Do While

While, has a slight disadvantage

The previous example has a small disadvantage:

```
string userAnswer = "No";
// Repeat the loop while the user has not answered "Yes", yet.
while(userAnswer != "Yes") // .....
```

We need to initialize userAnswer in line 1, because else, in line 3, we'd access an uninitialized variable when comparing it to "Yes".
This would be invalid.
Therefore, we need to initialize userAnswer with some value.
In this case by using = "No";.

This is due to us checking the answer first and then asking the user.
If only, there was a way of doing it the other way round.
Like this:

```
AskAgain:
Console.WriteLine("Do you want to buy my game? Yes or No?");
string userAnswer = Console.ReadLine();
if(userAnswer != "Yes"){
  goto AskAgain;
}
Console.WriteLine("Thanks for buying my game.");
```

Of course there is!

We can make use of yet another loop called the Do-While loop:

```
string userAnswer;
do {
  Console.WriteLine("Do you want to buy my game? Yes or No?");
  userAnswer = Console.ReadLine();
} while (userAnswer != "Yes");
Console.WriteLine("Thanks for buying my game.");
```

Now we can execute the code before checking the condition!

Same same.. but different

Do-While loops are very similar to **While loops**, but, while **While loops** check the condition first, the **Do-While loop** executes the code and THEN it checks the condition.
Therefore the code in the **Do-While loop**, is guaranteed to execute AT LEAST one time:

```
do { // while-scope-start
  // put the code here, that you want to repeat while the bool-expression is true
} // while-scope-end
while (conditionExpression);
```

Example

To give you an example:

```
int i = 100;

do {
  // First, print the current number
  Console.WriteLine(i);
  i++;
} while (i < 3);
// Then, check, if the current Number is still smaller than 3. If not, interrupt.
```

This time the number 100 was still printed because it was printed first, then afterwards the check **101 < 3** was performed, regardless of how large our number was.

Whereas the **While loop** would first check the condition **100 < 3** and then not print any number at all.

So:

- **Do-While**
    - Executes code first and THEN checks the condition
- **While**
    - Checks the condition first and if true THEN executes the code

Goal - Ask a number

- Create a Function named AskNumberBetween
    - It has two number parameters: min and max
    - It returns a number.
    - It asks the user as many times as necessary to give a number between min and max
    - It returns the result, when it is valid.
- Use above method two times and print the result (the returned value) each time to the console:
    - Once for a number between 1 and 3
    - Once for a number between 41 and 103
- **Use a do while loop for this**