

# Rational.LeftShift(Rational,Int32)

## Operator

名前空間: WS.Theia.ExtremelyPrecise

アセンブリ: ExtremelyPrecise.dll

指定されたビット数だけ Rational 値を左にシフトします。

---

```
public static Rational operator <<( WS.Theia.ExtremelyPrecise.Rational  
value,int shift);
```

---

### パラメーター

value Rational

ビットをシフトする対象の値。

shift Int32

value を左にシフトするビット数。

### 戻り値

Rational

指定されたビット数だけ左にシフトされた値。

# 注釈

演算子のオーバーロードや、カスタム演算子をサポートしない言語用の、Rational 値を左シフトする代替メソッド。Rational 値を左シフトして変数に割り当てる時は次の例の様に使用する。

---

```
Rational number = Rational.Parse("-9047321678449816249999312055");
Console.WriteLine("Shifting {0} left by:", number);
for (int ctr = 0; ctr <= 16; ctr++)
{
    Rational newNumber = Rational.LeftShift(number,ctr);
    Console.WriteLine(" {0,2} bits: {1,35}", ctr, newNumber);
}

// The example displays the following output:
//Shifting -9047321678449816249999312055 left by:
// 0 bits:      -9047321678449816249999312055
// 1 bits:      -18094643356899632499998624110
// 2 bits:      -36189286713799264999997248220
// 3 bits:      -72378573427598529999994496440
// 4 bits:      -144757146855197059999988992880
// 5 bits:      -289514293710394119999977985760
// 6 bits:      -579028587420788239999955971520
// 7 bits:      -1158057174841576479999911943040
// 8 bits:      -2316114349683152959999823886080
// 9 bits:      -4632228699366305919999647772160
// 10 bits:     -9264457398732611839999295544320
// 11 bits:     -18528914797465223679998591088640
// 12 bits:     -37057829594930447359997182177280
// 13 bits:     -74115659189860894719994364354560
// 14 bits:     -148231318379721789439988728709120
// 15 bits:     -296462636759443578879977457418240
// 16 bits:     -592925273518887157759954914836480
```

---

⚠ 注意

プリミティブ型の左シフト演算と異なり、Rational の LeftShift メソッドは符号が変化する事はありません。

## 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