Iterations (commonly referred to as Loops)

Loops are statements that execute a statement or set of statements repeatedly until a specified condition is met. Loops consist of an iteration statement (defines how often the code repeats) and a body (set of instructions that will execute once per iteration). A loop's body may contain many statements or it may contain only one. If we aren't careful, we can accidentally cause a loop to never exit. To the user of the program, it looks like the program simply stopped working, but the programmer can recognize this as a condition called an infinite loop.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace Iterations
{
    class Program
    {
        static void Main(string[] args)
        {
            for (int i = 0; i < 10; i++) //Creating the integer variable i and setting it
equal to zero
                //then evaluating to ask is it true that i is less then 10 and lastly
increasing the value by 1 every time that the statement is executed.  It is executing
until the value i < 10 is no longer true.
                //When its no longer true move on to the next line.  It will display 0
through 9 then stop.

            {
                Console.WriteLine(i.ToString());
            }
            Console.ReadLine();
        }
    }
}
```

Always three parts to a loop: index, check condition, increment index

Example of a while loop in JavaScript:

```
var a = 1;              //index

while (a < 10 ) {      //body of loop (check condition)

        alert (a);

        a++;    //increment index- says to add one to the value of a once it gets to 10 the loop will end

}
```

Check out this resource to see several different types of loops you can use:
file:///C:/ProgramData/GGI/DevPro/GDC-1500-000/Content/index.html#/course/1/arc/0/module/6/topic/5

Hint: Code snippets help you write code and be more accurate; type in for then press tab twice to automatically enter: `for (int i = 0; i < length; i++)`