# Math.Log Method

名前空間: WS.Theia.ExtremelyPrecise

アセンブリ: ExtremelyPrecise.dll

指定した数値の対数を返します。

# オーバーロード

|  |  |
| --- | --- |
| Log(Rational) | 指定した数の自然 (底 e) 対数を返します。 |
| Log(Rational,Rational) | 指定した数値の指定した底での対数を返します。 |

# Log(Rational)

指定した数の自然 (底 e) 対数を返します。

public static WS.Theia.ExtremelyPrecise.Rational Log(WS.Theia.ExtremelyPrecise.Rational value);

## パラメーター

value　Rational  
対数を求める対象の数値。

## 戻り値

Rational  
次の表に示した値のいずれか

|  |  |
| --- | --- |
| valueパラメーター | 戻り値 |
| 正 | value の自然対数。つまり、ln value または log e value |
| 0 | NegativeInfinity |
| 負 | NaN |
| NaN | NaN |
| PositiveInfinity | PositiveInfinity |

## 例

次の例はLog(Rational)メソッドの使用例です。

using System;  
using WS.Theia.ExtremelyPrecise;  
public class Example  
{  
 public static void Main()  
 {  
 Console.WriteLine(" Evaluate this identity with selected values for X:");  
 Console.WriteLine(" ln(x) = 1 / log[X](B)");  
 Console.WriteLine();  
   
 double[] XArgs = { 1.2, 4.9, 9.9, 0.1 };  
   
 foreach (double argX in XArgs)  
 {  
 // Find natural log of argX.  
 Console.WriteLine(" Math.Log({0}) = {1:E16}",  
 argX, Math.Log(argX));  
  
 // Evaluate 1 / log[X](e).  
 Console.WriteLine(" 1.0 / Math.Log(e, {0}) = {1:E16}",  
 argX, 1.0 / Math.Log(Math.E, argX));  
 Console.WriteLine();  
 }  
 }   
}  
// This example displays the following output:  
// Evaluate this identity with selected values for X:  
// ln(x) = 1 / log[X](B)  
//   
// Math.Log(1.2) = 1.8232155679395459E-001  
// 1.0 / Math.Log(e, 1.2) = 1.8232155679395459E-001  
//   
// Math.Log(4.9) = 1.5892352051165810E+000  
// 1.0 / Math.Log(e, 4.9) = 1.5892352051165810E+000  
//   
// Math.Log(9.9) = 2.2925347571405443E+000  
// 1.0 / Math.Log(e, 9.9) = 2.2925347571405443E+000  
//   
// Math.Log(0.1) = -2.3025850929940455E+000  
// 1.0 / Math.Log(e, 0.1) = -2.3025850929940455E+000

# 注釈

eは約2.71828の数学定数です。Log(Rational)メソッドはパラメーターのeの対数を算出します。Exp(Rational)メソッドとは逆の動作になります。

# Log(Rational,Rational)

指定した数値の指定した底での対数を返します。

public static WS.Theia.ExtremelyPrecise.Rational Log(WS.Theia.ExtremelyPrecise.Rational value, WS.Theia.ExtremelyPrecise.Rational newBase);

## パラメーター

value　Rational  
対数を求める対象の数値。

newBase　Rational  
対数の底。

## 戻り値

Rational  
次の表に示した値のいずれか (+Infinity は PositiveInfinity、-Infinity は NegativeInfinity、NaN は NaN をそれぞれ示しています。)

|  |  |  |
| --- | --- | --- |
| value | newBase | 戻り値 |
| value > 0 | (0 < newBase < 1) - または -( newBase > 1) | lognewBase(a) |
| value < 0 | (任意の値) | NaN |
| (任意の値) | newBase < 0 | NaN |
| value != 1 | newBase = 0 | NaN |
| value != 1 | newBase = +Infinity | NaN |
| value = NaN | (任意の値) | NaN |
| (任意の値) | newBase = NaN | NaN |
| (任意の値) | newBase = 1 | NaN |
| value = 0 | 0 < newBase < 1 | +Infinity |
| value = 0 | newBase > 1 | -Infinity |
| value = +無限大 | 0 <newBase< 1 | -Infinity |
| value = +無限大 | newBase> 1 | +Infinity |
| value = 1 | newBase = 0 | 0 |
| value = 1 | newBase = +Infinity | 0 |

## 例

次の例はLog(Rational、Rational)メソッドの使用例です。

// Example for the Math.Log( Rational ) and Math.Log( Rational, Rational ) methods.  
using System;  
using WS.Theia.ExtremelyPrecise;  
  
class LogDLogDD  
{  
 public static void Main()   
 {  
 Console.WriteLine(   
 "This example of Math.Log( Rational ) and " +  
 "Math.Log( Rational, Rational )\n" +  
 "generates the following output.\n" );  
 Console.WriteLine(   
 "Evaluate these identities with " +  
 "selected values for X and B (base):" );  
 Console.WriteLine( " log(B)[X] == 1 / log(X)[B]" );  
 Console.WriteLine( " log(B)[X] == ln[X] / ln[B]" );  
 Console.WriteLine( " log(B)[X] == log(B)[e] \* ln[X]" );  
  
 UseBaseAndArg(0.1, 1.2);  
 UseBaseAndArg(1.2, 4.9);  
 UseBaseAndArg(4.9, 9.9);  
 UseBaseAndArg(9.9, 0.1);  
 }  
 // Evaluate logarithmic identities that are functions of two arguments.  
 static void UseBaseAndArg(Rational argB, Rational argX)  
 {  
 // Evaluate log(B)[X] == 1 / log(X)[B].  
 Console.WriteLine(   
 "\n Math.Log({1}, {0}) == {2:E16}" +   
 "\n 1.0 / Math.Log({0}, {1}) == {3:E16}",   
 argB, argX, Math.Log(argX, argB),  
 1.0 / Math.Log(argB, argX) );  
  
 // Evaluate log(B)[X] == ln[X] / ln[B].  
 Console.WriteLine(   
 " Math.Log({1}) / Math.Log({0}) == {2:E16}",  
 argB, argX, Math.Log(argX) / Math.Log(argB) );  
  
 // Evaluate log(B)[X] == log(B)[e] \* ln[X].  
 Console.WriteLine(   
 "Math.Log(Math.E, {0}) \* Math.Log({1}) == {2:E16}",   
 argB, argX, Math.Log(Math.E, argB) \* Math.Log(argX) );  
 }  
}  
  
/\*  
This example of Math.Log( Rational ) and Math.Log( Rational, Rational )  
generates the following output.  
Evaluate these identities with selected values for X and B (base):  
 log(B)[X] == 1 / log(X)[B]  
 log(B)[X] == ln[X] / ln[B]  
 log(B)[X] == log(B)[e] \* ln[X]  
  
 Math.Log(1.2, 0.1) == -7.9181246047624818E-002  
 1.0 / Math.Log(0.1, 1.2) == -7.9181246047624818E-002  
 Math.Log(1.2) / Math.Log(0.1) == -7.9181246047624818E-002

Math.Log(Math.E, 0.1) \* Math.Log(1.2) == -7.9181246047624804E-002  
  
 Math.Log(4.9, 1.2) == 8.7166610085093179E+000  
 1.0 / Math.Log(1.2, 4.9) == 8.7166610085093161E+000  
 Math.Log(4.9) / Math.Log(1.2) == 8.7166610085093179E+000  
Math.Log(Math.E, 1.2) \* Math.Log(4.9) == 8.7166610085093179E+000  
  
 Math.Log(9.9, 4.9) == 1.4425396251981288E+000  
 1.0 / Math.Log(4.9, 9.9) == 1.4425396251981288E+000  
 Math.Log(9.9) / Math.Log(4.9) == 1.4425396251981288E+000  
Math.Log(Math.E, 4.9) \* Math.Log(9.9) == 1.4425396251981288E+000  
  
 Math.Log(0.1, 9.9) == -1.0043839404494075E+000  
 1.0 / Math.Log(9.9, 0.1) == -1.0043839404494075E+000  
 Math.Log(0.1) / Math.Log(9.9) == -1.0043839404494075E+000  
Math.Log(Math.E, 9.9) \* Math.Log(0.1) == -1.0043839404494077E+000  
\*/

# 適用対象

### .NET Core

2.0

### .NET Framework

4.6.1

### .NET Standard

2.0

### UWP

10.0.16299

### Xamarin.Android

8.0

### Xamarin.iOS

10.14

### Xamarin.Mac

3.8