

How to write conditional statements in C#

Sometimes, you'll want your C# code to run only under certain conditions. The commands and operators in C# below will help you do just that.

Conditional statements allow a program to perform different tasks based on specific conditions. They are usually written as 'if-then' (if-then) commands. Here, a block of code runs only if a certain condition is true.

In C#, you can write conditional statements in a variety of ways, including standard if-else statements, switch statements, ternary operators, and nullable ternary operators.

How to write conditional statements in C#

You have many different ways to write conditional statements in JavaScript, Java, C++, Python, and other languages. In C#, you can start executing these commands in a simple C# application in Visual Studio.

Command if/else - standard if/else

The if statement tests a condition to evaluate its accuracy. If the condition evaluates to true, the code in this block will run. If not, it won't work.

```
int age = 25; if (age >= 18) { ??Console.WriteLine("You are an adult."); }
```

You can add else-if blocks if you want another block of code to run for a different set of conditions. An else block will run when none of the previous conditions evaluate to true.

```
int age = 25; if (age >= 18) { ??
Console.WriteLine("You are an adult."); } else if (age >= 13) { ??
Console.WriteLine("You are a teenager."); } else { ??
Console.WriteLine("You are a child."); }
```

In the above example, the first if statement is true and runs the first block of code. If the value of the age variable is 15, this program will run the second block of code instead. If the age variable is 8, the program will run the code inside the else block.

One-line If statement (without brackets)

If the block of code you want to run consists of only one line, you can remove the curly braces after the if condition:

```
if (age >= 18) ??Console.WriteLine("You are an adult.");
```

You can also apply this statement to else-if and else blocks:

```
if (age >= 18) ??  
Console.WriteLine("You are an adult."); else if (age >= 13) ??  
Console.WriteLine("You are a teenager."); else ??  
Console.WriteLine("You are a child.");
```

Command Switch

A switch statement in C# can be a more convenient way to arrange conditions if you want to avoid using too many if statements.

In a switch, you can enter a value to compare multiple cases, including possible selections with matching values.

```
int score = 4; char grade; switch (score) { ??case 5: ???grade = 'A'; ???  
break; ?case 4: ???grade = 'B'; ???break; ?case 3: ???grade = 'C'; ???  
break; ?case 2: ???grade = 'D'; ???break; ?case 1: ???grade = 'E'; ???  
break; ?default: ???grade = 'F'; ???  
break; } Console.WriteLine("Your grade is: " + grade);
```

Ternary operator

A ternary operator is shorthand for the if statement in C#. It follows the following syntax:

```
condition ? code when true : code when false
```

This command contains a condition, followed by a question mark. To the left of the colon is the program code that will run when the condition is true - true. To the right of the colon is the program code that will run when the condition is false - false.

```
int result = 49; var message = result > 50 ? "You passed!" : "You failed!"; Cons
```

nullable ternary operator

You can use the nullable ternary operator to assign a value to a variable that can be null. It uses the following spell:

```
var result = value ?? defaultValue
```

The example below assigns the value of the variable **num** to the variable **number** . If the **num** variable is **null** , it will assign the default value of 0 to the variable.

```
int? num = null; int number = num ?? 0;
```

Using conditional statements in C#

You can use conditional commands to create apps that can respond to specific conditions. When building an app, it's also important to consider other structures. That can make your code run more efficiently.

You finished reading the article "**How to write conditional statements in C#**" edited by the [TipsMake](#) team. We hope this article has provided you with many useful tech tips and tricks. You can search for similar articles on tips and guides. Thank you for reading and for following us regularly.

