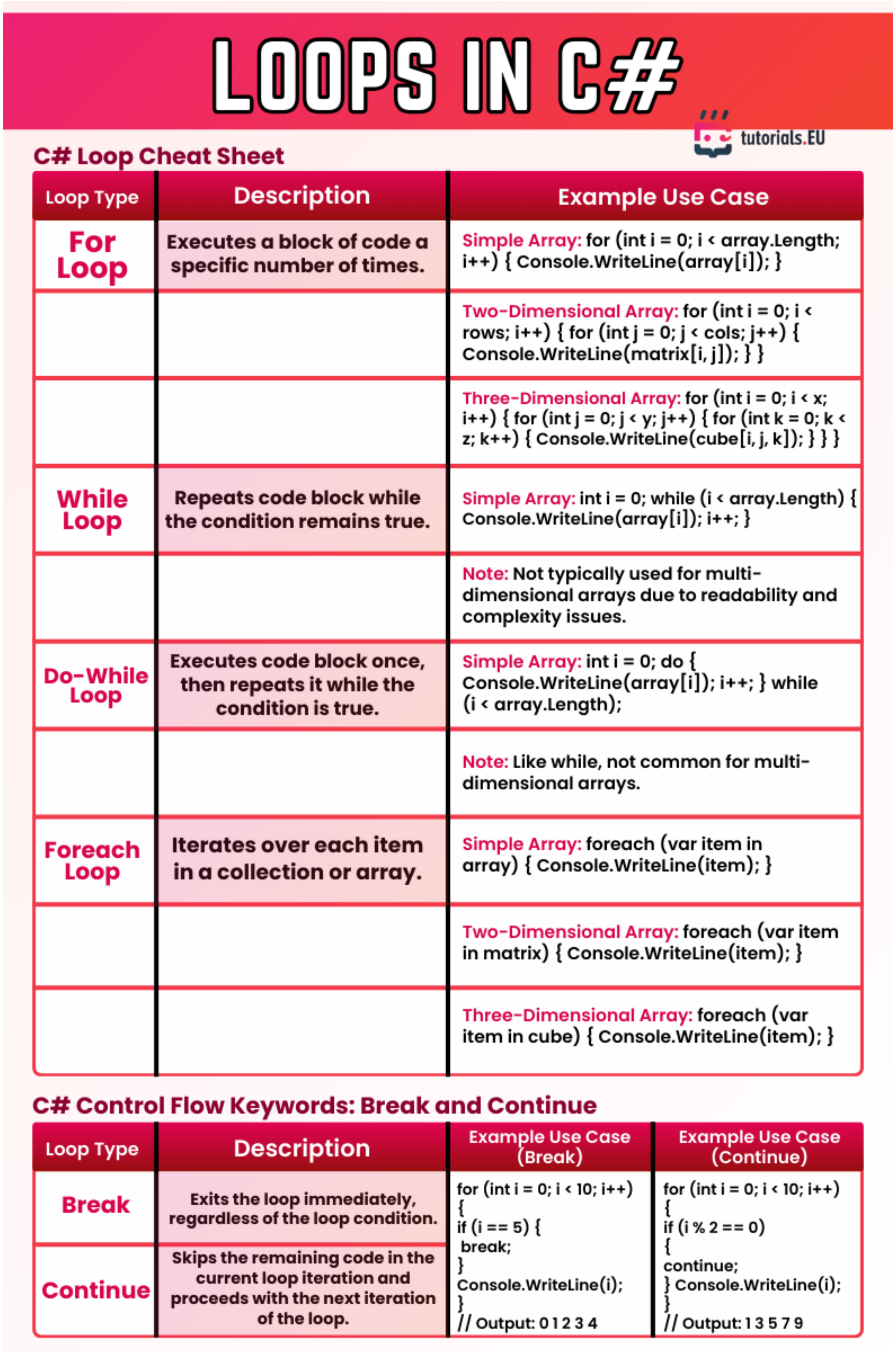
CHEATSHEET - Loops in C#



LOOPS IN C#

tutorials.EU

C# Loop Cheat Sheet

Loop Type	Description	Example Use Case
For Loop	Executes a block of code a specific number of times.	Simple Array: for (int i = 0; i < array.Length; i++) { Console.WriteLine(array[i]); }
		Two-Dimensional Array: for (int i = 0; i < rows; i++) { for (int j = 0; j < cols; j++) { Console.WriteLine(matrix[i, j]); } }
		Three-Dimensional Array: for (int i = 0; i < x; i++) { for (int j = 0; j < y; j++) { for (int k = 0; k < z; k++) { Console.WriteLine(cube[i, j, k]); } } }
While Loop	Repeats code block while the condition remains true.	Simple Array: int i = 0; while (i < array.Length) { Console.WriteLine(array[i]); i++; }
		Note: Not typically used for multi-dimensional arrays due to readability and complexity issues.
Do-While Loop	Executes code block once, then repeats it while the condition is true.	Simple Array: int i = 0; do { Console.WriteLine(array[i]); i++; } while (i < array.Length);
		Note: Like while, not common for multi-dimensional arrays.
Foreach Loop	Iterates over each item in a collection or array.	Simple Array: foreach (var item in array) { Console.WriteLine(item); }
		Two-Dimensional Array: foreach (var item in matrix) { Console.WriteLine(item); }
		Three-Dimensional Array: foreach (var item in cube) { Console.WriteLine(item); }

C# Control Flow Keywords: Break and Continue

Loop Type	Description	Example Use Case (Break)	Example Use Case (Continue)

Break	Exits the loop immediately, regardless of the loop condition.	for (int i = 0; i < 10; i++)
		{
		if (i == 5) {
		break;
		}
		Console.WriteLine(i);
		}
		// Output: 0 1 2 3 4
			for (int i = 0; i < 10; i++)
			{
			if (i % 2 == 0)
			{
			continue;
			} Console.WriteLine(i);
			}
			// Output: 1 3 5 7 9

Continue	Skips the remaining code in the current loop iteration and proceeds with the next iteration of the loop.		