**C# break Statement**

In this tutorial, you will learn about the working C# break statement with the help of examples.

In C#, we use the break statement to terminate the loop.

As we know, loops iterate over a block of code until the test expression is false. However, sometimes we may need to terminate the loop immediately without checking the test expression.

In such cases, the break statement is used. The syntax of break statement is,

break;

Before we learn about break, make sure to learn about

* [for loop](https://www.programiz.com/csharp-programming/for-loop)
* [if...else](https://www.programiz.com/csharp-programming/if-else-statement)
* [while loop](https://www.programiz.com/csharp-programming/do-while-loop)

**Example: C# break statement with for loop**

using System;

namespace CSharpBreak {

class Program {

static void Main(string[] args) {

for (int i = 1; i <= 4; ++i) {

if (i == 3) {

break; //terminates the loop if i equals 3

}

Console.WriteLine(i);

}

Console.ReadLine();

}

}

}

**Output**

1

2

In the above program, our for loop runs **4** times from i = 1 to **4**. However, when i is equal to **3**, the break statement is encountered.

if (i == 3) {

break;

}

Now, the loop is terminated suddenly. So, we only get **1** and **2** as output.

**Note**: The break statement is used with decision-making statements like if…else.

**Example: C# break statement with while loop**

using System;

namespace WhileBreak {

class Program {

static void Main(string[] args) {

int i = 1;

while (i <= 5) {

Console.WriteLine(i);

i++;

if (i == 4) {

// terminates the loop

break;

}

}

Console.ReadLine();

}

}

}

**Output**

1

2

3

In the above example, we have created a while loop that is supposed to run from i = 1 to 5.

However, when i is equal to **4**, the break statement is encountered.

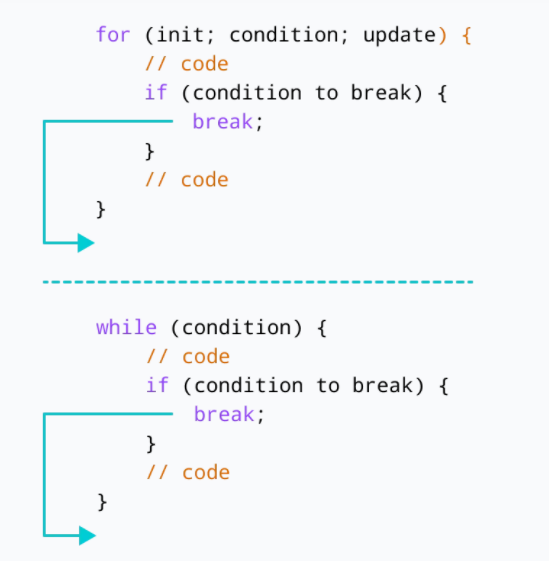
if (i == 4) {

break;

}

Now, the while loop is terminated.

**Working of break statement in C#**

Working of break statement

**break Statement with Nested Loop**

We can also use the break statement with nested loops. For example,

using System;

namespace NestedBreak {

class Program {

static void Main(string[] args) {

int sum = 0;

for(int i = 1; i <= 3; i++) { //outer loop

// inner loop

for(int j = 1; j <= 3; j++) {

if (i == 2) {

break;

}

Console.WriteLine("i = " + i + " j = " +j);

}

}

Console.ReadLine();

}

}

}

**Output**

i = 1 j = 1

i = 1 j = 2

i = 1 j = 3

i = 3 j = 1

i = 3 j = 2

i = 3 j = 3

In the above example, we have used the break statement inside the inner for loop. Here, the break statement is executed when i == 2.

Hence, the value of i = 2 is never printed.

**Note**: The break statement only terminates the inner for loop. This is because we have used the break statement inside the inner loop.

If you want to learn the working of nested loops, visit [C# Nested Loops](https://www.programiz.com/csharp-programming/nested-loops).

**break with foreach Loop**

We can also use the break statement with foreach loops. For example,

using System;

namespace ForEachBreak {

class Program {

static void Main(string[] args) {

int[] num = { 1, 2, 3, 4, 5 };

// use of for each loop

foreach(int number in num) {

// terminates the loop

if(number==3) {

break;

}

Console.WriteLine(number);

}

}

}

}

**Output**

1

2

In the above example, we have created an array with values: **1**, **2**, **3**, **4**, **5**. Here, we have used the foreach loop to print each element of the array.

However, the loop only prints **1** and **2**. This is because when the number is equal to **3**, the break statement is executed.

if (number == 3) {

break;

}

This immediately terminates the [foreach loop](https://www.programiz.com/csharp-programming/foreach-loop).

**break with Switch Statement**

We can also use the break statement inside a switch case statement. For example,

using System;

namespace ConsoleApp1 {

class Program {

static void Main(string[] args) {

char ch='e';

switch (ch) {

case 'a':

Console.WriteLine("Vowel");

break;

case 'e':

Console.WriteLine("Vowel");

break;

case 'i':

Console.WriteLine("Vowel");

break;

case 'o':

Console.WriteLine("Vowel");

break;

case 'u':

Console.WriteLine("Vowel");

break;

default:

Console.WriteLine("Not a vowel");

}

}

}

}

**Output**

Vowel

Here, we have used the break statement inside each case. It helps us to terminate the switch statement when a matching case is found.

To learn more, visit [C# switch statement](https://www.programiz.com/csharp-programming/switch-statement).