Using the Formatter

Formatters add a placeholder for inputting a data value into a string output using the curly brace {}. The format method from the str class is invoked to receive the value as a parameter. The number of parameters in the format method should match the number of placeholders in the string representation. Other format specifiers can be added with the placeholder curly brackets.

```
print("{} {} {}".format(a, b, c))
'Output': I Love You
# re-ordering the output
print("{2} {1} {0}".format(a, b, c))
'Output': You Love I
```

Control Structures

Programs need to make decisions which result in executing a particular set of instructions or a specific block of code repeatedly. With control structures, we would have the ability to write programs that can make logical decisions and execute an instruction set until a terminating condition occurs.

The if/elif (else-if) Statements

The if/elif (else-if) statement executes a set of instructions if the tested condition evaluates to true. The else statement specifies the code that should execute if none of the previous conditions evaluate to true. It can be visualized by the flowchart in Figure 9-2.

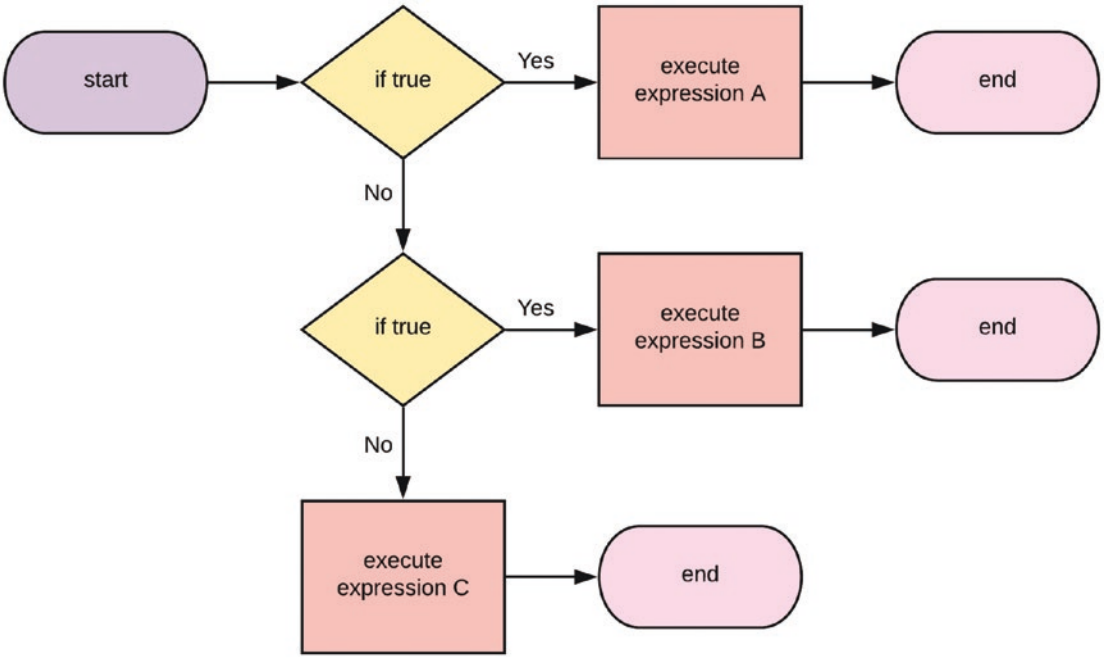

Figure 9-2. *Flowchart of the if statement*

The syntax for the if/elif statement is given as follows:

```
if expressionA:
    statementA
elif expressionB:
    statementB
...
...
else:
    statementC
```

Here is a program example:

```
a = 8
if type(a) is int:
    print('Number is an integer')
elif a > 0:
    print('Number is positive')
```

```
else:
    print('The number is negative and not an integer')

'Output': Number is an integer
```

The while Loop

The while loop evaluates a condition, which, if true, repeatedly executes the set of instructions within the while block. It does so until the condition evaluates to false. The while statement is visualized by the flowchart in Figure 9-3.

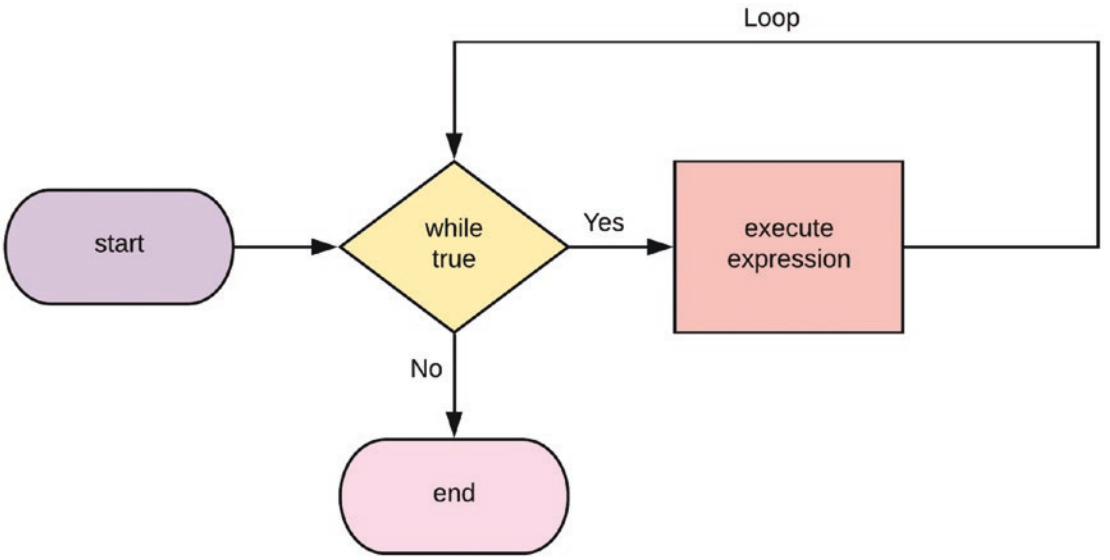

Figure 9-3. *Flowchart of the while loop*

Here is a program example:

```
a = 8
while a > 0:
    print('Number is', a)

    # decrement a
    a -= 1
```