C# Math.Truncate() – Syntax & Examples

C# Math.Truncate() – Examples

In this tutorial, we will learn about the C# Math.Truncate() method, and learn how to use this method to truncate a given decimal or double value, with the help of examples.

Truncate(Decimal)

Math.Truncate(d) returns the integral part of a specified decimal number `d`.

Syntax

The syntax of Truncate() method is

```
Math.Truncate(Decimal d)
```

where

Parameter	Description
d	The decimal value which is to be truncated.

Return Value

The method returns decimal value after truncation.

Example 1 – Truncate()

In this example, we will take some decimal values and use Truncate() method to get the integral part of the respective numbers.

C# Program

```
using System;
```

```
class Example {
    static void Main(string[] args) {
        Decimal d, result;

        d = 7.5785M;
        result = Math.Truncate(d);
        Console.WriteLine($"Truncate of {d} is {result}.");

        d = -7.5785M;
        result = Math.Truncate(d);
        Console.WriteLine($"Truncate of {d} is {result}.");

        d = 0.17525M;
        result = Math.Truncate(d);
        Console.WriteLine($"Truncate of {d} is {result}.");
    }
}
```

Output

```
Truncate of 7.5785 is 7.
Truncate of -7.5785 is -7.
Truncate of 0.17525 is 0.
```

Truncate(Double)

Math.Truncate(d) returns the integral part of a specified double-precision floating-point number `d`.

Syntax

The syntax of Truncate() method is

```
Math.Truncate(Double d)
```

where

Parameter	Description
d	The double value which is to be truncated.

Return Value

The method returns double value after truncation.

Example 2 – Truncate()

In this example, we will take some double-precision floating-point numbers and use Truncate() method to get the integral part of the respective numbers.

C# Program

```
using System;

class Example {
    static void Main(string[] args) {
        Double d, result;

        d = 7.5785;
        result = Math.Truncate(d);
        Console.WriteLine($"Truncate of {d} is {result}.");

        d = -7.5785;
        result = Math.Truncate(d);
        Console.WriteLine($"Truncate of {d} is {result}.");

        d = Double.NaN;
        result = Math.Truncate(d);
        Console.WriteLine($"Truncate of {d} is {result}.");

        d = Double.PositiveInfinity;
        result = Math.Truncate(d);
        Console.WriteLine($"Truncate of {d} is {result}.");

        d = Double.NegativeInfinity;
        result = Math.Truncate(d);
        Console.WriteLine($"Truncate of {d} is {result}.");
    }
}
```

Output

```
Truncate of 7.5785 is 7.
Truncate of -7.5785 is -7.
Truncate of NaN is NaN.
Truncate of ∞ is ∞.
Truncate of -∞ is -∞.
```

Conclusion

In this C# Tutorial, we have learnt the syntax of C# Math.Truncate() method, and also learnt how to use this method with the help of examples.

C# Math

- C# Math.Abs()

- C# Math.Acos()
- C# Math.Acosh()
- C# Math.Asin()
- C# Math.Asinh()
- C# Math.Atan()
- C# Math.Atan2()
- C# Math.Atanh()
- C# Math.BigMul()
- C# Math.BitDecrement()
- C# Math.BitIncrement()
- C# Math.Cbrt()
- C# Math.Ceiling()
- C# Math.Clamp()
- C# Math.CopySign()
- C# Math.Cos()
- C# Math.Cosh()
- C# Math.DivRem()
- C# Math.Exp()
- C# Math.Floor()
- C# Math.FusedMultiplyAdd()
- C# Math.IEEERemainder()
- C# Math.ILogB()
- C# Math.Log()
- C# Math.Log10()
- C# Math.Log2()
- C# Math.Max()
- C# Math.MaxMagnitude()
- C# Math.Min()
- C# Math.MinMagnitude()
- C# Math.Pow()
- C# Math.Round()
- C# Math.ScaleB()

C# Tutorial