C# Math.Round() – Syntax & Examples

C# Math.Round() – Examples

In this tutorial, we will learn about the C# Math.Round() method, and learn how to use this method to round a given number to the nearest integral value, with the help of examples.

Round(Decimal)

Math.Round(d) rounds a decimal value `d` to the nearest integral value, and rounds midpoint values to the nearest even number.

Syntax

The syntax of Round(d) method is

```
Math.Round(Decimal d)
```

where

Parameter	Description
d	The decimal number to be rounded.

Return Value

The method returns rounded Decimal value.

Example 1 – Round(d)

In this example, we will take some decimal values and round them to the nearest integral values using Math.Round() method.

C# Program

```
using System;

class Example {
    static void Main(string[] args) {
        Decimal d, result;

        d = 10.2563M;
        result = Math.Round(d);
        Console.WriteLine($"Round({d}) = {result}");

        d = 10.63M;
        result = Math.Round(d);
        Console.WriteLine($"Round({d}) = {result}");

        d = 10.5M;
        result = Math.Round(d);
        Console.WriteLine($"Round({d}) = {result}");
    }
}
```

Output

```
Round(10.2563) = 10
Round(10.63) = 11
Round(10.5) = 10
```

Round(Decimal, Int32)

Math.Round(d, decimals) rounds a decimal value `d` to a specified number of fractional digits `decimals`, and rounds midpoint values to the nearest even number.

Syntax

The syntax of Round(d, decimals) method is

```
Math.Round(Decimal d, Int32 decimals)
```

where

Parameter	Description
d	The decimal number to be rounded.
decimals	The number of decimal places in the return value.

Return Value

The method returns value.

Example 2 – Round(d, decimals)

In this example, we will take some decimal values and round them to specific number of decimal points using Math.Round() method.

C# Program

```
using System;

class Example {
    static void Main(string[] args) {
        Decimal d, result;
        Int32 decimals;

        d = 10.2563M;
        decimals = 2;
        result = Math.Round(d, decimals);
        Console.WriteLine($"Round({d}, {decimals}) = {result}");

        d = 10.63524M;
        decimals = 1;
        result = Math.Round(d, decimals);
        Console.WriteLine($"Round({d}, {decimals}) = {result}");

        d = 10.5M;
        decimals = 0;
        result = Math.Round(d, decimals);
        Console.WriteLine($"Round({d}, {decimals}) = {result}");
    }
}
```

Output

```
Round(10.2563, 2) = 10.26
Round(10.63524, 1) = 10.6
Round(10.5, 0) = 10
```

Round(Decimal, Int32, MidpointRounding)

Math.Round(d, decimals, mode) rounds a decimal value `d` to a specified number of fractional digits `decimals`, and uses the specified rounding convention `mode` for midpoint values.

Syntax

The syntax of Round(d, decimals, MidpointRounding) method is

```
Math.Round(Decimal d, Int32 decimals, MidpointRounding mode)
```

```
Math.Round(Decimal d, Int32 decimals, MidpointRounding mode)
```

where

Parameter	Description
d	The decimal number to be rounded.
decimals	The number of decimal places in the return value.
mode	Specification for how to round `d` if it is midway between two other numbers.

Return Value

The method returns rounded Decimal value.

Example 3 – Round(d, decimals, mode)

In this example, we will take some decimal values and round them to specific number of decimal points with different modes using Math.Round() method.

C# Program

```
using System;

class Example {
    static void Main(string[] args) {
        Decimal d, result;
        Int32 decimals;

        d = 10.2M;
        decimals = 0;
        result = Math.Round(d, decimals, MidpointRounding.AwayFromZero);
        Console.WriteLine($"Round({d}, {decimals}, MidpointRounding.AwayFromZero)

        d = 10.8M;
        decimals = 0;
        result = Math.Round(d, decimals, MidpointRounding.ToEven);
        Console.WriteLine($"Round({d}, {decimals}, MidpointRounding.ToEven)

        d = 10.8M;
        decimals = 0;
        result = Math.Round(d, decimals, MidpointRounding.ToNegativeInfinity);
        Console.WriteLine($"Round({d}, {decimals}, MidpointRounding.ToNegativeInfinity)

        d = 10.2M;
        decimals = 0;
        result = Math.Round(d, decimals, MidpointRounding.ToPositiveInfinity);
        Console.WriteLine($"Round({d}, {decimals}, MidpointRounding.ToPositiveInfinity)

        d = 10.8M;
        decimals = 0;
        result = Math.Round(d, decimals, MidpointRounding.ToZero);
```

```
        Console.WriteLine($"Round({d}, {decimals}, MidpointRounding.ToZero)
    }
}
```

Output

```
Round(10.2, 0, MidpointRounding.AwayFromZero)       = 10
Round(10.8, 0, MidpointRounding.ToEven)             = 11
Round(10.8, 0, MidpointRounding.ToNegativeInfinity) = 10
Round(10.2, 0, MidpointRounding.ToPositiveInfinity) = 11
Round(10.8, 0, MidpointRounding.ToZero)             = 10
```

Round(Decimal, MidpointRounding)

Math.Round(d, mode) rounds a decimal value `d` to the nearest integer, and uses the specified rounding convention `mode` for midpoint values.

Syntax

The syntax of Round(d, mode) method is

```
Math.Round(Decimal d, MidpointRounding mode)
```

where

Parameter	Description
d	The decimal number to be rounded.
mode	Specification for how to round the value `d` if it is midway between two other numbers.

Return Value

The method returns rounded Decimal value.

Example 4 – Round(Decimal, MidpointRounding)

In this example, we will take some decimal values and round them with different modes using Math.Round() method.

C# Program

```
using System;

class Example {
    static void Main(string[] args) {
        Decimal d, result;

        d = 10.2M;
        result = Math.Round(d, MidpointRounding.AwayFromZero);
        Console.WriteLine($"Round({d}, MidpointRounding.AwayFromZero)       = {result}")

        d = 10.8M;
        result = Math.Round(d, MidpointRounding.ToEven);
        Console.WriteLine($"Round({d}, MidpointRounding.ToEven)             = {result}")

        d = 10.8M;
        result = Math.Round(d, MidpointRounding.ToNegativeInfinity);
        Console.WriteLine($"Round({d}, MidpointRounding.ToNegativeInfinity) = {result}")

        d = 10.2M;
        result = Math.Round(d, MidpointRounding.ToPositiveInfinity);
        Console.WriteLine($"Round({d}, MidpointRounding.ToPositiveInfinity) = {result}")

        d = 10.8M;
        result = Math.Round(d, MidpointRounding.ToZero);
        Console.WriteLine($"Round({d}, MidpointRounding.ToZero)             = {result}")
    }
}
```

Output

```
Round(10.2, MidpointRounding.AwayFromZero)       = 10
Round(10.8, MidpointRounding.ToEven)             = 11
Round(10.8, MidpointRounding.ToNegativeInfinity) = 10
Round(10.2, MidpointRounding.ToPositiveInfinity) = 11
Round(10.8, MidpointRounding.ToZero)             = 10
```

Round(Double)

Math.Round(d) rounds a double-precision floating-point value `d` to the nearest integral value, and rounds midpoint values to the nearest even number.

Syntax

The syntax of Round(d) method is

```
Math.Round(Double d)
```

where

Parameter Parameter	Description Description
d	The double-precision floating-point number to be rounded.

Return Value

The method returns rounded Double value.

Example 5 – Round(d)

In this example, we will take some double-precision floating-point numbers and round them to the nearest integral values using Math.Round() method.

C# Program

```
using System;

class Example {
    static void Main(string[] args) {
        Double d, result;

        d = 10.2563;
        result = Math.Round(d);
        Console.WriteLine($"Round({d}) = {result}");

        d = 10.63;
        result = Math.Round(d);
        Console.WriteLine($"Round({d}) = {result}");

        d = 10.5;
        result = Math.Round(d);
        Console.WriteLine($"Round({d}) = {result}");
    }
}
```

Output

```
Round(10.2563) = 10
Round(10.63) = 11
Round(10.5) = 10
```

Round(Double, Int32)

Math.Round(d, decimals) rounds a double-precision floating-point value `d` to a specified number of fractional digits `decimals`, and rounds midpoint values to the nearest even number.

Syntax

The syntax of Round(d, decimals) method is

```
Math.Round(Double d, Int32 decimals)
```

where

Parameter	Description
d	The double-precision floating-point number to be rounded.
decimals	The number of decimal places in the return value.

Return Value

The method returns rounded Double value.

Example 6 – Round(Double, Int32)

In this example, we will take some decimal double-precision floating-point numbers and round them to specific number of decimal points using Math.Round() method.

C# Program

```
using System;

class Example {
    static void Main(string[] args) {
        Double d, result;
        Int32 decimals;

        d = 10.2563;
        decimals = 2;
        result = Math.Round(d, decimals);
        Console.WriteLine($"Round({d}, {decimals}) = {result}");

        d = 10.63524;
        decimals = 1;
        result = Math.Round(d, decimals);
        Console.WriteLine($"Round({d}, {decimals}) = {result}");

        d = 10.5;
        decimals = 0;
        result = Math.Round(d, decimals);
        Console.WriteLine($"Round({d}, {decimals}) = {result}");
    }
}
```

Output

```
Round(10.2563, 2) = 10.26
Round(10.63524, 1) = 10.6
Round(10.5, 0) = 10
```

Round(Double, Int32, MidpointRounding)

Math.Round(d, decimals, mode) rounds a double-precision floating-point value `d` to a specified number of fractional digits `decimals`, and uses the specified rounding convention `mode` for midpoint values.

Syntax

The syntax of Round(d, decimals, mode) method is

```
Math.Round(Double d, Int32 decimals, MidpointRounding mode)
```

where

Parameter	Description
d	The double-precision floating-point number to be rounded.
decimals	The number of decimal places in the return value.
mode	Specification for how to round `d` if it is midway between two other numbers.

Return Value

The method returns rounded Double value.

Example 7 – Round(Double, Int32, MidpointRounding)

In this example, we will take some decimal double-precision floating-point numbers and round them to specific number of decimal points with different modes using Math.Round() method.

C# Program

```
using System;

class Example {
    static void Main(string[] args) {
        Double d, result;
        Int32 decimals;
```

```
        d = 10.2;
        decimals = 0;
        result = Math.Round(d, decimals, MidpointRounding.AwayFromZero);
        Console.WriteLine($"Round({d}, {decimals}, MidpointRounding.AwayFromZero)

        d = 10.8;
        decimals = 0;
        result = Math.Round(d, decimals, MidpointRounding.ToEven);
        Console.WriteLine($"Round({d}, {decimals}, MidpointRounding.ToEven)

        d = 10.8;
        decimals = 0;
        result = Math.Round(d, decimals, MidpointRounding.ToNegativeInfinity);
        Console.WriteLine($"Round({d}, {decimals}, MidpointRounding.ToNegativeInfinity)

        d = 10.2;
        decimals = 0;
        result = Math.Round(d, decimals, MidpointRounding.ToPositiveInfinity);
        Console.WriteLine($"Round({d}, {decimals}, MidpointRounding.ToPositiveInfinity)

        d = 10.8;
        decimals = 0;
        result = Math.Round(d, decimals, MidpointRounding.ToZero);
        Console.WriteLine($"Round({d}, {decimals}, MidpointRounding.ToZero)
    }
}
```

Output

```
Round(10.2, 0, MidpointRounding.AwayFromZero)       = 10
Round(10.8, 0, MidpointRounding.ToEven)             = 11
Round(10.8, 0, MidpointRounding.ToNegativeInfinity) = 10
Round(10.2, 0, MidpointRounding.ToPositiveInfinity) = 11
Round(10.8, 0, MidpointRounding.ToZero)             = 10
```

Round(Double, MidpointRounding)

Math.Round(d, mode) rounds a double-precision floating-point value `d` to the nearest integer, and uses the specified rounding convention `mode` for midpoint values.

Syntax

The syntax of Round(d, mode) method is

```
Math.Round(Double d, MidpointRounding mode)
```

where

Parameter	Description

Parameter	Description
d	The double-precision floating-point number to be rounded.
mode	Specification for how to round the value `d` if it is midway between two other numbers.

Return Value

The method returns rounded Double value.

Example 8 – Round(Double, MidpointRounding)

In this example, we will take some double-precision floating-point numbers and round them with different modes using Math.Round() method.

C# Program

```
using System;

class Example {
    static void Main(string[] args) {
        Double d, result;

        d = 10.2;
        result = Math.Round(d, MidpointRounding.AwayFromZero);
        Console.WriteLine($"Round({d}, MidpointRounding.AwayFromZero)       = {result}")

        d = 10.8;
        result = Math.Round(d, MidpointRounding.ToEven);
        Console.WriteLine($"Round({d}, MidpointRounding.ToEven)             = {result}")

        d = 10.8;
        result = Math.Round(d, MidpointRounding.ToNegativeInfinity);
        Console.WriteLine($"Round({d}, MidpointRounding.ToNegativeInfinity) = {result}")

        d = 10.2;
        result = Math.Round(d, MidpointRounding.ToPositiveInfinity);
        Console.WriteLine($"Round({d}, MidpointRounding.ToPositiveInfinity) = {result}")

        d = 10.8;
        result = Math.Round(d, MidpointRounding.ToZero);
        Console.WriteLine($"Round({d}, MidpointRounding.ToZero)             = {result}")
    }
}
```

Output

```
Round(10.2, MidpointRounding.AwayFromZero)       = 10
Round(10.8, MidpointRounding.ToEven)             = 11
Round(10.8, MidpointRounding.ToNegativeInfinity) = 10
Round(10.2, MidpointRounding.ToPositiveInfinity) = 11
Round(10.8, MidpointRounding.ToZero)             = 10
```

Conclusion

In this C# Tutorial, we have learnt the syntax of C# Math.Round() method, and also learnt how to use this method with the help of examples.

C# Math

- C# Math.Abs()
- C# Math.Acos()
- C# Math.Acosh()
- C# Math.Asin()
- C# Math.Asinh()
- C# Math.Atan()
- C# Math.Atan2()
- C# Math.Atanh()
- C# Math.BigMul()
- C# Math.BitDecrement()
- C# Math.BitIncrement()
- C# Math.Cbrt()
- C# Math.Ceiling()
- C# Math.Clamp()
- C# Math.CopySign()
- C# Math.Cos()
- C# Math.Cosh()
- C# Math.DivRem()
- C# Math.Exp()
- C# Math.Floor()
- C# Math.FusedMultiplyAdd()
- C# Math.IEEERemainder()
- C# Math.ILogB()
- C# Math.Log()
- C# Math.Log10()

C# Tutorial