We can rewrite this as

```
[expression for item in iterable]
```

Let's have some program examples.

```
squares = []
for elem in range(0,5):
    squares.append((elem+1)**2)

squares
'Output': [1, 4, 9, 16, 25]
```

The preceding code can be concisely written as

```
[(elem+1)**2 for elem in range(0,5)]
'Output': [1, 4, 9, 16, 25]
```

This is even more elegant in the presence of nested control structures.

```
evens = []
for elem in range(0,20):
    if elem % 2 == 0 and elem != 0:
        evens.append(elem)

evens
'Output': [2, 4, 6, 8, 10, 12, 14, 16, 18]
```

With list comprehension, we can code this as

```
[elem for elem in range(0,20) if elem % 2 == 0 and elem != 0]
'Output': [2, 4, 6, 8, 10, 12, 14, 16, 18]
```

The break and continue Statements

The break statement terminates the execution of the nearest enclosing loop (for, while loops) in which it appears.

```
for val in range(0,10):
    print("The variable val is:", val)
    if val > 5:
```

```
        print("Break out of for loop")
        break
```

```
'Output': The variable val is: 0
     The variable val is: 1
     The variable val is: 2
     The variable val is: 3
     The variable val is: 4
     The variable val is: 5
     The variable val is: 6
     Break out of for loop
```

The continue statement skips the next iteration of the loop to which it belongs, ignoring any code after it.

```
a = 6
while a > 0:
    if a != 3:
        print("The variable a is:", a)
    # decrement a
    a = a - 1
    if a == 3:
        print("Skip the iteration when a is", a)
        continue
```

```
'Output': The variable a is: 6
     The variable a is: 5
     The variable a is: 4
     Skip the iteration when a is 3
     The variable a is: 2
     The variable a is: 1
```

Functions

A function is a code block that carries out a particular action (Figure 9-5). Functions are called by the programmer when needed by making a **function call**. Python comes pre-packaged with lots of useful functions to simplify programming. The programmer can also write custom functions.