We have already used a few functions earlier. Also, you saw two different types of functions. Functions, like the `show_message` function, only perform certain tasks. Other functions, like `random()` and `get_string()`, return a value that can then be used in expressions or assignments.

When we use a function, we say that we call the function. A function call consists of the name of the function, followed by the arguments, separated by commas, in parentheses. Even when a function has no arguments, you must still use the parentheses. Arguments can be values as well as expressions. Most functions have a fixed number of arguments, but some can have an arbitrary number of arguments.

Let’s look at some examples. GML provides a large number of functions for drawing objects. These functions should normally only be used in the `draw` event or objects. There are functions to draw shapes, sprites, backgrounds, text, and so on. If you have registered your version of Game Maker, you also have access to lots of additional drawing functions for creating colorized shapes, rotated text, and even three-dimensional objects. To use some drawing functions, create a script with the following code:

```gml
draw_set_color(c_red);
draw_rectangle(x-50,y-50,x+50,y+50,false);
draw_set_color(c_blue);
draw_circle(x,y,40,false);
```

This piece of code first calls a function to set the drawing color. `c_red` is a built-in value that indicates the red color. Next, this code draws a rectangle with the indicated corners. The fifth argument, which has the value `false`, indicates that this must not be an outlined rectangle. Next, the color is set to blue, and finally a filled circle is drawn. To use this script, create an object (it does not need a sprite), and then add a `Draw` event and include an `Execute Script` action to call the script. Add a number of instances of the object to a room and check out the result. You can find the program in the file `Games/Chapter12/draw_shapes.gmk` on the CD.

As a second example, let’s consider the creation of instances. Often during games you want to create instances of objects. The function `instance_create()` can be used for doing just this. This function has three arguments. The first two arguments must indicate the position where you want to create the instance and the third argument must indicate the object of which the instance must be created. Assume we want to create a bullet and fire it in the direction of the instance that creates it. This can be achieved with the following piece of code (which assumes that an object `obj_bullet` exists):

```gml
var bullet_instance;
bullet_instance = instance_create(x,y,obj_bullet);
bullet_instance.speed = 12;
bullet_instance.direction = direction;
```