Practice 7:
FOR LOOPS

7

For Loop

For Loop is just another type of loop!

What is For Loop for?

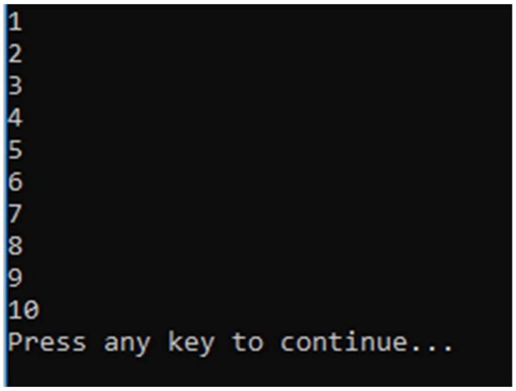
We saw While Loops in the previous section as a way of repeating something—like a command or an action—over and over. Let's look at an easier way to do this: the for loop.

Let's create the Microsoft Small Basic code to print the first 10 natural numbers. We start with 1 and go till 10. This can be done in a loop.

We are going to use the **For** loop to do this. We can write this exact process in Microsoft Small Basic like this:

```
For number = 1 To 10
      TextWindow.WriteLine(number)
EndFor
```

And this is what happens when you run the code from above:



```
1
2
3
4
5
6
7
8
9
10
Press any key to continue...
```

The code above shows a variable called number being declared and given a value of 1 in the For loop itself. Every time the code runs through the loop, *it automatically increments the current value of the variable number*. Then the line of code within the for loop references the variable and prints it out using `TextWindow.WriteLine(number).` You can see this in action in the picture of the code after it runs.

The loop condition is given as a range using the `To` keyword. The variable number is incremented until it reaches the end of the range given in the loop condition. The loop in this case runs 10 times.

Once the loop reaches its end, we use the keyword `EndFor` to signal that the for loop will end.

The variable that is getting assigned the number can be used however you want. You can print it out, use it in a conditional statement, use it in other variables (NewNumber = number* 5), or not use it within the loop at all.

Challenge: Looped Drawings

Now that you know how to use a For Loop, you can program the computer to repeat strings instead of doing it yourself! In this case, you don't use the variable in the For Loop, *you simply use the For Loop to repeat a line of code.*

First, decide how many times you want it to repeat. Then create a string you want to repeat. This could be a pattern: `-*^^*-+-*^^*`, or a face: `(*_*)`. Put this string in the `TextWindow.Write("")`. Don't forget to ask yourself if you want to repeat the string next to each other `TextWindow.Write()(TextWindow.write())`, or every line `(TextWindow.Writeline()TextWindow.WriteLine()).`

Challenge: Fibonacci Sequence

Do you know the Fibonacci sequence? It is a sequence found in nature that can be easily computed with a for loop! It starts 1, 1, 2, 3, 5, 8, 13, 21, … where the next number is the last two numbers added together. Try doing this up to say, 10 digits in the sequence with a for loop in the code editor! If you need more information on the Fibonacci sequence, look at the Fibonacci sequence link in the "additional links" section.

7

Double For Loops

Now that you have mastered For Loops, we can introduce a new topic: Double For Loops! Double For Loops are when one For Loop is nested inside the other. They look like this:

```
number = 1
For row = 1 To 10
      For column = 1 To 10

            TextWindow.Write(number + “ ”)
            number = number + 1
      EndFor
      TextWindow.WriteLine(“”)
EndFor
```

In Double For Loops, we call the Outer Loop with an inner Inside Loop. In a For Loop, everything inside the loop is repeated however many times is specified. In this case, the outer loop goes `1 To 10`, or 10 times. Every time the outer loop is repeated, the **whole** inner loop is called. This means if the outer loop repeats 10 times and the inner loop repeats 10 times, and for every 1 time the outer loop is repeated the inner loop goes through its whole 10 times, the code in the middle is called 100 times, since 10*10 equals 100.

Looking at the double for loop written above, how do you think it would print out on the text window? Type the code into your code editor to try it out.

**Note: If you copy and paste the quotation marks(“ ”), your code might not work. Please type it out. **

Let’s think about the code we just wrote. Notice the variable before the double loop. It starts at 1, what would happen if it started at another number? What happens if you take away “ ” in `TextWindow.Write(number + “ ”)`? Try these changes in the code editor and see what else you can change. Maybe instead of a number variable you can put a string!

Technically, you can put as many for loops as you want inside one another. They are also called 'Nested For Loops'.

Challenge: Looped Drawings continued

One part of the loop we haven't talked about changing is the amount we are looping through (`1 To 10`). You might already have done this, but have you thought about making the inner for loop dependent on the variable in the outer loop?

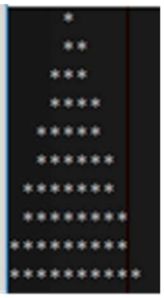
```
For row = 1 To 10

      For column = 1 To row
            TextWindow.Write("*")
      EndFor

      TextWindow.WriteLine("")

EndFor
```

The code makes a triangle! See if you can make a triangle that looks like this:



```
    *
    **
   ***
   ****
  *****
  ******
 *******
 ********
*********
**********
```

Discussion Questions

- What are the advantages of a For Loop over a while loop?
- Is an 'infinite' (never ending) loop possible when we use a For Loop?
- Where might you use a For Loop in real life?

- When might a For Loop not be a good fit for a loop?
- Why would the nested for loop variables be called row and columns? What can you apply nested for loops to, given your answer to the previous question?
- What is a limitation of for loops?

Additional Resources

Loops

- https://aka.ms/SBForLoop1
- https://aka.ms/SBForLoop2

Fibonacci Sequence

- https://www.mathsisfun.com/numbers/fibonacci-sequence.html