Rational.NaN Property

名前空間: WS.Theia.ExtremelyPrecise
アセンブリ: ExtremelyPrecise.dll
数（NaN）を表す値を取得します。

```
public WS.Theia.ExtremelyPrecise.Rational NaN { get; }
```

プロパティ値
Boolean
値が非数（NaN）である Rational オブジェクト。

注釈

NaN は 0 を 0 で除算を行った場合等、数値として定義できない場合に返却されます。0 以外の被除数を 0 で除算した場合は、被除数と除数の符号により、PositiveInfinity、または、NegativeInfinity となります。

```
Rational zero = 0.0;
Console.WriteLine("{0} / {1} = {2}", zero, zero, zero/zero);
// The example displays the following output:
//               0 / 0 = NaN
```

また、NaN と他の値の演算子、又は、メソッドの操作結果は NaN となります。

```
Rational nan1 = Rational.NaN;

Console.WriteLine("{0} + {1} = {2}", 3, nan1, 3 + nan1);
Console.WriteLine("Abs({0}) = {1}", nan1,
WS.Theia.ExtremelyPrecise.Math.Abs(nan1));
// The example displays the following output:
//          3 + NaN = NaN
//          Abs(NaN) = NaN
```

一般的に NaN に演算子は使用する事ができませんが、比較演算子（Equals、CompereTo 等）を使用することができます。次の例では各演算子と値による比較結果を示しています。

```
using System;

public class Example
{
    public static void Main()
    {
        Console.WriteLine("NaN == NaN: {0}", Rational.NaN ==
Rational.NaN);
        Console.WriteLine("NaN != NaN: {0}", Rational.NaN != Rational.NaN);
        Console.WriteLine("NaN.Equals(NaN): {0}",
Rational.NaN.Equals(Rational.NaN));
        Console.WriteLine("! NaN.Equals(NaN): {0}", !
Rational.NaN.Equals(Rational.NaN));
        Console.WriteLine("IsNaN: {0}", Rational.IsNaN(Rational.NaN));

        Console.WriteLine("¥nNaN > NaN: {0}", Rational.NaN >
Rational.NaN);
        Console.WriteLine("NaN >= NaN: {0}", Rational.NaN >=
Rational.NaN);
        Console.WriteLine("NaN < NaN: {0}", Rational.NaN < Rational.NaN);
```

```
        Console.WriteLine("NaN < 100.0: {0}", Rational.NaN < 100.0);
        Console.WriteLine("NaN <= 100.0: {0}", Rational.NaN <= 100.0);
        Console.WriteLine("NaN >= 100.0: {0}", Rational.NaN > 100.0);
        Console.WriteLine("NaN.CompareTo(NaN): {0}",
Rational.NaN.CompareTo(Rational.NaN));
        Console.WriteLine("NaN.CompareTo(100.0): {0}",
Rational.NaN.CompareTo(100.0));
        Console.WriteLine("(100.0).CompareTo(Rational.NaN): {0}",new
Rational(100.0).CompareTo(Rational.NaN));
    }
}
// The example displays the following output:
//          NaN == NaN: False
//          NaN != NaN: True
//          NaN.Equals(NaN): True
//          ! NaN.Equals(NaN): False
//          IsNaN: True
//
//          NaN > NaN: False
//          NaN >= NaN: False
//          NaN < NaN: False
//          NaN < 100.0: False
//          NaN <= 100.0: False
//          NaN >= 100.0: False
//          NaN.CompareTo(NaN): 0
//          NaN.CompareTo(100.0): -1
//          (100.0).CompareTo(Rational.NaN): 1
```

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