TutorialKart

C# break

C# break

C# break statement is used to break a loop abruptly and come out of the loop.

The syntax of break statement is:

```
break;
```

Example 1 – C# break statement with for loop

In the following example, we use **break** statement to come out of the for loop even before the condition in for loop is evaluated to false.

Program.cs

```
using System;

namespace CSharpExamples {
    class Program {
        static void Main(string[] args) {
            for(int i=0;i<10;i++){
                if(i==5){
                    break;
                }
                Console.WriteLine(i);
            }
        }
    }
}
```

When `i` equals `5`, we are breaking the for loop. The program execution then proceeds with the statement(s) after for statement (if any present).

Output

```
PS D:\workspace\csharp\HelloWorld> dotnet run
0
1
2
3
4
```

```
4
```

Example 2 – C# break statement with while loop

In the following example, we use **break** statement to come out of the while loop even before the condition in for loop is evaluated to false.

Program.cs

```
using System;

namespace CSharpExamples {
    class Program {
        static void Main(string[] args) {
            int i=0;
            while(i<10){
                if(i==5){
                    break;
                }
                Console.WriteLine(i);
                i++;
            }
        }
    }
}
```

when `i` equals `5`, we are breaking the while loop, and the program execution proceeds with the statements after while statement.

Output

```
PS D:\workspace\csharp\HelloWorld> dotnet run
0
1
2
3
4
```

Example 3 – C# break statement with foreach

In the following example, we use **break** statement to come out of the foreach iteration before foreach executes for all the elements in the given array.

Program.cs

```
using System;

namespace CSharpExamples {
    class Program {
        static void Main(string[] args) {
```

```
        static void Main(string[] args) {
            int[] nums = {2, 41, 15, 64};
            foreach(int num in nums){
                if(num==15){
                    break;
                }
                Console.WriteLine(num);
            }
        }
    }
}
```

When num is equal to 15, we are breaking the foreach, and the program execution continues with the rest of statement(s) after foreach block.

Output

```
PS D:\workspace\csharp\HelloWorld> dotnet run
2
41
```

Note

- You can have multiple break statements inside a loop for different conditions.

Summary

In this C# Tutorial, we learned to use C# continue statement to skip the execution of subsequent statements in the loop for a particular state of iteration.

C# Tutorial

- C# Tutorial
- C# Basic Example
- C# Comments
- C# Variables
- C# Constants
- C# if, if-else
- C# switch
- C# while loop
- C# for loop
- C# foreach

C# foreach
C# break
C# continue
C# struct
C# enum
C# String
C# Array
C# Command Line Arguments

C# Console Operations

C# Write to Console
C# Read from Console

C# Object Oriented Programming Concepts

C# Class & Constructors
C# Encapsulation
C# Polymorphism
C# Method Overloading
C# Interfaces

C# String Operations

C# String Length
C# Substring

C# File Operations

C# Read Text File
C# Write to File
C# Delete File
C# Copy File

C# Exception Handling

C# try-catch
C# finally
C# throw

- C# Custom Exception
- C# SystemException
- C# DivideByZeroException
- C# NullReferenceException
- C# InvalidCastException
- C# IOException
- C# FieldAccessException

C# Collections

- C# List
- C# SortedList
- C# HashSet
- C# SortedSet
- C# Stack
- C# Queue
- C# LinkedList
- C# Dicionary
- C# SortedDictionary

C# Errors [Solved]

- C# Error: Class does not contain a constructor that takes arguments