

Forum:[Hack Forums](#)

Thread: [VB.net Ability for single line code](#)

Schematic - October 10, 2015, 11:35 pm

Hi, I would like to have multiple line code in just one line, Like I am able to do in other software that I am familiar with.

Example:

```
Code If AetherTextbox5.Text Is "" Then
    AetherTextbox5.Text = "0"
End If
```

I would like to turn it into

```
Code If AetherTextbox5.Text Is "" Then AetherTextbox5.Text = "0" End If
```

Is this possible?

o??law - October 10, 2015, 11:42 pm

It's possible but people will make fun of you.

TooBaked - October 10, 2015, 11:44 pm

(Oct 10, 2015 11:42 PM)o??law Wrote: [\[CITE\]](#) It's possible but people will make fun of you.

+1. I was going to comment that it's stupid and pointless. Visual Studio holds your hand while you code VB.net anyways. Lmao

6e756c6c - October 10, 2015, 11:52 pm

```
Code If AetherTextbox5.Text = "" Then AetherTextbox5.Text = "0"
```

OR:

```
Code AetherTextBox5.Text = If(AetherTextBox5.Text = "", AetherTextBox5.Text = "0",
AetherTextBox5.Text)
```

OR (please don't ever use this method, just demonstrating that it works):

```
Code If AetherTextBox5.Text = "" Then : AetherTextBox5.Text = "0" : End If
```

My personal preference though would be something like..

```
Code If String.IsNullOrEmpty(AetherTextBox5.Text) Then
    AetherTextBox5.Text = "0"
End If
```

It really depends on your particular scenario though, I assume you are doing this to multiple textboxes in which case you should probably create a helper function to reduce code redundancy.

Edit:

Don't use 'Is' to compare strings, it's only by magic .NET hackery that your usage of it actually works. In all other cases if you were to initiate a new reference type and use 'Is' it would return false even if they were initiated with the same value. Instead, you should be using '='. :)

Schematic - October 11, 2015, 12:43 am

What would a helper function look like?

```

Code If AetherTextbox5.Text = String.Empty Then
    AetherTextbox5.Text = "0"
End If

If AetherTextbox6.Text = String.Empty Then
    AetherTextbox6.Text = "0"
End If
If AetherTextbox7.Text = String.Empty Then
    AetherTextbox7.Text = "0"
End If
If AetherTextbox8.Text = String.Empty Then
    AetherTextbox8.Text = "0"
End If
If AetherTextbox9.Text = String.Empty Then
    AetherTextbox9.Text = "0"
End If
If AetherTextbox3.Text = String.Empty Then
    AetherTextbox3.Text = "0"
End If
If AetherTextbox11.Text = String.Empty Then
    AetherTextbox11.Text = "0"
End If
If AetherTextbox10.Text = String.Empty Then
    AetherTextbox10.Text = "0"
End If
If AetherTextbox12.Text = String.Empty Then
    AetherTextbox12.Text = "0"
End If
If AetherTextbox13.Text = String.Empty Then
    AetherTextbox13.Text = "0"
End If
If AetherTextbox14.Text = String.Empty Then
    AetherTextbox14.Text = "0"
End If

```

This is what my redundant script looks like lol

Odus - October 11, 2015, 12:51 am

Create a universal handler and cast the sender to type TextBox. Or if this is for all, just loop through your form's controls (but this won't account for child controls) of type TextBox, then run your check.

To reiterate what Aeon said, never use the colon in place of a new line.

SIM-2362662 - October 11, 2015, 4:47 am

If you want you'r code to look smaller you can use :

```

Code If AetherTextbox5.Text = String.Empty Then AetherTextbox5.Text = "0"
If AetherTextbox6.Text = String.Empty Then AetherTextbox6.Text = "0"
If AetherTextbox7.Text = String.Empty Then AetherTextbox7.Text = "0"
If AetherTextbox8.Text = String.Empty Then AetherTextbox8.Text = "0"
If AetherTextbox9.Text = String.Empty Then AetherTextbox9.Text = "0"
If AetherTextbox3.Text = String.Empty Then AetherTextbox3.Text = "0"
If AetherTextbox11.Text = String.Empty Then AetherTextbox11.Text = "0"
If AetherTextbox10.Text = String.Empty Then AetherTextbox10.Text = "0"

```

```
If AetherTextbox12.Text = String.Empty Then AetherTextbox12.Text = "0"  
If AetherTextbox13.Text = String.Empty Then AetherTextbox13.Text = "0"  
If AetherTextbox14.Text = String.Empty Then AetherTextbox14.Text = "0"
```

Odus - October 11, 2015, 6:05 am

I had a thought after reading what you were actually doing. Why not set the Text property of your TextBoxes to 0 or even better, use a NumericUpDown if the input is numeric.

nazist - October 11, 2015, 11:11 pm

(Oct 10, 2015 11:52 PM)Aeonhack Wrote: [\[CITE\]](#) Code If AetherTextbox5.Text = "" Then
AetherTextbox5.Text = "0"
OR:

Code AetherTextBox5.Text = If(AetherTextBox5.Text = "", AetherTextBox5.Text = "0",
AetherTextBox5.Text)

OR (please don't ever use this method, just demonstrating that it works):

Code If AetherTextBox5.Text = "" Then : AetherTextBox5.Text = "0" : End If
My personal preference though would be something like..

Code If String.IsNullOrEmpty(AetherTextBox5.Text) Then
AetherTextBox5.Text = "0"
End If

It really depends on your particular scenario though, I assume you are doing this to multiple textboxes in which case you should probably create a helper function to reduce code redundancy.

Edit:

Don't use 'Is' to compare strings, it's only by magic .NET hackery that your usage of it actually works. In all other cases if you were to initiate a new reference type and use 'Is' it would return false even if they were initiated with the same value. Instead, you should be using '='. :)but why we can't use this method

i am used this method some time !!!

Code some code Then====> : <== some code ====> : <== some code

6e756c6c - October 11, 2015, 11:52 pm

(Oct 11, 2015 11:11 PM)nazist Wrote: [\[CITE\]](#) but why we can't use this method

i am used this method some time !!!

Code some code Then====> : <== some code ====> : <== some code

Because it doesn't have any effect on your code, it will still be treated as if the lines of code were on separate lines and it makes code extremely difficult to read / maintain. I promise if you work on any kind of major project and have to go back later on to update it, dealing with all those condensed lines is going to be a nightmare.