How to Round to the nearest whole number in C#

Ask Question

▲

76

▼

★

8

How can I round values like this:

```
1.1 => 1
1.5 => 2
1.9 => 2
```

`Math.Ceiling()` is not helping me. Any ideas?

c# rounding

edited Mar 8 '14 at 3:26
Slipp D. Thompson
22.5k 3 33 37

asked Jan 13 '12 at 1:11
SOF User
2,974 20 61 110

2 Use `Math.Round` – ean5533 Jan 13 '12 at 1:13

Math.Round() can do the trick. – Only Bolivian Here Jan 13 '12 at 1:13

Something like `Math.Round` ? – M.Babcock Jan 13 '12 at 1:13

possible duplicate of [How to round up value C# to the nearest integer?](https://stackoverflow.com) –

Alex Angas Jan 20 '14 at 1:03

15 Answers

173

✓

See the official documentation for more. For example:

Basically you give the `Math.Round` method three parameters.

1. The value you want to round.
2. The number of decimals you want to keep after the value.
3. An optional parameter you can invoke to use AwayFromZero rounding. (*ignored unless rounding is ambiguous, e.g. 1.5*)

Sample code:

```
var roundedA = Math.Round(1.1, 0); // Output: 1
var roundedB = Math.Round(1.5, 0, MidpointRounding.AwayFromZero); //
var roundedC = Math.Round(1.9, 0); // Output: 2
var roundedD = Math.Round(2.5, 0); // Output: 2
var roundedE = Math.Round(2.5, 0, MidpointRounding.AwayFromZero); //
var roundedF = Math.Round(3.49, 0, MidpointRounding.AwayFromZero); /
```

Live Demo

You need `MidpointRounding.AwayFromZero` if you want a .5 value to be rounded up. Unfortunately this isn't the default behavior for `Math.Round()` . If using `MidpointRounding.ToEven` (the default) the value is rounded to the nearest **even** number (`1.5` is rounded to `2` , but `2.5` is also rounded to `2`).

edited Jul 11 '17 at 9:14
monkey0506
1,382 14 22

answered Jan 13 '12 at 1:14

Only Bolivian Here
16.5k 54 141 239

14 on the other hand, using `away from zero` also means that `-1.5` will round to `-2` . – davogotland Jan 13 '12 at 1:35

2 use Math.Ceiling, its not a good practice to use Math.Round for frictions, read: stackoverflow.com/questions/9221205/…, – Yakir Manor Sep 16 '13 at 11:27

3 I am finding that Math.Round(1.5, 0) returns 2 – David Sykes Jan 20 '14 at 13:45

@davogotland is their anyway to round 137.5 to 140 not to 138 ? I mean rounding to nearest tenth ? – sam Feb 27 '17 at 11:50

@sam perhaps divide by 10, then round with Math.Ceiling, and finally multiple by 10? – davogotland Mar 1 '17 at 9:54

57

```
Math.Ceiling
```

always rounds up (towards the ceiling)

```
Math.Floor
```

always rounds down (towards to floor)

what you are after is simply

```
Math.Round
```

which rounds as per this post

edited May 23 '17 at 12:18
Community ♦
1 1

answered Jan 13 '12 at 1:15
devrooms
2,569 13 24

is their anyway to round 137.5 to 140 not to 138 ? I mean rounding to nearest tenth ? – sam Feb 27 '17 at 11:53

6

You need `Math.Round` , not `Math.Ceiling` . `Ceiling` always "rounds" up, while `Round` rounds up or down depending on the value after the decimal point.

edited Jul 31 '13 at 14:33
Dozer789
1,385 1 17 40

answered Jan 13 '12 at 1:14
dasblinkenlight
617k 61 792 1210

6

there's this manual, and kinda cute way too:

```
double d1 = 1.1;
double d2 = 1.5;
double d3 = 1.9;

int i1 = (int)(d1 + 0.5);
int i2 = (int)(d2 + 0.5);
int i3 = (int)(d3 + 0.5);
```

simply add 0.5 to any number, and cast it to int (or floor it) and it will be mathematically correctly rounded :D

edited Nov 20 '12 at 18:10

answered Jan 13 '12 at 1:28

davogotland

2,414 1 11 17

Does it also work for 1.0? :/ – ver Nov 17 '12 at 20:02

1 yes. 1.0 + 0.5 = 1.5. 1.5 rounded down is 1 :) – davogotland Nov 18 '12 at 4:30

It still looks suspicious. Firstly, the question asks about rounding *up* and secondly, when I tried it just now, the default implementation of Math.Round(1.5) rounds to 2. So this may not be what he wanted. – ver Nov 20 '12 at 7:46

also, your example mixes decimal point with decimal comma. Which one do you normally use (in Sweden, I guess)? :) – ver Nov 20 '12 at 7:48

oops... oh yeah, sorry. in programming the decimal point of course, but in formal text we use the decimal comma. and yes, sweden ^^ about the question, and the "rounding up" part: i think that's just some language mistake. in the examples given by op, some decimal numbers round down. – davogotland Nov 20 '12 at 18:09

Just a reminder. Beware for double.

5

```
Math.Round(0.3 / 0.2 ) result in 1, because in double 0.3 / 0.2 = 1.
Math.Round( 1.5 ) = 2
```

answered May 26 '15 at 14:05

Kosmas

176 2 6

4

You can use Math.Round as others have suggested (recommended), or you could add 0.5 and cast to an int (which will drop the decimal part).

```
double value = 1.1;
int roundedValue = (int)(value + 0.5); // equals 1

double value2 = 1.5;
int roundedValue2 = (int)(value2 + 0.5); // equals 2
```

answered Jan 13 '12 at 1:16

Marlon

14.6k 9 53 88

3

You have the Math.Round function that does exactly what you want.

```
Math.Round(1.1) results with 1
Math.Round(1.8) will result with 2.... and so one.
```

answered Jan 13 '12 at 1:14

user496607

152 1 7 20

1 What about `1.5` as the value? You need more parameters. – Only Bolivian Here Jan 13 '12 at 1:27

1

```
var roundedVal = Math.Round(2.5, 0);
```

It will give result:

```
var roundedVal = 3
```

edited Apr 28 '17 at 15:58
Liam
16.3k 16 76 130

answered Apr 28 '17 at 13:38
Ankita_systematix
315 3 4

1

I was looking for this, but my example was to take a number, such as 4.2769 and drop it in a span as just 4.3. Not exactly the same, but if this helps:

```
Model.Statistics.AverageReview   <= it's just a double from the mode
```

Then:

```
@Model.Statistics.AverageReview.ToString("n1")   <=gives me 4.3
@Model.Statistics.AverageReview.ToString("n2")   <=gives me 4.28
```

etc...

edited Dec 26 '13 at 21:00

mgibsonbr
18.9k 7 52 88

answered Dec 26 '13 at 20:31

user2970629
11 1

1

this will round up to the nearest 5 or not change if it already is divisible by 5

```
public static double R(double x)
{
    // markup to nearest 5
    return (((int)(x / 5)) * 5) + ((x % 5) > 0 ? 5 : 0);
}
```

edited Apr 22 '13 at 7:39

Baby Groot
3,989 11 45 66

answered Apr 22 '13 at 7:22

rick
11 1

1

Use `Math.Round` :

```
double roundedValue = Math.Round(value, 0)
```

answered Jan 13 '12 at 1:12

Thomas Levesque
237k 53 499 670

0

Using `Math.Round(number)` rounds to the nearest whole number.

answered Nov 15 '17 at 10:33

John Kennedy
2,718 2 11 27

0

If your working with integers rather than floating point numbers, here is the way.

```
#define ROUNDED_FRACTION(numr,denr) ((numr/denr)+(((numr%denr)<(denr,
```

Here both **"numr"** and **"denr"** are unsigned integers.

edited Dec 9 '15 at 10:40

Kumar Saurabh
2,120 5 23 37

answered Dec 9 '15 at 10:26

Sarath Kumar
1 1

0

Write your own round method. Something like,

```
function round(x)
    rx = Math.ceil(x)
    if (rx - x <= .000001)
```

```
        return int(rx)
    else
        return int(x)
end
```

answered Nov 27 '17 at 11:22

Sandeep Anand

31 12

-1

```
decimal RoundTotal = Total - (int)Total;
if ((double)RoundTotal <= .50)
   Total = (int)Total;
else
   Total = (int)Total + 1;
lblTotal.Text = Total.ToString();
```

edited Oct 31 '17 at 12:20

Roman Marusyk

12.3k 12 37 69

answered Oct 31 '17 at 11:51

Salma

1