100 XP

Introduction

3 minutes

As we've said several times in previous modules featuring iteration and decision statements, there are several techniques you can use to accomplish similar results. Just like written and spoken languages, in programming languages you can express the same idea in different ways. Even so, each expression may have a nuanced difference in meaning.

The `do-while` and `while` statements allow us to control the flow of code execution by looping through a block of code until a condition is met. When working with the `foreach` statement, we iterate once for each item in sequence, such as an array. The `for` statement allows us to iterate a pre-determined number of times, and control the process of iteration. The `do-while` and `while` statements allow us to iterate through a block of code with the intent that the logic inside of the code block will affect when we can stop iterating.

Suppose you want to accept and process user input. You want to continue accepting and processing input until the user presses the `q` key for "quit". You can use the `do-while` and the `while` statements to keep iterating through the logic to accept user input and process it until the user is ready to stop.

In this module, you'll use the `do-while` statement and the `while` statement to iterate through code block. You'll understand when to choose one over the other. You'll use the `continue` statement to skip processing the remainder of code in the code block and go directly to the Boolean evaluation of the `while` statement.

By the end of this module, you will be able to confidently use the `do-while` and `while` statements to add looping logic to your application.

Learning objectives

In this module, you will:

- Write code that uses the `do-while` statement to iterate through a code block.
- Write code that uses the `while` statement to iterate through a code block.
- Use the `continue` statement to step directly to the Boolean evaluation.

Prerequisites:

- Experience using the `if` statement
- Experience using `foreach` and `for` iteration statements.
- Experience writing Boolean expressions
- Experience generating random numbers using the `System.Random` class and the `Random.Next()` method

Next unit: Exercise - Create do and while iteration loops

Continue >