11   Looping

Looping statements are used when repeating sequences of instructions. We'll practise using **for** and **while** statements in this section.

→ Create a new file and copy the code below into the le using the Spyder editor:

```
myword = raw_input( 'enter a word: ' )
print 'you entered: ' + myword
L = len( myword )
print 'your word is ' + str(L) + ' letters long.'
for i in range( 0, L ):
    print 'the ' + str(i) + 'th letter is ' + myword[i]
```

Run the program. What does it do?

→ Modify the program as follows:

1. Before the loop, define a counter variable and set its initial value to 0, like this: counter = 0

2. Inside the loop, check if each letter in the word is a vowel (A, E, I, O or U). If it is, then increment the counter, like this:
   counter = counter + 1

3. After the loop, display the number of vowels found in the word.

Save your program, test and debug it.

→ Create a new file in the Spyder editor, and write a new program as follows:

1. Use a condition-controlled loop (e.g., while).

2. Inside the loop, ask the user to enter a number.

3. Display the number that the user entered.

4. If the user enters the number 99, then exit the loop.

5. After the loop, display the word "done".

Save your program, test and debug it.

→ Modify your program so that if the user enters the word quit, the loop will exit and, as before, the program will display the word "done" and then exit. Save your modified code, test and debug it.

→ Modify your program so that it computes the total of all the numbers entered by the user, and then displays that total before the program exits. Save your modified code, test and debug it.

→ Go back to the last program you wrote in the previous section (simulating drawing cards from a deck). Modify your program so that you draw 5 cards from the deck, using a loop.

Note that because you are not checking to see if you've picked the same card twice, this is simulating drawing 5 cards "with replacement", meaning that it's like you draw a card, look at it (display its value) and then put it back in the deck before you draw another card.

Think about how you might keep track of which cards have been drawn so that you can simulate drawing cards from a deck "without replacement." In the next Python session, you'll learn an elegant technique for doing this.