

C# Math.MinMagnitude() – Syntax & Examples

C# Math.MinMagnitude() – Examples

In this tutorial, we will learn about the C# Math.MinMagnitude() method, and learn how to use this method to get the minimum of two given numbers based on magnitude, with the help of examples.

MinMagnitude(Double, Double)

Math.MinMagnitude(x, y) returns the smaller magnitude of two double-precision floating-point numbers: `x`, `y`.

Syntax

The syntax of MinMagnitude() method is

```
Math.MinMagnitude(Double x, Double y)
```

where

Parameter	Description
x	The first of two double-precision floating-point numbers to compare.
y	The second of two double-precision floating-point numbers to compare.

Return Value

The method returns value.

Example 1 – MinMagnitude(x, y)

In this example, we will take different combinations for x and y and find minimum of x and y based on magnitude but not sign.

C# Program

```
using System;

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

        x = 10;
        y = 4;
        result = Math.MinMagnitude(x, y);
        Console.WriteLine($"MinMagnitude({x}, {y}) = {result}");

        x = 79;
        y = -853;
        result = Math.MinMagnitude(x, y);
        Console.WriteLine($"MinMagnitude({x}, {y}) = {result}");

        x = 10;
        y = 10;
        result = Math.MinMagnitude(x, y);
        Console.WriteLine($"MinMagnitude({x}, {y}) = {result}");
    }
}
```

Output

```
MinMagnitude(10, 4) = 4
MinMagnitude(79, -853) = 79
MinMagnitude(10, 10) = 10
```

MinMagnitude(10, 4) returns 4, since magnitude of 4 is less than that of 10.

MinMagnitude(79, -853) returns 79, since magnitude of 79 is less than that of -853. Magnitude of 79 is 79, and magnitude of -853 is 853.

MinMagnitude(10, 10) returns 10.

Conclusion

In this [C# Tutorial](#), we have learnt the syntax of C# Math.MinMagnitude() method, and also learnt how to use this method with the help of C# example programs.

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# Math.Log2()

◆ C# Math.Max()

◆ C# Math.MaxMagnitude()

◆ C# Math.Min()

⇒ **C# Math.MinMagnitude()**

◆ C# Math.Pow()

◆ C# Math.Round()

◆ C# Math.ScaleB()

◆ [C# Math.Sign\(\)](#)

◆ [C# Math.Sin\(\)](#)

◆ [C# Math.Sinh\(\)](#)

◆ [C# Math.Sqrt\(\)](#)

◆ [C# Math.Tan\(\)](#)

◆ [C# Math.Tanh\(\)](#)

◆ [C# Math.Truncate\(\)](#)

C# Tutorial

◆ [C# Tutorial](#)

◆ [C# List](#)

◆ [C# Dictionary](#)