< Previous Unit 3 of 7 ⌄ Next >

✓ 100 XP

Exercise - Remove code blocks from if statements

6 minutes

Software developers love it when they can write code that saves keystrokes and visual space without sacrificing readability. Only apply the phrase "less is more" in the development process when it makes your code more readable and understandable.

If a code block needs only one line of code, chances are you don't need to define a formal code block using curly braces. Although technically you don't even need to separate your code into multiple lines, combining statements on a single line can make your code hard to read.

Removing the curly braces as described above is a stylistic change that shouldn't affect the functionality of your code. However, you should take steps to ensure that your changes do not negatively impact how readable the code is. You can evaluate the impact of removing the curly braces and white space, then revert back to the original code if you find that the changes made your code less readable.

Create an example of an if statement that uses a code block

1. Ensure that you have an empty Program.cs file open in Visual Studio Code.

   If necessary, open Visual Studio Code, and then complete the following steps to prepare a Program.cs file in the Editor:

   a. On the **File** menu, select **Open Folder**.

   b. Use the Open Folder dialog to navigate to, and then open, the **CsharpProjects** folder.

   c. In the Visual Studio Code EXPLORER panel, select **Program.cs**.

   d. On the Visual Studio Code **Selection** menu, select **Select All**, and then press the Delete key.

2. Type the following code into the Visual Studio Code Editor:

   C#

```csharp
bool flag = true;
if (flag)
{
    Console.WriteLine(flag);
}
```

3. Save your code file, and then use Visual Studio Code to run your code.

   Enter `dotnet run` from the Terminal command prompt to run your code.

4. Verify that you see the following output:

   Output

   ```
   True
   ```

   Your code represents a good starting point, but you have a code block that includes a single line of code. In this case, is defining a code block necessary?

Remove the curly braces

The code above executes a code block when `flag` is `true`. Since this code block contains a single code line, you have the option to remove the curly braces.

1. Update your code in the Visual Studio Code Editor as follows:

   C#

   ```csharp
   bool flag = true;
   if (flag)
       Console.WriteLine(flag);
   ```

   > **Note**
   >
   > Removing the curly braces does not change the fact that `Console.WriteLine(flag);` is the code block for the `if` statement.

2. Save your code file, and then use Visual Studio Code to run your code.

3. Notice that the output is the same as before.

You should still see the following output:

Output

```
True
```

However, there's two fewer lines of code. Most importantly, with the code line below the `if` indented, your code is still easily readable.

Examine the readability of single-line form if statements

In this step, you're going to look at a case where code readability may be negatively impacted.

Since both the `if` statement and the method call to `Console.WriteLine()` are short, you might be tempted to combine them on a single line. After all, C# syntax for the `if` statement allows you to combine statements in this way.

1. Update your code in the Visual Studio Code Editor as follows:

   C#

   ```
   bool flag = true;
   if (flag) Console.WriteLine(flag);
   ```

2. Save your code file, and then use Visual Studio Code to run your code.

3. Notice that the output is still the same.

4. Take a minute to consider code readability.

   Imagine the previous two code samples nested within a larger section of code. Combining code statements (like you did in step 4) could make it harder to read.

   When implementing an `if` statement that includes a single-statement code block, Microsoft recommends that you consider these conventions:

   - Never use single-line form (for example: if (flag) Console.WriteLine(flag);
   - Using braces is always accepted, and required if any block of an if/else if/.../else compound statement uses braces or if a single statement body spans multiple lines.
   - Braces may be omitted only if the body of every block associated with an if/else if/.../else compound statement is placed on a single line.

5. To examine the readability impact for larger if-elseif-else constructs, update your code as follows:

   c#

   ```
   string name = "steve";
   if (name == "bob") Console.WriteLine("Found Bob");
   else if (name == "steve") Console.WriteLine("Found Steve");
   else Console.WriteLine("Found Chuck");
   ```

6. Save your code file, and then use Visual Studio Code to run your code.

   When you run the code, it should produce the following output:

   Output

   ```
   Found Steve
   ```

   The code runs, but these lines of code are dense and hard to read. You may want to reformat the code to include a line break after the `if`, `else if`, and `else` statements.

7. Compare the code that you just ran with the following code:

   c#

   ```
   string name = "steve";

   if (name == "bob")
       Console.WriteLine("Found Bob");
   else if (name == "steve")
       Console.WriteLine("Found Steve");
   else
       Console.WriteLine("Found Chuck");
   ```

   Notice how much easier it is to read the second code sample.

Recap

Here are a few important things to remember about `if` statement code blocks and readability:

- If you realize you have only one line of code listed within the code blocks of an `if-elseif-else` statement, you can remove the curly braces of the code block and white space.

Microsoft recommends that curly braces be used consistently for all of the code blocks of an `if-elseif-else` statement (either present or removed consistently).

- Only remove the curly braces of a code block when it makes the code more readable. It's always acceptable to include curly braces.
- Only remove the line feed if it makes the code more readable. Microsoft suggests that your code will be more readable when each statement is placed on its own code line.

Next unit: Exercise - Complete a challenge activity using variable scope

Continue ›