

متد [System.Math.Round](#) که برای گرد کردن اعداد اعشاری به کار می‌رود، دارای 8 نوع overload می‌باشد، که عدم توجه به موارد مربوط به آن باعث بروز خطا در محاسبات خواهد شد. به طور مثال پیش بینی شما از گرد کردن عدد 3.45 عدد 3.5 است ولی گاهی 3.5 و گاهی 3.4 گرد خواهد شد. پس بهتر است تا با نکات زیر به شکل دقیق آشنا باشید.

**Round(Decimal)**  
Rounds a decimal value to the nearest integral value.

(ورودی: دسیمال) به نزدیکترین عدد کامل گرد میکند، 4.3 به 4 و 4.8 به 5 گرد می‌شود. ولی در صورتیکه فاصله تا عدد کامل قبل و بعد برابر باشد به نزدیکترین عدد زوج گرد میکند، 4.5 به 4 گرد می‌شود چرا که 5 عددی فرد است.

**نکته 1:** خروجی تابع از نوع دسیمال است نه عدد کامل.

**نکته 2:** این تابع بر طبق استاندارد IEEE Standard 754, section 4 پیاده سازی شده است که در اصطلاح rounding to nearest یا banker's rounding نیز گفته می‌شود. نتیجه برای به حداقل رساندن خطا است. نتیجه این حالت از متد با نتیجه Round(Decimal, MidpointRounding.ToEven) برابر است.

**Round(Double)**  
Rounds a double-precision floating-point value to the nearest integral value.

(ورودی: double) به نزدیکترین عدد کامل گرد میکند، 4.3 به 4 و 4.8 به 5 گرد می‌شود. ولی در صورتیکه فاصله تا عدد کامل قبل و بعد برابر باشد به نزدیکترین عدد زوج گرد میکند، 4.5 به 4 گرد می‌شود چرا که 5 عددی فرد است.

**نکته 1:** خروجی تابع از نوع double است نه عدد کامل.

**نکته 2:** این تابع بر طبق استاندارد IEEE Standard 754, section 4 پیاده سازی شده است که در اصطلاح rounding to nearest یا banker's rounding نیز گفته می‌شود. نتیجه برای به حداقل رساندن خطا است. نتیجه این حالت از متد با نتیجه Round(Double, MidpointRounding.ToEven) برابر است.

**نکته 3:** گاهی اوقات به دلیل از دست دادن دقت، ناشی از استفاده از مقادیر دسیمال به جای ممیز شناور و یا انجام محاسبات ریاضی بر روی بخش ممیزی خواهد بود. مثلاً: زمانی که 11.5 ماحصل جمع 1 و 11.4 باشد به جای 12 که عدد زوج است به 11 گرد می‌شود!

**Round(Decimal, Int32)**  
Rounds a decimal value to a specified number of fractional digits.

عدد دسیمال ورودی خود را به صورتی گرد میکند که:

1: تعداد ارقام اعشاری بعد از ممیز به اندازه پارامتر دوم این نوع ورودی متد Round باشد (بین صفر تا 28).

2: استفاده از این متد همانند فراخوانی آن با ورودی‌های Round(Decimal, Int32, MidpointRounding.ToEven) است. یعنی اینکه اگر رقم آخر بعد از ممیز دقیقاً وسط مقدار قبل و بعد باشد (3.75) در صورتی که رقم ماقبل آخر فرد باشد رو به بالا گرد خواهد شد (مثال: 3.75 به 3.8 گرد خواهد شد) و اگر رقم ماقبل آخر زوج باشد تغییر نخواهد کرد (مثال: 3.45 به 3.4 گرد خواهد شد)

**نکته 1:** این تابع بر طبق استاندارد IEEE Standard 754, section 4 پیاده سازی شده است که در اصطلاح rounding to nearest یا banker's rounding نیز گفته می‌شود.

```
Math.Round(3.44, 1); //Returns 3.4.  
Math.Round(3.45, 1); //Returns 3.4.  
Math.Round(3.46, 1); //Returns 3.5.
```

```
Math.Round(4.34, 1); // Returns 4.3
Math.Round(4.35, 1); // Returns 4.4
Math.Round(4.36, 1); // Returns 4.4
```

```
Round(Decimal, MidpointRounding)
Rounds a decimal value to the nearest integer.
A parameter specifies how to round the value if it is midway between two other numbers.
```

عدد دسیمال ورودی خود را به نزدیک‌ترین عدد integer گرد میکند، پارامتر اول عدد گرد نشده و پارامتر دوم مشخص میکند که در صورتیکه مقدار اعشاری عددی میانی (5 و 50 و 500 و 5000 و .... باشد) چگونه این گرد کردن صورت گیرد.

#### حالت‌های MidpointRounding:

MidpointRounding.ToEven: در صورتیکه مقدار اعشاری عددی میانی (5 و 50 و 500 و 5000 و ....) باشد، و رقم ماقبل آخر اعشار فرد باشد، رو به بالا گرد خواهد شد و در صورتیکه رقم ماقبل آخر اعشار زوج باشد، بدون تغییر باقی خواهد ماند (3.75 به 3.8 و 3.65 به 3.6 گرد می‌شود).

**نکته 1:** این تابع بر طبق استاندارد IEEE Standard 754, section 4 پیاده سازی شده است که در اصطلاح rounding to nearest یا banker's rounding نیز گفته می‌شود.

MidpointRounding.AwayFromZero: در صورتیکه مقدار اعشاری عددی میانی (5 و 50 و 500 و 5000 و ....) باشد، در این حالت همواره عمل گرد کردن رو به رقم بعدی خواهد بود. این رایج‌ترین حالت گرد کردن است که به symmetric arithmetic rounding شناخته می‌شود.

```
Round(Double, Int32)
Rounds a double-precision floating-point value to a specified number of fractional digits.
```

یک عدد اعشاری از نوع Double (با دقت مضاعف) که تعداد مشخصی رقم بعد از ممیز دارد (به طور مثال 10 رقم اعشار)، به تعداد رقم اعشاری که کاربر به عنوان پارامتر دوم ذکر میکند (بین صفر تا 15 رقم)، (مثلا 4 رقم) گرد میکند. اگر تعداد رقم اعشار بیش از 15 تعیین شود، عدد 15 جایگزین خواهد شد. استفاده از این متد همانند فراخوانی آن به صورت Round(Double, Int32, MidpointRounding.ToEven) می‌باشد. یعنی اینکه اگر رقم آخر بعد از ممیز دقیقاً وسط مقدار قبل و بعد باشد (3.75) در صورتی که رقم ماقبل آخر فرد باشد رو به بالا گرد خواهد شد (مثال: 3.75 به 3.8 گرد خواهد شد) و اگر رقم ماقبل آخر زوج باشد تغییر نخواهد کرد (مثال: 3.45 به 3.4 گرد خواهد شد).

**نکته 1:** این تابع بر طبق استاندارد IEEE Standard 754, section 4 پیاده سازی شده است که در اصطلاح rounding to nearest یا banker's rounding نیز گفته می‌شود.

```
Round(Double, MidpointRounding)
Rounds a double-precision floating-point value to the nearest integer.
A parameter specifies how to round the value if it is midway between two other numbers.
```

(عدد اعشاری ورودی: Double) عدد با دقت مضاعف ورودی خود را به نزدیک‌ترین عدد integer گرد میکند، پارامتر اول عدد گرد نشده و پارامتر دوم مشخص میکند در صورتیکه مقدار اعشاری عددی میانی (5 و 50 و 500 و 5000 و .... باشد) چگونه این گرد کردن صورت گیرد.

#### حالت‌های MidpointRounding:

MidpointRounding.ToEven: در صورتیکه مقدار اعشاری عددی میانی (5 و 50 و 500 و 5000 و ....) باشد، و رقم ماقبل آخر اعشار فرد باشد، رو به بالا گرد خواهد شد و در صورتیکه رقم ماقبل آخر اعشار زوج باشد، بدون تغییر باقی خواهد ماند (3.75 به 3.8 و 3.65 به 3.6 گرد می‌شود).

**نکته 1:** این تابع بر طبق استاندارد IEEE Standard 754, section 4 پیاده سازی شده است که در اصطلاح rounding to nearest یا banker's rounding نیز گفته می‌شود.

MidpointRounding.AwayFromZero: در صورتیکه مقدار اعشاری عددی میانی (5 و 50 و 500 و 5000 و ....) باشد، در این حالت همواره عمل گرد کردن رو به رقم بعدی خواهد بود. این رایج‌ترین حالت گرد کردن است که به symmetric arithmetic rounding شناخته می‌شود.

```
Round(Decimal, Int32, MidpointRounding)
Rounds a decimal value to a specified number of fractional digits.
A parameter specifies how to round the value if it is midway between two other numbers.
```

عدد دسیمال ورودی خود را با تعداد اعشار اعلام شده و به صورتی گرد میکند که:

1: تعداد ارقام اعشاری بعد از ممیز به اندازه پارامتر دوم این نوع ورودی متد Round باشد (بین صفر تا 28).

#### حالت‌های MidpointRounding:

MidpointRounding.ToEven: در صورتیکه مقدار اعشاری عددی میانی (5 و 50 و 500 و 5000 و ....) باشد، و رقم ماقبل آخر اعشار فرد باشد، رو به بالا گرد خواهد شد و در صورتیکه رقم ماقبل آخر اعشار زوج باشد، بدون تغییر باقی خواهد ماند (3.75 به 3.8 و 3.65 به 3.6 گرد می‌شود).

**نکته 1:** این تابع بر طبق استاندارد IEEE Standard 754, section 4 پیاده سازی شده است که در اصطلاح rounding to nearest یا banker's rounding نیز گفته می‌شود.

MidpointRounding.AwayFromZero: در صورتیکه مقدار اعشاری عددی میانی (5 و 50 و 500 و 5000 و ....) باشد، در این حالت همواره عمل گرد کردن به سمت رقم بعدی خواهد بود. این رایج‌ترین حالت گرد کردن است که به symmetric arithmetic rounding شناخته می‌شود.

```
3.4 = Math.Round( 3.45, 1, MidpointRounding.ToEven)
3.5 = Math.Round( 3.45, 1, MidpointRounding.AwayFromZero)

-3.4 = Math.Round(-3.45, 1, MidpointRounding.ToEven)
-3.5 = Math.Round(-3.45, 1, MidpointRounding.AwayFromZero)
```

```
Round(Double, Int32, MidpointRounding)
Rounds a double-precision floating-point value to the specified number of fractional digits.
A parameter specifies how to round the value if it is midway between two other numbers.
```

(عدد اعشاری ورودی: Double) عدد با دقت مضاعف ورودی خود را به نزدیک‌ترین عدد با تعداد رقم اعشار مشخص شده گرد میکند، پارامتر اول متد، عدد گرد نشده و پارامتر دوم تعداد رقم اعشار (بین صفر تا 15 رقم) تعیین شده گرد می‌کند. اگر تعداد رقم اعشار بیش از 15 تعیین شود، عدد 15 جایگزین خواهد شد. و پارامتر سوم مشخص میکند که در صورتیکه مقدار اعشاری عددی میانی (5 و 50 و 500 و 5000 و .... باشد) چگونه این گرد کردن صورت گیرد.

#### حالت‌های MidpointRounding:

MidpointRounding.ToEven: در صورتیکه مقدار اعشاری عددی میانی (5 و 50 و 500 و 5000 و ....) باشد، و رقم ماقبل آخر اعشار فرد باشد، رو به بالا گرد خواهد شد و در صورتیکه رقم ماقبل آخر اعشار زوج باشد، بدون تغییر باقی خواهد ماند (3.75 به 3.8 و 3.65 به 3.6 گرد می‌شود).

**نکته 1:** این تابع بر طبق استاندارد IEEE Standard 754, section 4 پیاده سازی شده است که در اصطلاح rounding to nearest یا banker's rounding نیز گفته می‌شود.

MidpointRounding.AwayFromZero: در صورتیکه مقدار اعشاری عددی میانی (5 و 50 و 500 و 5000 و ....) باشد، در این حالت همواره عمل گرد کردن به سمت رقم بعدی خواهد بود. این رایج‌ترین حالت گرد کردن است که به symmetric arithmetic rounding شناخته می‌شود.

```
// The example displays the following output:  
// 2.125 --> 2.13  
// 2.135 --> 2.13  
// 2.145 --> 2.15  
// 3.125 --> 3.13  
// 3.135 --> 3.14  
// 3.145 --> 3.15  
This code example produces the following results:  
  
3.4 = Math.Round( 3.45, 1)  
-3.4 = Math.Round(-3.45, 1)  
  
3.4 = Math.Round( 3.45, 1, MidpointRounding.ToEven)  
3.5 = Math.Round( 3.45, 1, MidpointRounding.AwayFromZero)  
  
-3.4 = Math.Round(-3.45, 1, MidpointRounding.ToEven)  
-3.5 = Math.Round(-3.45, 1, MidpointRounding.AwayFromZero)
```

## نظرات خوانندگان

نویسنده:

سعید

تاریخ:

۱۳:۳۸ ۱۳۹۱/۰۶/۰۹

جالب بود. هیچ وقت به این مساله تا این حد دقیق نشده بودم!

نویسنده:

مسعود زبانی

تاریخ:

۱۷:۳۳ ۱۳۹۱/۰۹/۱۹

با سلام1

من اعداد دسیمال رو به صورت زیر رند میکنم:

```
decimal.Round(x,2)
```

حالا اگه عدد مثلا 12 باشه به صورت زیر نشون میده:

12.00

در صورتیکه نمیخوام دوتا صفر رو نشون بده... باید چیکار کنم؟

نویسنده:

وحید نصیری

تاریخ:

۱۸:۴۲ ۱۳۹۱/۰۹/۱۹

```
dec.ToString("G29")
```

این مطلب را مطالعه کنید: « [String Formatting in CS](#) » و [همچنین](#)

نویسنده:

مسعود زبانی

تاریخ:

۹:۱ ۱۳۹۱/۰۹/۲۰

ممنون آقای نصیری