Intro to CSharp Lesson 4

Built-in Methods From Class Math

Method – is simply a group of one or more lines of code that perform some task and are given their own name. In other languages these are called functions, procedures or subroutines. Methods are easily observed as they have parentheses immediately following the name. What is inside the parentheses is known as the argument. Not all methods contain an argument think about method named Main() as example. WriteLine() is a pre-written method to display whatever argument is placed inside parentheses.

access modifier> optional static modifier> return type> method name> ( parameter list>)

public static int Sum (int number1, int number2)

I find that often times writing the method header is the hardest part of writing a method. To help solve this problem, I usually require my students to write three lines of comments before every method that they write. The first comment line should start with the word task, followed by the task that this method will accomplish. The second line should start with the word incoming. Next, list all of the items that this method will need to accomplish its task. This will help you figure out what needs to go in the parameter list. The final comment line should start with the word outgoing. It will then describe what, if anything, is going to be returned by the method. Notice how the comments directly correlate with the different parts of the method header.

### [All Methods and Variables in Class Math](http://msdn.microsoft.com/en-us/library/system.math.aspx)

http://msdn.microsoft.com/en-us/library/system.math.aspx

This site lists all of the variables and method methods that are in class Math. Each method header is given.

Math Class

Definition

Namespace:

[System](https://docs.microsoft.com/en-us/dotnet/api/system?view=netcore-3.1)

Assembly:

System.Runtime.Extensions.dll

Assembly:

System.Runtime.dll

Assembly:

mscorlib.dll

Assembly:

netstandard.dll

Provides constants and static methods for trigonometric, logarithmic, and other common mathematical functions.

In this article

[Definition](https://docs.microsoft.com/en-us/dotnet/api/system.math?redirectedfrom=MSDN&view=netcore-3.1#definition)

[Examples](https://docs.microsoft.com/en-us/dotnet/api/system.math?redirectedfrom=MSDN&view=netcore-3.1#examples)

[Fields](https://docs.microsoft.com/en-us/dotnet/api/system.math?redirectedfrom=MSDN&view=netcore-3.1#fields)

[Methods](https://docs.microsoft.com/en-us/dotnet/api/system.math?redirectedfrom=MSDN&view=netcore-3.1#methods)

[Applies to](https://docs.microsoft.com/en-us/dotnet/api/system.math?redirectedfrom=MSDN&view=netcore-3.1#moniker-applies-to)

public ref class Math abstract sealed

public ref class Math sealed

C#Copy

public static class Math

C#Copy

public sealed class Math

type Math = class

Public Class Math

Public NotInheritable Class Math

Inheritance

[Object](https://docs.microsoft.com/en-us/dotnet/api/system.object?view=netcore-3.1)

Math

Examples

The following example uses several mathematical and trigonometric functions from the [Math](https://docs.microsoft.com/en-us/dotnet/api/system.math?view=netcore-3.1) class to calculate the inner angles of a trapezoid.

/// The following class represents simple functionality of the trapezoid.

using namespace System;

public ref class MathTrapezoidSample

{

private:

double m\_longBase;

double m\_shortBase;

double m\_leftLeg;

double m\_rightLeg;

public:

MathTrapezoidSample( double longbase, double shortbase, double leftLeg, double rightLeg )

{

m\_longBase = Math::Abs( longbase );

m\_shortBase = Math::Abs( shortbase );

m\_leftLeg = Math::Abs( leftLeg );

m\_rightLeg = Math::Abs( rightLeg );

}

private:

double GetRightSmallBase()

{

return (Math::Pow( m\_rightLeg, 2.0 ) - Math::Pow( m\_leftLeg, 2.0 ) + Math::Pow( m\_longBase, 2.0 ) + Math::Pow( m\_shortBase, 2.0 ) - 2 \* m\_shortBase \* m\_longBase) / (2 \* (m\_longBase - m\_shortBase));

}

public:

double GetHeight()

{

double x = GetRightSmallBase();

return Math::Sqrt( Math::Pow( m\_rightLeg, 2.0 ) - Math::Pow( x, 2.0 ) );

}

double GetSquare()

{

return GetHeight() \* m\_longBase / 2.0;

}

double GetLeftBaseRadianAngle()

{

double sinX = GetHeight() / m\_leftLeg;

return Math::Round( Math::Asin( sinX ), 2 );

}

double GetRightBaseRadianAngle()

{

double x = GetRightSmallBase();

double cosX = (Math::Pow( m\_rightLeg, 2.0 ) + Math::Pow( x, 2.0 ) - Math::Pow( GetHeight(), 2.0 )) / (2 \* x \* m\_rightLeg);

return Math::Round( Math::Acos( cosX ), 2 );

}

double GetLeftBaseDegreeAngle()

{

double x = GetLeftBaseRadianAngle() \* 180 / Math::PI;

return Math::Round( x, 2 );

}

double GetRightBaseDegreeAngle()

{

double x = GetRightBaseRadianAngle() \* 180 / Math::PI;

return Math::Round( x, 2 );

}

};

int main()

{

MathTrapezoidSample^ trpz = gcnew MathTrapezoidSample( 20.0,10.0,8.0,6.0 );

Console::WriteLine( "The trapezoid's bases are 20.0 and 10.0, the trapezoid's legs are 8.0 and 6.0" );

double h = trpz->GetHeight();

Console::WriteLine( "Trapezoid height is: {0}", h.ToString() );

double dxR = trpz->GetLeftBaseRadianAngle();

Console::WriteLine( "Trapezoid left base angle is: {0} Radians", dxR.ToString() );

double dyR = trpz->GetRightBaseRadianAngle();

Console::WriteLine( "Trapezoid right base angle is: {0} Radians", dyR.ToString() );

double dxD = trpz->GetLeftBaseDegreeAngle();

Console::WriteLine( "Trapezoid left base angle is: {0} Degrees", dxD.ToString() );

double dyD = trpz->GetRightBaseDegreeAngle();

Console::WriteLine( "Trapezoid left base angle is: {0} Degrees", dyD.ToString() );

}

C#Copy

Run

/// The following class represents simple functionality of the trapezoid.

using System;

namespace MathClassCS

{

class MathTrapezoidSample

{

private double m\_longBase;

private double m\_shortBase;

private double m\_leftLeg;

private double m\_rightLeg;

public MathTrapezoidSample(double longbase, double shortbase, double leftLeg, double rightLeg)

{

m\_longBase = Math.Abs(longbase);

m\_shortBase = Math.Abs(shortbase);

m\_leftLeg = Math.Abs(leftLeg);

m\_rightLeg = Math.Abs(rightLeg);

}

private double GetRightSmallBase()

{

return (Math.Pow(m\_rightLeg,2.0) - Math.Pow(m\_leftLeg,2.0) + Math.Pow(m\_longBase,2.0) + Math.Pow(m\_shortBase,2.0) - 2\* m\_shortBase \* m\_longBase)/ (2\*(m\_longBase - m\_shortBase));

}

public double GetHeight()

{

double x = GetRightSmallBase();

return Math.Sqrt(Math.Pow(m\_rightLeg,2.0) - Math.Pow(x,2.0));

}

public double GetSquare()

{

return GetHeight() \* m\_longBase / 2.0;

}

public double GetLeftBaseRadianAngle()

{

double sinX = GetHeight()/m\_leftLeg;

return Math.Round(Math.Asin(sinX),2);

}

public double GetRightBaseRadianAngle()

{

double x = GetRightSmallBase();

double cosX = (Math.Pow(m\_rightLeg,2.0) + Math.Pow(x,2.0) - Math.Pow(GetHeight(),2.0))/(2\*x\*m\_rightLeg);

return Math.Round(Math.Acos(cosX),2);

}

public double GetLeftBaseDegreeAngle()

{

double x = GetLeftBaseRadianAngle() \* 180/ Math.PI;

return Math.Round(x,2);

}

public double GetRightBaseDegreeAngle()

{

double x = GetRightBaseRadianAngle() \* 180/ Math.PI;

return Math.Round(x,2);

}

static void Main(string[] args)

{

MathTrapezoidSample trpz = new MathTrapezoidSample(20.0, 10.0, 8.0, 6.0);

Console.WriteLine("The trapezoid's bases are 20.0 and 10.0, the trapezoid's legs are 8.0 and 6.0");

double h = trpz.GetHeight();

Console.WriteLine("Trapezoid height is: " + h.ToString());

double dxR = trpz.GetLeftBaseRadianAngle();

Console.WriteLine("Trapezoid left base angle is: " + dxR.ToString() + " Radians");

double dyR = trpz.GetRightBaseRadianAngle();

Console.WriteLine("Trapezoid right base angle is: " + dyR.ToString() + " Radians");

double dxD = trpz.GetLeftBaseDegreeAngle();

Console.WriteLine("Trapezoid left base angle is: " + dxD.ToString() + " Degrees");

double dyD = trpz.GetRightBaseDegreeAngle();

Console.WriteLine("Trapezoid left base angle is: " + dyD.ToString() + " Degrees");

}

}

}

'The following class represents simple functionality of the trapezoid.

Class MathTrapezoidSample

Private m\_longBase As Double

Private m\_shortBase As Double

Private m\_leftLeg As Double

Private m\_rightLeg As Double

Public Sub New(ByVal longbase As Double, ByVal shortbase As Double, ByVal leftLeg As Double, ByVal rightLeg As Double)

m\_longBase = Math.Abs(longbase)

m\_shortBase = Math.Abs(shortbase)

m\_leftLeg = Math.Abs(leftLeg)

m\_rightLeg = Math.Abs(rightLeg)

End Sub

Private Function GetRightSmallBase() As Double

GetRightSmallBase = (Math.Pow(m\_rightLeg, 2) - Math.Pow(m\_leftLeg, 2) + Math.Pow(m\_longBase, 2) + Math.Pow(m\_shortBase, 2) - 2 \* m\_shortBase \* m\_longBase) / (2 \* (m\_longBase - m\_shortBase))

End Function

Public Function GetHeight() As Double

Dim x As Double = GetRightSmallBase()

GetHeight = Math.Sqrt(Math.Pow(m\_rightLeg, 2) - Math.Pow(x, 2))

End Function

Public Function GetSquare() As Double

GetSquare = GetHeight() \* m\_longBase / 2

End Function

Public Function GetLeftBaseRadianAngle() As Double

Dim sinX As Double = GetHeight() / m\_leftLeg

GetLeftBaseRadianAngle = Math.Round(Math.Asin(sinX), 2)

End Function

Public Function GetRightBaseRadianAngle() As Double

Dim x As Double = GetRightSmallBase()

Dim cosX As Double = (Math.Pow(m\_rightLeg, 2) + Math.Pow(x, 2) - Math.Pow(GetHeight(), 2)) / (2 \* x \* m\_rightLeg)

GetRightBaseRadianAngle = Math.Round(Math.Acos(cosX), 2)

End Function

Public Function GetLeftBaseDegreeAngle() As Double

Dim x As Double = GetLeftBaseRadianAngle() \* 180 / Math.PI

GetLeftBaseDegreeAngle = Math.Round(x, 2)

End Function

Public Function GetRightBaseDegreeAngle() As Double

Dim x As Double = GetRightBaseRadianAngle() \* 180 / Math.PI

GetRightBaseDegreeAngle = Math.Round(x, 2)

End Function

Public Shared Sub Main()

Dim trpz As MathTrapezoidSample = New MathTrapezoidSample(20, 10, 8, 6)

Console.WriteLine("The trapezoid's bases are 20.0 and 10.0, the trapezoid's legs are 8.0 and 6.0")

Dim h As Double = trpz.GetHeight()

Console.WriteLine("Trapezoid height is: " + h.ToString())

Dim dxR As Double = trpz.GetLeftBaseRadianAngle()

Console.WriteLine("Trapezoid left base angle is: " + dxR.ToString() + " Radians")

Dim dyR As Double = trpz.GetRightBaseRadianAngle()

Console.WriteLine("Trapezoid right base angle is: " + dyR.ToString() + " Radians")

Dim dxD As Double = trpz.GetLeftBaseDegreeAngle()

Console.WriteLine("Trapezoid left base angle is: " + dxD.ToString() + " Degrees")

Dim dyD As Double = trpz.GetRightBaseDegreeAngle()

Console.WriteLine("Trapezoid left base angle is: " + dyD.ToString() + " Degrees")

End Sub

End Class

Fields

|  |  |
| --- | --- |
| Table 1 | |
| [E](https://docs.microsoft.com/en-us/dotnet/api/system.math.e?view=netcore-3.1) | Represents the natural logarithmic base, specified by the constant, e. |
| [PI](https://docs.microsoft.com/en-us/dotnet/api/system.math.pi?view=netcore-3.1) | Represents the ratio of the circumference of a circle to its diameter, specified by the constant, π. |

Methods

|  |  |
| --- | --- |
| Table 2 | |
| [Abs(Decimal)](https://docs.microsoft.com/en-us/dotnet/api/system.math.abs?view=netcore-3.1#System_Math_Abs_System_Decimal_) | Returns the absolute value of a [Decimal](https://docs.microsoft.com/en-us/dotnet/api/system.decimal?view=netcore-3.1) number. |
| [Abs(Double)](https://docs.microsoft.com/en-us/dotnet/api/system.math.abs?view=netcore-3.1#System_Math_Abs_System_Double_) | Returns the absolute value of a double-precision floating-point number. |
| [Abs(Int16)](https://docs.microsoft.com/en-us/dotnet/api/system.math.abs?view=netcore-3.1#System_Math_Abs_System_Int16_) | Returns the absolute value of a 16-bit signed integer. |
| [Abs(Int32)](https://docs.microsoft.com/en-us/dotnet/api/system.math.abs?view=netcore-3.1#System_Math_Abs_System_Int32_) | Returns the absolute value of a 32-bit signed integer. |
| [Abs(Int64)](https://docs.microsoft.com/en-us/dotnet/api/system.math.abs?view=netcore-3.1#System_Math_Abs_System_Int64_) | Returns the absolute value of a 64-bit signed integer. |
| [Abs(SByte)](https://docs.microsoft.com/en-us/dotnet/api/system.math.abs?view=netcore-3.1#System_Math_Abs_System_SByte_) | Returns the absolute value of an 8-bit signed integer. |
| [Abs(Single)](https://docs.microsoft.com/en-us/dotnet/api/system.math.abs?view=netcore-3.1#System_Math_Abs_System_Single_) | Returns the absolute value of a single-precision floating-point number. |
| [Acos(Double)](https://docs.microsoft.com/en-us/dotnet/api/system.math.acos?view=netcore-3.1#System_Math_Acos_System_Double_) | Returns the angle whose cosine is the specified number. |
| [Acosh(Double)](https://docs.microsoft.com/en-us/dotnet/api/system.math.acosh?view=netcore-3.1#System_Math_Acosh_System_Double_) | Returns the angle whose hyperbolic cosine is the specified number. |
| [Asin(Double)](https://docs.microsoft.com/en-us/dotnet/api/system.math.asin?view=netcore-3.1#System_Math_Asin_System_Double_) | Returns the angle whose sine is the specified number. |
| [Asinh(Double)](https://docs.microsoft.com/en-us/dotnet/api/system.math.asinh?view=netcore-3.1#System_Math_Asinh_System_Double_) | Returns the angle whose hyperbolic sine is the specified number. |
| [Atan(Double)](https://docs.microsoft.com/en-us/dotnet/api/system.math.atan?view=netcore-3.1#System_Math_Atan_System_Double_) | Returns the angle whose tangent is the specified number. |
| [Atan2(Double, Double)](https://docs.microsoft.com/en-us/dotnet/api/system.math.atan2?view=netcore-3.1#System_Math_Atan2_System_Double_System_Double_) | Returns the angle whose tangent is the quotient of two specified numbers. |
| [Atanh(Double)](https://docs.microsoft.com/en-us/dotnet/api/system.math.atanh?view=netcore-3.1#System_Math_Atanh_System_Double_) | Returns the angle whose hyperbolic tangent is the specified number. |
| [BigMul(Int32, Int32)](https://docs.microsoft.com/en-us/dotnet/api/system.math.bigmul?view=netcore-3.1#System_Math_BigMul_System_Int32_System_Int32_) | Produces the full product of two 32-bit numbers. |
| [BitDecrement(Double)](https://docs.microsoft.com/en-us/dotnet/api/system.math.bitdecrement?view=netcore-3.1#System_Math_BitDecrement_System_Double_) | Returns the next smallest value that compares less than x. |
| [BitIncrement(Double)](https://docs.microsoft.com/en-us/dotnet/api/system.math.bitincrement?view=netcore-3.1#System_Math_BitIncrement_System_Double_) | Returns the next largest value that compares greater than x. |
| [Cbrt(Double)](https://docs.microsoft.com/en-us/dotnet/api/system.math.cbrt?view=netcore-3.1#System_Math_Cbrt_System_Double_) | Returns the cube root of a specified number. |
| [Ceiling(Decimal)](https://docs.microsoft.com/en-us/dotnet/api/system.math.ceiling?view=netcore-3.1#System_Math_Ceiling_System_Decimal_) | Returns the smallest integral value that is greater than or equal to the specified decimal number. |
| [Ceiling(Double)](https://docs.microsoft.com/en-us/dotnet/api/system.math.ceiling?view=netcore-3.1#System_Math_Ceiling_System_Double_) | Returns the smallest integral value that is greater than or equal to the specified double-precision floating-point number. |
| [Clamp(Byte, Byte, Byte)](https://docs.microsoft.com/en-us/dotnet/api/system.math.clamp?view=netcore-3.1#System_Math_Clamp_System_Byte_System_Byte_System_Byte_) | Returns value clamped to the inclusive range of min and max. |
| [Clamp(Decimal, Decimal, Decimal)](https://docs.microsoft.com/en-us/dotnet/api/system.math.clamp?view=netcore-3.1#System_Math_Clamp_System_Decimal_System_Decimal_System_Decimal_) | Returns value clamped to the inclusive range of min and max. |
| [Clamp(Double, Double, Double)](https://docs.microsoft.com/en-us/dotnet/api/system.math.clamp?view=netcore-3.1#System_Math_Clamp_System_Double_System_Double_System_Double_) | Returns value clamped to the inclusive range of min and max. |
| [Clamp(Int16, Int16, Int16)](https://docs.microsoft.com/en-us/dotnet/api/system.math.clamp?view=netcore-3.1#System_Math_Clamp_System_Int16_System_Int16_System_Int16_) | Returns value clamped to the inclusive range of min and max. |
| [Clamp(Int32, Int32, Int32)](https://docs.microsoft.com/en-us/dotnet/api/system.math.clamp?view=netcore-3.1#System_Math_Clamp_System_Int32_System_Int32_System_Int32_) | Returns value clamped to the inclusive range of min and max. |
| [Clamp(Int64, Int64, Int64)](https://docs.microsoft.com/en-us/dotnet/api/system.math.clamp?view=netcore-3.1#System_Math_Clamp_System_Int64_System_Int64_System_Int64_) | Returns value clamped to the inclusive range of min and max. |
| [Clamp(SByte, SByte, SByte)](https://docs.microsoft.com/en-us/dotnet/api/system.math.clamp?view=netcore-3.1#System_Math_Clamp_System_SByte_System_SByte_System_SByte_) | Returns value clamped to the inclusive range of min and max. |
| [Clamp(Single, Single, Single)](https://docs.microsoft.com/en-us/dotnet/api/system.math.clamp?view=netcore-3.1#System_Math_Clamp_System_Single_System_Single_System_Single_) | Returns value clamped to the inclusive range of min and max. |
| [Clamp(UInt16, UInt16, UInt16)](https://docs.microsoft.com/en-us/dotnet/api/system.math.clamp?view=netcore-3.1#System_Math_Clamp_System_UInt16_System_UInt16_System_UInt16_) | Returns value clamped to the inclusive range of min and max. |
| [Clamp(UInt32, UInt32, UInt32)](https://docs.microsoft.com/en-us/dotnet/api/system.math.clamp?view=netcore-3.1#System_Math_Clamp_System_UInt32_System_UInt32_System_UInt32_) | Returns value clamped to the inclusive range of min and max. |
| [Clamp(UInt64, UInt64, UInt64)](https://docs.microsoft.com/en-us/dotnet/api/system.math.clamp?view=netcore-3.1#System_Math_Clamp_System_UInt64_System_UInt64_System_UInt64_) | Returns value clamped to the inclusive range of min and max. |
| [CopySign(Double, Double)](https://docs.microsoft.com/en-us/dotnet/api/system.math.copysign?view=netcore-3.1#System_Math_CopySign_System_Double_System_Double_) | Returns a value with the magnitude of x and the sign of y. |
| [Cos(Double)](https://docs.microsoft.com/en-us/dotnet/api/system.math.cos?view=netcore-3.1#System_Math_Cos_System_Double_) | Returns the cosine of the specified angle. |
| [Cosh(Double)](https://docs.microsoft.com/en-us/dotnet/api/system.math.cosh?view=netcore-3.1#System_Math_Cosh_System_Double_) | Returns the hyperbolic cosine of the specified angle. |
| [DivRem(Int32, Int32, Int32)](https://docs.microsoft.com/en-us/dotnet/api/system.math.divrem?view=netcore-3.1#System_Math_DivRem_System_Int32_System_Int32_System_Int32__) | Calculates the quotient of two 32-bit signed integers and also returns the remainder in an output parameter. |
| [DivRem(Int64, Int64, Int64)](https://docs.microsoft.com/en-us/dotnet/api/system.math.divrem?view=netcore-3.1#System_Math_DivRem_System_Int64_System_Int64_System_Int64__) | Calculates the quotient of two 64-bit signed integers and also returns the remainder in an output parameter. |
| [Exp(Double)](https://docs.microsoft.com/en-us/dotnet/api/system.math.exp?view=netcore-3.1#System_Math_Exp_System_Double_) | Returns e raised to the specified power. |
| [Floor(Decimal)](https://docs.microsoft.com/en-us/dotnet/api/system.math.floor?view=netcore-3.1#System_Math_Floor_System_Decimal_) | Returns the largest integral value less than or equal to the specified decimal number. |
| [Floor(Double)](https://docs.microsoft.com/en-us/dotnet/api/system.math.floor?view=netcore-3.1#System_Math_Floor_System_Double_) | Returns the largest integral value less than or equal to the specified double-precision floating-point number. |
| [FusedMultiplyAdd(Double, Double, Double)](https://docs.microsoft.com/en-us/dotnet/api/system.math.fusedmultiplyadd?view=netcore-3.1#System_Math_FusedMultiplyAdd_System_Double_System_Double_System_Double_) | Returns (x \* y) + z, rounded as one ternary operation. |
| [IEEERemainder(Double, Double)](https://docs.microsoft.com/en-us/dotnet/api/system.math.ieeeremainder?view=netcore-3.1#System_Math_IEEERemainder_System_Double_System_Double_) | Returns the remainder resulting from the division of a specified number by another specified number. |
| [ILogB(Double)](https://docs.microsoft.com/en-us/dotnet/api/system.math.ilogb?view=netcore-3.1#System_Math_ILogB_System_Double_) | Returns the base 2 integer logarithm of a specified number. |
| [Log(Double)](https://docs.microsoft.com/en-us/dotnet/api/system.math.log?view=netcore-3.1#System_Math_Log_System_Double_) | Returns the natural (base e) logarithm of a specified number. |
| [Log(Double, Double)](https://docs.microsoft.com/en-us/dotnet/api/system.math.log?view=netcore-3.1#System_Math_Log_System_Double_System_Double_) | Returns the logarithm of a specified number in a specified base. |
| [Log10(Double)](https://docs.microsoft.com/en-us/dotnet/api/system.math.log10?view=netcore-3.1#System_Math_Log10_System_Double_) | Returns the base 10 logarithm of a specified number. |
| [Log2(Double)](https://docs.microsoft.com/en-us/dotnet/api/system.math.log2?view=netcore-3.1#System_Math_Log2_System_Double_) | Returns the base 2 logarithm of a specified number. |
| [Max(Byte, Byte)](https://docs.microsoft.com/en-us/dotnet/api/system.math.max?view=netcore-3.1#System_Math_Max_System_Byte_System_Byte_) | Returns the larger of two 8-bit unsigned integers. |
| [Max(Decimal, Decimal)](https://docs.microsoft.com/en-us/dotnet/api/system.math.max?view=netcore-3.1#System_Math_Max_System_Decimal_System_Decimal_) | Returns the larger of two decimal numbers. |
| [Max(Double, Double)](https://docs.microsoft.com/en-us/dotnet/api/system.math.max?view=netcore-3.1#System_Math_Max_System_Double_System_Double_) | Returns the larger of two double-precision floating-point numbers. |
| [Max(Int16, Int16)](https://docs.microsoft.com/en-us/dotnet/api/system.math.max?view=netcore-3.1#System_Math_Max_System_Int16_System_Int16_) | Returns the larger of two 16-bit signed integers. |
| [Max(Int32, Int32)](https://docs.microsoft.com/en-us/dotnet/api/system.math.max?view=netcore-3.1#System_Math_Max_System_Int32_System_Int32_) | Returns the larger of two 32-bit signed integers. |
| [Max(Int64, Int64)](https://docs.microsoft.com/en-us/dotnet/api/system.math.max?view=netcore-3.1#System_Math_Max_System_Int64_System_Int64_) | Returns the larger of two 64-bit signed integers. |
| [Max(SByte, SByte)](https://docs.microsoft.com/en-us/dotnet/api/system.math.max?view=netcore-3.1#System_Math_Max_System_SByte_System_SByte_) | Returns the larger of two 8-bit signed integers. |
| [Max(Single, Single)](https://docs.microsoft.com/en-us/dotnet/api/system.math.max?view=netcore-3.1#System_Math_Max_System_Single_System_Single_) | Returns the larger of two single-precision floating-point numbers. |
| [Max(UInt16, UInt16)](https://docs.microsoft.com/en-us/dotnet/api/system.math.max?view=netcore-3.1#System_Math_Max_System_UInt16_System_UInt16_) | Returns the larger of two 16-bit unsigned integers. |
| [Max(UInt32, UInt32)](https://docs.microsoft.com/en-us/dotnet/api/system.math.max?view=netcore-3.1#System_Math_Max_System_UInt32_System_UInt32_) | Returns the larger of two 32-bit unsigned integers. |
| [Max(UInt64, UInt64)](https://docs.microsoft.com/en-us/dotnet/api/system.math.max?view=netcore-3.1#System_Math_Max_System_UInt64_System_UInt64_) | Returns the larger of two 64-bit unsigned integers. |
| [MaxMagnitude(Double, Double)](https://docs.microsoft.com/en-us/dotnet/api/system.math.maxmagnitude?view=netcore-3.1#System_Math_MaxMagnitude_System_Double_System_Double_) | Returns the larger magnitude of two double-precision floating-point numbers. |
| [Min(Byte, Byte)](https://docs.microsoft.com/en-us/dotnet/api/system.math.min?view=netcore-3.1#System_Math_Min_System_Byte_System_Byte_) | Returns the smaller of two 8-bit unsigned integers. |
| [Min(Decimal, Decimal)](https://docs.microsoft.com/en-us/dotnet/api/system.math.min?view=netcore-3.1#System_Math_Min_System_Decimal_System_Decimal_) | Returns the smaller of two decimal numbers. |
| [Min(Double, Double)](https://docs.microsoft.com/en-us/dotnet/api/system.math.min?view=netcore-3.1#System_Math_Min_System_Double_System_Double_) | Returns the smaller of two double-precision floating-point numbers. |
| [Min(Int16, Int16)](https://docs.microsoft.com/en-us/dotnet/api/system.math.min?view=netcore-3.1#System_Math_Min_System_Int16_System_Int16_) | Returns the smaller of two 16-bit signed integers. |
| [Min(Int32, Int32)](https://docs.microsoft.com/en-us/dotnet/api/system.math.min?view=netcore-3.1#System_Math_Min_System_Int32_System_Int32_) | Returns the smaller of two 32-bit signed integers. |
| [Min(Int64, Int64)](https://docs.microsoft.com/en-us/dotnet/api/system.math.min?view=netcore-3.1#System_Math_Min_System_Int64_System_Int64_) | Returns the smaller of two 64-bit signed integers. |
| [Min(SByte, SByte)](https://docs.microsoft.com/en-us/dotnet/api/system.math.min?view=netcore-3.1#System_Math_Min_System_SByte_System_SByte_) | Returns the smaller of two 8-bit signed integers. |
| [Min(Single, Single)](https://docs.microsoft.com/en-us/dotnet/api/system.math.min?view=netcore-3.1#System_Math_Min_System_Single_System_Single_) | Returns the smaller of two single-precision floating-point numbers. |
| [Min(UInt16, UInt16)](https://docs.microsoft.com/en-us/dotnet/api/system.math.min?view=netcore-3.1#System_Math_Min_System_UInt16_System_UInt16_) | Returns the smaller of two 16-bit unsigned integers. |
| [Min(UInt32, UInt32)](https://docs.microsoft.com/en-us/dotnet/api/system.math.min?view=netcore-3.1#System_Math_Min_System_UInt32_System_UInt32_) | Returns the smaller of two 32-bit unsigned integers. |
| [Min(UInt64, UInt64)](https://docs.microsoft.com/en-us/dotnet/api/system.math.min?view=netcore-3.1#System_Math_Min_System_UInt64_System_UInt64_) | Returns the smaller of two 64-bit unsigned integers. |
| [MinMagnitude(Double, Double)](https://docs.microsoft.com/en-us/dotnet/api/system.math.minmagnitude?view=netcore-3.1#System_Math_MinMagnitude_System_Double_System_Double_) | Returns the smaller magnitude of two double-precision floating-point numbers. |
| [Pow(Double, Double)](https://docs.microsoft.com/en-us/dotnet/api/system.math.pow?view=netcore-3.1#System_Math_Pow_System_Double_System_Double_) | Returns a specified number raised to the specified power. |
| [Round(Decimal)](https://docs.microsoft.com/en-us/dotnet/api/system.math.round?view=netcore-3.1#System_Math_Round_System_Decimal_) | Rounds a decimal value to the nearest integral value, and rounds midpoint values to the nearest even number. |
| [Round(Decimal, Int32)](https://docs.microsoft.com/en-us/dotnet/api/system.math.round?view=netcore-3.1#System_Math_Round_System_Decimal_System_Int32_) | Rounds a decimal value to a specified number of fractional digits, and rounds midpoint values to the nearest even number. |
| [Round(Decimal, Int32, MidpointRounding)](https://docs.microsoft.com/en-us/dotnet/api/system.math.round?view=netcore-3.1#System_Math_Round_System_Decimal_System_Int32_System_MidpointRounding_) | Rounds a decimal value to a specified number of fractional digits, and uses the specified rounding convention for midpoint values. |
| [Round(Decimal, MidpointRounding)](https://docs.microsoft.com/en-us/dotnet/api/system.math.round?view=netcore-3.1#System_Math_Round_System_Decimal_System_MidpointRounding_) | Rounds a decimal value to the nearest integer, and uses the specified rounding convention for midpoint values. |
| [Round(Double)](https://docs.microsoft.com/en-us/dotnet/api/system.math.round?view=netcore-3.1#System_Math_Round_System_Double_) | Rounds a double-precision floating-point value to the nearest integral value, and rounds midpoint values to the nearest even number. |
| [Round(Double, Int32)](https://docs.microsoft.com/en-us/dotnet/api/system.math.round?view=netcore-3.1#System_Math_Round_System_Double_System_Int32_) | Rounds a double-precision floating-point value to a specified number of fractional digits, and rounds midpoint values to the nearest even number. |
| [Round(Double, Int32, MidpointRounding)](https://docs.microsoft.com/en-us/dotnet/api/system.math.round?view=netcore-3.1#System_Math_Round_System_Double_System_Int32_System_MidpointRounding_) | Rounds a double-precision floating-point value to a specified number of fractional digits, and uses the specified rounding convention for midpoint values. |
| [Round(Double, MidpointRounding)](https://docs.microsoft.com/en-us/dotnet/api/system.math.round?view=netcore-3.1#System_Math_Round_System_Double_System_MidpointRounding_) | Rounds a double-precision floating-point value to the nearest integer, and uses the specified rounding convention for midpoint values. |
| [ScaleB(Double, Int32)](https://docs.microsoft.com/en-us/dotnet/api/system.math.scaleb?view=netcore-3.1#System_Math_ScaleB_System_Double_System_Int32_) | Returns x \* 2^n computed efficiently. |
| [Sign(Decimal)](https://docs.microsoft.com/en-us/dotnet/api/system.math.sign?view=netcore-3.1#System_Math_Sign_System_Decimal_) | Returns an integer that indicates the sign of a decimal number. |
| [Sign(Double)](https://docs.microsoft.com/en-us/dotnet/api/system.math.sign?view=netcore-3.1#System_Math_Sign_System_Double_) | Returns an integer that indicates the sign of a double-precision floating-point number. |
| [Sign(Int16)](https://docs.microsoft.com/en-us/dotnet/api/system.math.sign?view=netcore-3.1#System_Math_Sign_System_Int16_) | Returns an integer that indicates the sign of a 16-bit signed integer. |
| [Sign(Int32)](https://docs.microsoft.com/en-us/dotnet/api/system.math.sign?view=netcore-3.1#System_Math_Sign_System_Int32_) | Returns an integer that indicates the sign of a 32-bit signed integer. |
| [Sign(Int64)](https://docs.microsoft.com/en-us/dotnet/api/system.math.sign?view=netcore-3.1#System_Math_Sign_System_Int64_) | Returns an integer that indicates the sign of a 64-bit signed integer. |
| [Sign(SByte)](https://docs.microsoft.com/en-us/dotnet/api/system.math.sign?view=netcore-3.1#System_Math_Sign_System_SByte_) | Returns an integer that indicates the sign of an 8-bit signed integer. |
| [Sign(Single)](https://docs.microsoft.com/en-us/dotnet/api/system.math.sign?view=netcore-3.1#System_Math_Sign_System_Single_) | Returns an integer that indicates the sign of a single-precision floating-point number. |
| [Sin(Double)](https://docs.microsoft.com/en-us/dotnet/api/system.math.sin?view=netcore-3.1#System_Math_Sin_System_Double_) | Returns the sine of the specified angle. |
| [Sinh(Double)](https://docs.microsoft.com/en-us/dotnet/api/system.math.sinh?view=netcore-3.1#System_Math_Sinh_System_Double_) | Returns the hyperbolic sine of the specified angle. |
| [Sqrt(Double)](https://docs.microsoft.com/en-us/dotnet/api/system.math.sqrt?view=netcore-3.1#System_Math_Sqrt_System_Double_) | Returns the square root of a specified number. |
| [Tan(Double)](https://docs.microsoft.com/en-us/dotnet/api/system.math.tan?view=netcore-3.1#System_Math_Tan_System_Double_) | Returns the tangent of the specified angle. |
| [Tanh(Double)](https://docs.microsoft.com/en-us/dotnet/api/system.math.tanh?view=netcore-3.1#System_Math_Tanh_System_Double_) | Returns the hyperbolic tangent of the specified angle. |
| [Truncate(Decimal)](https://docs.microsoft.com/en-us/dotnet/api/system.math.truncate?view=netcore-3.1#System_Math_Truncate_System_Decimal_) | Calculates the integral part of a specified decimal number. |
| [Truncate(Double)](https://docs.microsoft.com/en-us/dotnet/api/system.math.truncate?view=netcore-3.1#System_Math_Truncate_System_Double_) | Calculates the integral part of a specified double-precision floating-point number. |

Applies to

.NET

5.0 Preview 1

.NET Core

3.13.02.22.12.01.11.0

.NET Framework

4.84.7.24.7.14.74.6.24.6.14.64.5.24.5.14.54.03.53.02.01.1

.NET Standard

2.12.01.61.51.41.31.21.11.0

UWP

10.0

Xamarin.Android

7.1

Xamarin.iOS

10.8

Xamarin.Mac

3.0