

Repeating things

So far all of our programs run once and then end. However, they could become more interesting and useful if certain sections of the code could be repeated. For example consider this simple 'traffic light' algorithm.

1. Turn on green light
2. Wait 1 minute
3. Turn off green light
4. Turn on amber light
5. Wait 10 seconds
6. Turn off amber light and turn on red light
7. Wait 1 minute
8. Turn on amber light
9. Wait 10 seconds
10. Turn off red light and amber light

The algorithm above serves as a good starting point for a traffic light system but has one major weakness - the code runs once, which is not very useful or safe. What is needed is an instruction that will make the computer repeat the code from lines 1 and 10. Something like this:

```
Repeat a number of times
  a.Turn on green light
  b.Wait 1 minute
  c.Turn off green light
  d.Turn on amber light
  e.Wait 10 seconds
  f.Turn off amber light and turn on
    red light
  g.Wait 1 minute
  h.Turn on amber light
  i.Wait 10 seconds
  j.Turn off red light and amber light
End repeat
```

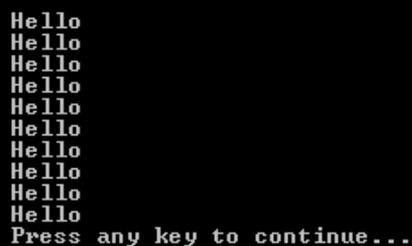
In the previous algorithm, the code between 'Repeat' and 'End Repeat' (lines a to b) will be repeated forever and the highways will be safe once more. In this example I've use the word 'repeat' which is used as a keyword in some languages such as Pascal and Logo. However, in the next section I will introduce Small Basic's keywords for loop structures.

The for loop

Small Basic uses the keywords For, To and EndFor to create loop structures. We use a For Loop structure when we know in advance how many times we would like the code to be repeated. The first line of the loop tells the computer how many times the loop should be repeated and sets up a variable counter to keep track of how many times the loop has been repeated. Here's an example:

```
1 'A very simple looping program.  
2 'It will print "Hello!" 10 times.  
3 For i = 1 To 10  
4   TextWindow.WriteLine("Hello")  
5 EndFor
```

In this example, everything between the For statement and the EndFor statement will be repeated 10 times. In this case it is the statment on line 4 - TextWindow.WriteLine("Hello") - which will be repeated 10 times. The variable i will keep track of the number of times the loop has been executed. Initially the variable i has the value 1 and after each loop it's value will increase



```
Hello  
Press any key to continue...
```

by 1. When i has increased to the value 11 the code will jump out of the loop and continue to the next line after EndFor, in this case the end of the program.

In the example below the value held by the variable `i`, which keeps track of the number of times that the loop is executed, is printed to the screen.

```
1 'This program prints the value of a loop variable
2
3 'Create a loop that will repeat 10 times.
4 For i = 1 To 10
5     TextWindow.WriteLine("We are in the loop because i = " + i)
6 EndFor
7 'Exit loop when i > 10 and continue to line 9..|
8 TextWindow.WriteLine("We are out of the loop because i = " + i)
9
```

```
We are in the loop because i = 1
We are in the loop because i = 2
We are in the loop because i = 3
We are in the loop because i = 4
We are in the loop because i = 5
We are in the loop because i = 6
We are in the loop because i = 7
We are in the loop because i = 8
We are in the loop because i = 9
We are in the loop because i = 10
We are out of the loop because i = 11
Press any key to continue...
```

Notice that in the output that the value held by the variable `i` increases from 1 to 10 as expected, but the loop is exited when the value is greater than 10. This is exactly as expected.



The Program object can be used to pause the execution of a program. The Delay(number) function takes a parameter which specifies in milliseconds how long the program should wait before continuing.



Remember that 1000 milliseconds is the same as 1 second.

In the next example we create a count down program by subtracting the counting variable *i* from 10.

```
1 This program prints a count down.
2
3 Create a loop that will repeat 10 times.
4 For i = 0 To 10
5     TextWindow.WriteLine(10 - i)
6     Wait 1 sec (1000 milliseconds).
7     Program.Delay(1000)
8     Clear the screen.
9     TextWindow.Clear()
10 EndFor
11 Exit the loop and print "Blast Off!"
12 TextWindow.WriteLine("Blast Off!")
```

The loop will write out the numbers 10 to 0 as a count down. However once the counting variable *i* is greater than 10 the program will exit the loop and execute the next statement which in this case prints the words “Blast Off!” on the screen.

While animating

Another common use for loops in computer programs is to create animations. In order to create an animation we need to repeatedly display and remove a picture in quick succession to create the impression of movement. The animation cycle can be represented in algorithm like this:

1. Repeat a number of times
 - a. clear screen
 - b. draw image
 - c. wait a moment
2. End repeat

This time however we will not use the For Loop EndFor construction but a construction called the While EndWhile loop. Unlike a For loop, which knows how many times to repeat in advance, a While loop will keep on repeating until a condition is satisfied.



When using a While loop it is always important to ensure that the condition can evaluate to false at some point in the running of the program. If the condition for the while loop cannot evaluate to false then the program will get stuck in an infinite loop and effectively crash the program.

1. `count = 0`
2. `While count < 10`
 - a. `TextWindow.WriteLine("Hello")`
 - b. `count = count + 1`
3. `EndWhile`

In the program above the variable called `count` is initialised to 0. Then `While count` is less than 10 it will write the word "Hello" on the screen. Then it will increment `count` by one. The program will keep writing the word "Hello" until the variable `count` is greater than 10, at which point the loop will be exited safely.

The program below uses the animation algorithm to create blinking eyes.

```
1 'Blinking Man, this program creates a blinking animation!  
2 'Create a count variable  
3 count = 1  
4 'Create a animation while loop  
5 While count < 11  
6     TextWindow.Clear()  
7     TextWindow.WriteLine("O O")  
8     Program.Delay(200)  
9     TextWindow.Clear()  
10    TextWindow.WriteLine("- -")  
11    Program.Delay(200)  
12    'Increase count by one.  
13    count = count +1  
14 EndWhile
```