Math.Exp(Rational) Method

名前空間: WS.Theia.ExtremelyPrecise

アセンブリ: ExtremelyPrecise.dll

指定した値で e を累乗した値を返します。

```
public static WS.Theia.ExtremelyPrecise.Rational
Exp(WS.Theia.ExtremelyPrecise.Rational value);
```

パラメーター

value　Rational

累乗を指定する数値。

戻り値

Rational

数値 e を value で累乗した値。value が NaN または PositiveInfinity のいずれかに等しい場合は、その値が返されます。 value が NegativeInfinity に等しい場合は、0 が返されます。

例

次の例では Exp(Rational)メソッドを使って E を累乗した結果を表示しています。

```
// Example for the Math.Exp( Rational ) method.
using System;
using WS.Theia.ExtremelyPrecise;

class ExpDemo
{
	public static void Main()
	{
    Console.WriteLine(
        "This example of Math.Exp( Rational ) " +
        "generates the following output.¥n" );
    Console.WriteLine(
        "Evaluate [e ^ ln(X) == ln(e ^ X) == X] " +
        "with selected values for X:" );

    UseLnExp(0.1);
    UseLnExp(1.2);
    UseLnExp(4.9);
    UseLnExp(9.9);

    Console.WriteLine(
        "¥nEvaluate these identities with " +
        "selected values for X and Y:" );
    Console.WriteLine( "      (e ^ X) * (e ^ Y) == e ^ (X + Y)" );
    Console.WriteLine( "      (e ^ X) ^ Y == e ^ (X * Y)" );
    Console.WriteLine( "      X ^ Y == e ^ (Y * ln(X))" );

    UseTwoArgs(0.1, 1.2);
    UseTwoArgs(1.2, 4.9);
    UseTwoArgs(4.9, 9.9);
}

// Evaluate logarithmic/exponential identity with a given argument.
static void UseLnExp(Rational arg)
{
    // Evaluate e ^ ln(X) == ln(e ^ X) == X.
    Console.WriteLine(
        "¥n      Math.Exp(Math.Log({0})) == {1:E16}¥n" +
        "      Math.Log(Math.Exp({0})) == {2:E16}",
        arg, Math.Exp(Math.Log(arg)), Math.Log(Math.Exp(arg)) );
}

// Evaluate exponential identities that are functions of two arguments.
static void UseTwoArgs(Rational argX, Rational argY)
```

```
    {
        // Evaluate (e ^ X) * (e ^ Y) == e ^ (X + Y).
        Console.WriteLine(
            "¥nMath.Exp({0}) * Math.Exp({1}) == {2:E16}" +
            "¥n             Math.Exp({0} + {1}) == {3:E16}",
            argX, argY, Math.Exp(argX) * Math.Exp(argY),
            Math.Exp(argX + argY) );

        // Evaluate (e ^ X) ^ Y == e ^ (X * Y).
        Console.WriteLine(
            " Math.Pow(Math.Exp({0}), {1}) == {2:E16}" +
            "¥n             Math.Exp({0} * {1}) == {3:E16}",
            argX, argY, Math.Pow(Math.Exp(argX), argY),
            Math.Exp(argX * argY) );

        // Evaluate X ^ Y == e ^ (Y * ln(X)).
        Console.WriteLine(
            "              Math.Pow({0}, {1}) == {2:E16}" +
            "¥nMath.Exp({1} * Math.Log({0})) == {3:E16}",
            argX, argY, Math.Pow(argX, argY),
            Math.Exp(argY * Math.Log(argX)) );
    }
}

/*
This example of Math.Exp( Rational ) generates the following output.

Evaluate [e ^ ln(X) == ln(e ^ X) == X] with selected values for X:

      Math.Exp(Math.Log(0.1)) == 1.0000000000000001E-001
      Math.Log(Math.Exp(0.1)) == 1.0000000000000008E-001

      Math.Exp(Math.Log(1.2)) == 1.2000000000000000E+000
      Math.Log(Math.Exp(1.2)) == 1.2000000000000000E+000
      Math.Exp(Math.Log(4.9)) == 4.9000000000000012E+000
      Math.Log(Math.Exp(4.9)) == 4.9000000000000004E+000
      Math.Exp(Math.Log(9.9)) == 9.9000000000000004E+000
      Math.Log(Math.Exp(9.9)) == 9.9000000000000004E+000

Evaluate these identities with selected values for X and Y:
   (e ^ X) * (e ^ Y) == e ^ (X + Y)
   (e ^ X) ^ Y == e ^ (X * Y)
   X ^ Y == e ^ (Y * ln(X))

Math.Exp(0.1) * Math.Exp(1.2) == 3.6692966676192444E+000
          Math.Exp(0.1 + 1.2) == 3.6692966676192444E+000
 Math.Pow(Math.Exp(0.1), 1.2) == 1.1274968515793757E+000
          Math.Exp(0.1 * 1.2) == 1.1274968515793757E+000
           Math.Pow(0.1, 1.2) == 6.3095734448019331E-002
Math.Exp(1.2 * Math.Log(0.1)) == 6.3095734448019344E-002

Math.Exp(1.2) * Math.Exp(4.9) == 4.4585777008251705E+002
          Math.Exp(1.2 + 4.9) == 4.4585777008251716E+002
 Math.Pow(Math.Exp(1.2), 4.9) == 3.5780924170885260E+002
          Math.Exp(1.2 * 4.9) == 3.5780924170885277E+002
           Math.Pow(1.2, 4.9) == 2.4433636334442981E+000
Math.Exp(4.9 * Math.Log(1.2)) == 2.4433636334442981E+000

Math.Exp(4.9) * Math.Exp(9.9) == 2.6764450551890982E+006
          Math.Exp(4.9 + 9.9) == 2.6764450551891015E+006
 Math.Pow(Math.Exp(4.9), 9.9) == 1.1684908531676833E+021
          Math.Exp(4.9 * 9.9) == 1.1684908531676829E+021
           Math.Pow(4.9, 9.9) == 6.8067718210957060E+006
Math.Exp(9.9 * Math.Log(4.9)) == 6.8067718210956985E+006
*/
```

注釈

e は約 2.71828 の数学定数です。Exp(Rational)メソッドは e を指定した数値で累乗します。Log(Rational)メソッドとは逆の動作になります。

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