convert double to int

Ask Question



What is the best way to convert a double to an int? Should a cast be used?

129





10

edited Mar 11 '16 at 22:48



Levent Divilioglu 6,729 3 34 86

asked Nov 15 '10 at 6:27



user496949

32.8k 124 278 397

- 7 define "best". depends if you want up/down rounding, etc. − RPM1984 Nov 15 '10 at 6:30 ✓
- 7 @RPM1984: Define etc. :) Armen Tsirunyan Nov 15 '10 at 6:37 ▶
- 0Armen touche`:) Doesn't really matter anyway Skeet is here, nothing else matters. – RPM1984 Nov 15 '10 at 6:42

10 Answers



202

You can use a cast if you want the default truncatetowards-zero behaviour. Alternatively, you might want to use Math.Ceiling, Math.Round, Math.Floor etc - although

Home

PUBLIC

Tags

Users

Jobs

Teams

Stack Overflow



Don't forget that the range of int is much smaller than the range of double. A cast from double to int won't throw an exception if the value is outside the range of int in an unchecked context, whereas a call to Convert.ToInt32(double) will. The result of the cast (in an unchecked context) is explicitly undefined if the value is outside the range.

answered Nov 15 '10 at 6:30



Jon Skeet 1107k 704 8051 8510

- 24 @user: The size of an int is always 32 bits, regardless of whether you're using a 32 or 64 bit machine. – Joren Nov 15 '10 at 10:15
- 2 And doubles can be much huger than even a 64 bit int. Adrian Ratnapala May 1 '13 at 8:07
- For C# noobs like me: both Math and Convert are part of System. So the complete answer looks like this: intVal = System.Convert.ToInt32(System.Math.Floor(dblVal)); − kris May 1 '16 at 8:07
- @user1290746: That would work, but you'd usually just have
 using System; at the top of the file, at which point it could
 just be intVal = Convert.ToInt32(Math.Floor(dblVal));
 Jon Skeet May 1 '16 at 8:08
- @user1290746: But in that case, I don't think the right solution is to advise people to use the fully-qualified names the better solution is to learn about using directives. – Jon Skeet May 1 '16 at 8:15





if you use cast, that is, (int)SomeDouble you will truncate the fractional part. That is, if SomeDouble were 4.9999 the result would be 4 not 5. Converting to int doesn't round



Armen Tsirunyan 94.4k 44 280 392



Yeah, why not?



double someDouble = 12323.2;
int someInt = (int)someDouble;



Using the Convert class works well too.

```
int someOtherInt = Convert.ToInt32(someDouble);
```

answered Nov 15 '10 at 6:29



Jeff Mercado

94.5k 20 191 217



Convert.ToInt32 is the best way to convert





answered Nov 15 '10 at 6:30



anishMarokey

10.1k 2 27 43

This a typically unsafe function because it accepts just anything. – Ant_222 Jul 5 '16 at 14:48



The best way is to simply use <code>Convert.ToInt32</code> . It is fast and also rounds correctly.





answered May 30 '13 at 1:59





Here is a complete example

2

}

```
class Example
{
  public static void Main()
  {
    double x, y;
    int i;

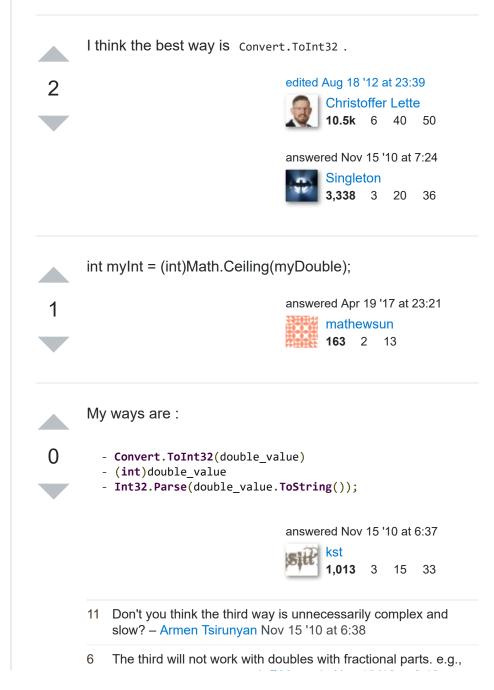
    x = 10.0;
    y = 3.0;

    // cast double to int, fractional component lost (Line i = (int) (x / y);
    Console.WriteLine("Integer outcome of x / y: " + i);
```

If you want to round the number to the closer integer do the following:

By using our site, you acknowledge that you have read and understand our Cookie Policy, Privacy Policy, and our Terms of Service.

maniana saffan





label8.Text = "" + years.ToString("00") + " years";





when you want to send it to a label, or something, and you don't want any fractional component, this is the best way

if you want with only 2, and it's always like that

answered Jul 14 '14 at 18:59

