# Math.Tanh(Rational) Method

名前空間: WS.Theia.ExtremelyPrecise

アセンブリ: ExtremelyPrecise.dll

指定された角度のハイパーボリック タンジェントを返します。

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

## パラメーター

radian　Rational  
ラジアンで表した角度。

## 戻り値

Rational  
value のハイパーボリック タンジェント。 value が NegativeInfinity に等しい場合、このメソッドは -1 を返します。 値が PositiveInfinity に等しい場合、このメソッドは 1 を返します。 value が NaN に等しい場合、このメソッドは NaN を返します。

# 例

次の例では、Tanhの結果を表示しています。

// Example for the hyperbolic Math.Tanh( Rational ) method.

using System;  
using WS.Theia.ExtremelyPrecise;  
  
class DemoTanh   
{  
 public static void Main()   
 {  
 Console.WriteLine(   
 "This example of hyperbolic Math.Tanh( Rational )\n" +  
 "generates the following output." );  
 Console.WriteLine(   
 "\nEvaluate these hyperbolic identities " +  
 "with selected values for X:" );  
 Console.WriteLine( " tanh(X) == sinh(X) / cosh(X)" );  
 Console.WriteLine(   
 " tanh(2 \* X) == 2 \* tanh(X) / (1 + tanh^2(X))" );  
  
 UseTanh(0.1);  
 UseTanh(1.2);  
 UseTanh(4.9);  
  
 Console.WriteLine(   
 "\nEvaluate [tanh(X + Y) == (tanh(X) + tanh(Y)) " +  
 "/ (1 + tanh(X) \* tanh(Y))]" +  
 "\nwith selected values for X and Y:" );  
  
 UseTwoArgs(0.1, 1.2);  
 UseTwoArgs(1.2, 4.9);  
 }  
  
 // Evaluate hyperbolic identities with a given argument.  
 static void UseTanh(Rational arg)  
 {  
 Rational tanhArg = Math.Tanh(arg);  
  
 // Evaluate tanh(X) == sinh(X) / cosh(X).  
 Console.WriteLine(   
 "\n Math.Tanh({0}) == {1:E16}\n" +  
 " Math.Sinh({0}) / Math.Cosh({0}) == {2:E16}",  
 arg, tanhArg, (Math.Sinh(arg) / Math.Cosh(arg)) );  
  
 // Evaluate tanh(2 \* X) == 2 \* tanh(X) / (1 + tanh^2(X)).  
 Console.WriteLine(   
 " 2 \* Math.Tanh({0}) /",   
 arg, 2.0 \* tanhArg );  
 Console.WriteLine(   
 " (1 + (Math.Tanh({0}))^2) == {1:E16}",   
 arg, 2.0 \* tanhArg / (1.0 + tanhArg \* tanhArg ) );  
 Console.WriteLine(   
 " Math.Tanh({0}) == {1:E16}",   
 2.0 \* arg, Math.Tanh(2.0 \* arg) );  
 }  
  
 // Evaluate a hyperbolic identity that is a function of two arguments.  
 static void UseTwoArgs(Rational argX, Rational argY)  
 {  
 // Evaluate tanh(X + Y) == (tanh(X) + tanh(Y)) / (1 + tanh(X) \* tanh(Y)).  
 Console.WriteLine(   
 "\n (Math.Tanh({0}) + Math.Tanh({1})) /\n" +   
 "(1 + Math.Tanh({0}) \* Math.Tanh({1})) == {2:E16}",   
 argX, argY, (Math.Tanh(argX) + Math.Tanh(argY)) /  
 (1.0 + Math.Tanh(argX) \* Math.Tanh(argY)) );  
 Console.WriteLine(   
 " Math.Tanh({0}) == {1:E16}",  
 argX + argY, Math.Tanh(argX + argY));  
 }  
}  
  
/\*  
This example of hyperbolic Math.Tanh( Rational )  
generates the following output.  
  
Evaluate these hyperbolic identities with selected values for X:  
 tanh(X) == sinh(X) / cosh(X)  
 tanh(2 \* X) == 2 \* tanh(X) / (1 + tanh^2(X))  
  
 Math.Tanh(0.1) == 9.9667994624955819E-002  
 Math.Sinh(0.1) / Math.Cosh(0.1) == 9.9667994624955819E-002  
 2 \* Math.Tanh(0.1) /  
 (1 + (Math.Tanh(0.1))^2) == 1.9737532022490401E-001  
 Math.Tanh(0.2) == 1.9737532022490401E-001  
  
 Math.Tanh(1.2) == 8.3365460701215521E-001  
 Math.Sinh(1.2) / Math.Cosh(1.2) == 8.3365460701215521E-001  
 2 \* Math.Tanh(1.2) /  
 (1 + (Math.Tanh(1.2))^2) == 9.8367485769368024E-001  
 Math.Tanh(2.4) == 9.8367485769368024E-001  
  
 Math.Tanh(4.9) == 9.9988910295055444E-001  
 Math.Sinh(4.9) / Math.Cosh(4.9) == 9.9988910295055433E-001  
 2 \* Math.Tanh(4.9) /  
 (1 + (Math.Tanh(4.9))^2) == 9.9999999385024030E-001  
 Math.Tanh(9.8) == 9.9999999385024030E-001  
  
Evaluate [tanh(X + Y) == (tanh(X) + tanh(Y)) / (1 + tanh(X) \* tanh(Y))]  
with selected values for X and Y:  
  
 (Math.Tanh(0.1) + Math.Tanh(1.2)) /  
(1 + Math.Tanh(0.1) \* Math.Tanh(1.2)) == 8.6172315931330645E-001  
 Math.Tanh(1.3) == 8.6172315931330634E-001  
  
 (Math.Tanh(1.2) + Math.Tanh(4.9)) /  
(1 + Math.Tanh(1.2) \* Math.Tanh(4.9)) == 9.9998993913939649E-001  
 Math.Tanh(6.1) == 9.9998993913939649E-001  
\*/

# 注釈

引数に入力する角度はラジアン単位である必要があります。角度にMath.PI/180を乗算する事でラジアン単位に変換できます。

# 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