**C# ternary (? :) Operator**

In this article, we will learn about C# ternary operator and how to use it to control the flow of program.

Ternary [operator](https://www.programiz.com/csharp-programming/operators) are a substitute for if...else statement. So before you move any further in this tutorial, go through [C# if...else statement](https://www.programiz.com/csharp-programming/if-else-statement) (if you haven't).

The syntax of ternary operator is:

Condition ? Expression1 : Expression2;

The ternary operator works as follows:

* If the expression stated by Condition is true, the result of Expression1 is returned by the ternary operator.
* If it is false, the result of Expression2 is returned.

For example, we can replace the following code

if (number % 2 == 0)

{

isEven = true;

}

else

{

isEven = false;

}

with

isEven = (number % 2 == 0) ? true : false ;

**Why is it called ternary operator?**

This operator takes 3 **operands**, hence called ternary operator.

**Example 1: C# Ternary Operator**

using System;

namespace Conditional

{

class Ternary

{

public static void Main(string[] args)

{

int number = 2;

bool isEven;

isEven = (number % 2 == 0) ? true : false ;

Console.WriteLine(isEven);

}

}

}

When we run the program, the output will be:

True

In the above program, 2 is assigned to a variable number. Then, the ternary operator is used to check if number is even or not.

Since, 2 is even, the expression (number % 2 == 0) returns true. We can also use ternary operator to return numbers, strings and characters.

Instead of storing the return value in variable isEven, we can directly print the value returned by ternary operator as,

Console.WriteLine((number % 2 == 0) ? true : false);

**When to use ternary operator?**

Ternary operator can be used to replace multi lines of code with a single line. However, we shouldn't overuse it.

For example, we can replace the following if…else if code

if (a > b)

{

result = "a is greater than b";

}

else if (a < b)

{

result = "b is greater than a";

}

else

{

result = "a is equal to b";

}

with a single line of code

result = a > b ? "a is greater than b" : a < b ? "b is greater than a" : "a is equal to b";

As we can see, the use of ternary operator may decrease the length of code but it makes us difficult to understand the logic of the code.

Hence, it's better to only use ternary operator to replace simple if else statements.