Control Statements and Program Development

3

Objectives

In this chapter, you'll:

- Decide whether to execute actions with the statements `if`, `if`…`else` and `if`…`elif`…`else`.
- Execute statements repeatedly with `while` and `for`.
- Shorten assignment expressions with augmented assignments.
- Use the `for` statement and the built-in `range` function to repeat actions for a sequence of values.
- Perform sentinel-controlled iteration with `while`.
- Learn problem-solving skills: understanding problem requirements, dividing problems into smaller pieces, developing algorithms to solve problems and implementing those algorithms in code.
- Develop algorithms through the process of top-down, stepwise refinement.
- Create compound conditions with the Boolean operators `and`, `or` and `not`.
- Stop looping with `break`.
- Force the next iteration of a loop with `continue`.
- Use some functional-style programming features to write scripts that are more concise, clearer, easier to debug and easier to parallelize.

Note: Throughout the Instructor Solutions Manual, solutions are ***not provided*** for project, research and challenge exercises—many of which are substantial and appropriate for term projects, directed-study projects, capstone-course projects and thesis topics. **Before assigning a particular exercise for homework, instructors should check the IRC to be sure the solution is available.** These Instructor Solutions Manual PDFs contain **only answers to short-answer exercises and any discussion questions asked in other exercises.** Code corresponding to programming exercises can be found in the **`solutions`** folder's chapter-specific subfolder—e.g., **`ch01`** for Chapter1, **`ch02`** for Chapter 2, etc. Code generally is provided both in **Python source-code files** (**`.py`**) and **Jupyter Notebooks** (**`.ipynb`**).

Exercises

Unless specified otherwise, use IPython sessions for each exercise.

3.2 *(What's Wrong with This Code?)* What is wrong with the following code?

```
a = b = 7
print('a =', a, '\nb =', b)
```

First, answer the question, then check your work in an IPython session.

Answer: There is nothing wrong with this code, though you'd normally use two separate assignment statements to assign `7` to `a` and `b`.

3.3 *(What Does This Code Do?)* What does the following program print?

```
for row in range(10):
    for column in range(10):
        print('<' if row % 2 == 1 else '>', end='')
    print()
```

Answer: It displays the following output:

```
>>>>>>>>>>
<<<<<<<<<<
>>>>>>>>>>
<<<<<<<<<<
>>>>>>>>>>
<<<<<<<<<<
>>>>>>>>>>
<<<<<<<<<<
>>>>>>>>>>
<<<<<<<<<<
```

3.4 *(Fill in the Missing Code)* In the code below

```
for ***:
    for ***:
        print('@')  # ERROR: This should be print('@', end='')
    print()
```

replace the *** so that when you execute the code, it displays two rows, each containing seven `@` symbols, as in:

```
@@@@@@@
@@@@@@@
```

Answer:

```
for row in range(2):
    for column in range(7):
        print('@', end='')
    print()
```

3.6 *(Turing Test)* The great British mathematician Alan Turing proposed a simple test to determine whether machines could exhibit intelligent behavior. A user sits at a computer and does the same text chat with a human sitting at a computer and a computer operating by itself. The user doesn't know if the responses are coming back from the human or the independent computer. If the user can't distinguish which responses are coming from the human and which are coming from the computer, then it's reasonable to say that the computer is exhibiting intelligence.

Create a script that plays the part of the independent computer, giving its user a simple medical diagnosis. The script should prompt the user with `'What is your problem?'` When the user answers and presses *Enter*, the script should simply ignore the user's input, then prompt the user again with `'Have you had this problem before (yes or no)?'` If the user enters `'yes'`, print `'Well, you have it again.'` If the user answers `'no'`, print `'Well, you have it now.'`

Would this conversation convince the user that the entity at the other end exhibited intelligent behavior? Why or why not?

Answer: It's really up to each individual to say whether this is convincing or not, but most people would say the computer is not displaying intelligent behavior.

3.28 *(Intro to Data Science: Mean, Median and Mode)* Calculate the mean, median and mode of the values 9, 11, 22, 34, 17, 22, 34, 22 and 40. Suppose the values included another 34. What problem might occur?

Answer: If you add another 34 to the list of numbers, the `statistics` module's `mode` function will raise a `StatisticsError`, because there will be two most common values.

3.29 *(Intro to Data Science: Problem with the Median)* For an odd number of values, to get the median you simply arrange them in order and take the middle value. For an even number, you average the two middle values. What problem occurs if those two values are different?

Answer: In this case, the median value will not be an actual value in the dataset.

3.30 *(Intro to Data Science: Outliers)* In statistics, outliers are values out of the ordinary and possibly way out of the ordinary. Sometimes, outliers are simply bad data. In the data science case studies, we'll see that outliers can distort results. Which of the three measures of central tendency we discussed—mean, median and mode—is most affected by outliers? Why? Which of these measures are not affected or least affected? Why?

Answer: Outliers would always change the mean, but they may have little or no affect on the median and mode. For the median, the middle value could be the depending on where the outliers are. For the mode, the value could be the same since outliers are not likely to be the most common value.

3.31 *(Intro to Data Science: Categorical Data)* Mean, median and mode work well with numerical values. You can use them in calculations and arrange them in meaningful order. Categorical values are descriptive names like Boxer, Poodle, Collie, Beagle, Bulldog and

Chihuahua. Normally, you don't use these in calculations nor associate an order with them. Which if any of the descriptive statistics are appropriate for categorical data?

Answer: The mode would be appropriate for categorical data. You can use it to determine the most common value.