-1 **Conditional Tests:** 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:

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

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

• Create at least ten tests. Have at least five tests evaluate to `True` and another five tests evaluate to `False`.

5-2 **More Conditional Tests:** You don't have to limit the number of tests you create to ten. If you want to try more comparisons, write more tests and add them to *conditional_tests.py*. Have at least one `True` and one `False` result for each of the following:

- Tests for equality and inequality with strings
- Tests using the `lower()` method
- Numerical tests involving equality and inequality, greater than and less than, greater than or equal to, and less than or equal to
- Tests using the `and` keyword and the `or` keyword
- Test whether an item is in a list
- Test whether an item is not in a list

5-3 **Alien Colors #1:** Imagine an alien was just shot down in a game. Create a variable called `alien_color` and assign it a value of `'green'`, `'yellow'`, or `'red'`.

- Write an `if` statement to test whether the alien's color is green. If it is, print a message that the player just earned 5 points.
- Write one version of this program that passes the `if` test and another that fails. (The version that fails will have no output.)

5-4 **Alien Colors #2:** Choose a color for an alien as you did in Exercise 5-3, and write an `if-else` chain.

• If the alien's color is green, print a statement that the player just earned 5 points for shooting the alien.

• If the alien's color isn't green, print a statement that the player just earned 10 points.

• Write one version of this program that runs the `if` block and another that runs the `else` block.

5-5 **Alien Colors #3:** Turn your `if-else` chain from Exercise 5-4 into an `if-elif-else` chain.

• If the alien is green, print a message that the player earned 5 points.

• If the alien is yellow, print a message that the player earned 10 points.

• If the alien is red, print a message that the player earned 15 points.

• Write three versions of this program, making sure each message is printed for the appropriate color alien.

5-6 **Stages of Life:** 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.

5-7 **Favorite Fruit:** 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!*

5-8 **Hello Admin:** 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 Jaden, thank you for logging in again.*

5-9 **No Users:** Add an `if` test to *hello_admin.py* 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.

5-10 **Checking Usernames:** 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. (To do this, you'll need to make a copy of `current_users` containing the lowercase versions of all existing users.)

5-11 **Ordinal Numbers:** 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.

5-12 **Styling `if` statements:** Review the programs you wrote in this chapter, and make sure you styled your conditional tests appropriately.

5-13 **Your Ideas:** At this point, you're a more capable programmer than you were when you started this book. 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 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.

End of assignment · Check your answers against the chapter at gurujazzy.com/subjects/python-crash-course/chapter-5-if-statements/