- Python uses type str to represent text as sequences of characters.
- Strings are created by placing pairs of single or double quotes around the text. Multiline strings can be created using matching pairs of triple quotes.
- Special characters like newline and tab are represented using escape sequences that begin with a backslash.
- Values can be printed using built-in function print, and input can be provided by the user using built-in function input.

4.7 Exercises

Here are some exercises for you to try on your own. Solutions are available at http://pragprog.com/titles/gwpy2/practical-programming.

1. What value does each of the following expressions evaluate to? Verify your answers by typing the expressions into the Python shell.
   a. 'Computer' + 'Science'
   b. 'Darwin's'
   c. 'H2O' * 3
   d. 'CO2' * 0

2. Express each of the following phrases as Python strings using the appropriate type of quotation marks (single, double, or triple) and, if necessary, escape sequences. There is more than one correct answer for each of these phrases.
   a. They'll hibernate during the winter.
   b. "Absolutely not," he said.
   c. "He said, 'Absolutely not,'" recalled Mel.
   d. hydrogen sulfide
   e. left\right

3. Rewrite the following string using single or double quotes instead of triple quotes:
   "A
   B
   C"

4. Use built-in function len to find the length of the empty string.

5. Given variables x and y, which refer to values 3 and 12.5, respectively, use function print to print the following messages. When numbers appear in the messages, variables x and y should be used.
Chapter 4. Working with Text • 76

a. The rabbit is 3.
b. The rabbit is 3 years old.
c. 12.5 is average.
d. 12.5 * 3
e. 12.5 * 3 is 37.5.

6. Consider this code:
>>> first = 'John'
>>> last = 'Doe'
>>> print(last + ', ' + first)

What is printed by the code above?

7. Use input to prompt the user for a number, store the number entered as a float in a variable named num, and then print the contents of num.

8. Complete the examples in the docstring and then write the body of the following function:

def repeat(s, n):
    """(str, int) -> str"

    Return s repeated n times; if n is negative, return the empty string.

    >>> repeat('yes', 4)
    'yesyesyesyes'
    >>> repeat('no', 0)
    >>> repeat('no', -2)
    >>> repeat('yesnomaybe', 3)

    """

9. Complete the examples in the docstring and then write the body of the following function:

def total_length(s1, s2):
    """(str, str) -> int"

    Return the sum of the lengths of s1 and s2.

    >>> total_length('yes', 'no')
    5
    >>> total_length('yes', '')
    >>> total_length('YES!!!!', 'Nooooo')

    """