



Loops

2025-01-31

I. Introduction

In C#, loops are used to repeatedly execute a block of code as long as a specified condition is true. There are several types of loops, each with its specific use cases: while, do-while, for, and foreach. Here's a breakdown of each loop type, along with an exercise for each.

I.1. while loop

The while loop repeatedly executes a block of code as long as the given condition evaluates to true. The condition is checked before each iteration.

Syntax:

```
while (condition)
{
    // Code to be executed
}
```

I.2. do-while loop

The do-while loop works similarly to the while loop, but with one key difference: it checks the condition after the code block is executed, ensuring that the loop runs at least once.

Syntax:

```
do
{
    // Code to be executed
} while (condition);
```

I.3. for loop

The for loop is commonly used when the number of iterations is known beforehand. It consists of three parts: initialization, condition, and iteration.

Syntax:

```
for (initialization; condition; iteration)
{
    // Code to be executed
}
```

I.4. foreach loop

The foreach loop is used to iterate over collections (arrays, lists, etc.) without needing to manage the loop counter manually. It automatically iterates through all elements of the collection.

Syntax:

```
foreach (var item in collection)
{
    // Code to be executed
}
```

II. Exercises

The amount of 🌶️ determines how hard the exercises are.

1. Write a while loop that prints the numbers from 1 to 10. 🌶️
2. Write a do-while loop that asks the user to enter a number until they enter a number greater than 100. 🌶️
3. Write a foreach loop that prints all elements in an array of strings. 🌶️
4. Write a for loop that prints the multiplication table for a number entered by the user (1 to 10). 🌶️🌶️