**Math .Max Method:**

|  |  |  |
| --- | --- | --- |
| **Method** | **Definition** | **Example** |
| Max(UInt64,UInt64) | Returns the larger of two 64-bit unsigned integers. | Math. Max(104, 114)=114 |
| Max(UInt32,UInt32) | Returns the larger of two 32-bit unsigned integers. | Math. Max(103, 113)=113 |
| Max(UInt16,UInt16) | Returns the larger of two 16-bit unsigned integers. | Math. Max(102, 112)=112 |
| Max(Single, Single) | Returns the larger of two single-precision floating-point numbers. | Math. Max(5, 55)=55 |
| Max(SByte , SByte) | Returns the larger of two 8-bit signed integers. | Math. Max(1,111)=111 |
| Max(Byte , Byte) | Returns the larger of two 8-bit unsigned integers. | Math. Max(1,51)=51 |
| Max( Int 32 , Int32) | Returns the larger of two 32-bit signed integers. | Math. Max(-3,53)=53 |
| Max( Int 16 , Int16) | Returns the larger of two 16-bit signed integers. | Math. Max(-2,52)=52 |
| Max( Double , Double) | Returns the larger of two double-precision floating-point numbers. | Math. Max(6.0,56.0)=56.0 |
| Max( Decimal , Decimal) | Returns the larger of two decimal numbers. | Math. Max(7m,57m)=57m |
| Max(Int 64 ,Int 64) | Returns the larger of two 64-bit signed integers. | Math. Max(-4,54)=54 |

**Math .Min Method:**

|  |  |  |
| --- | --- | --- |
| **Method** | **Definition** | **Example** |
| Min(UInt64,UInt64) | Returns the Smaller of two 64-bit unsigned integers. | Math. Min(104, 114)=104 |
| Min(UInt32,UInt32) | Returns the Smaller of two 32-bit unsigned integers. | Math. Min(103, 113)=103 |
| Min(UInt16,UInt16) | Returns the Smaller of two 16-bit unsigned integers. | Math. Min(102, 112)=102 |
| Min(Single, Single) | Returns the Smaller of two single-precision floating-point numbers. | Math. Min(5, 55)=5 |
| Min(SByte , SByte) | Returns the Smaller of two 8-bit signed integers. | Math. Min(1,111)=1 |
| Min(Byte , Byte) | Returns the Smaller of two 8-bit unsigned integers. | Math. Min(1,51)=1 |
| Min( Int 32 , Int32) | Returns the Smaller of two 32-bit signed integers. | Math. Min(-3,53)=-3 |
| Min( Int 16 , Int16) | Returns the Smaller of two 16-bit signed integers. | Math. Min(-2,52)=-2 |
| Min ( Double , Double) | Returns the Smaller of two double-precision floating-point numbers. | Math. Min(6.0,56.0)=6.0 |
| Min( Decimal , Decimal) | Returns the Smaller of two decimal numbers. | Math. Min(7m,57m)=7m |
| Min (Int 64 ,Int 64) | Returns the Smaller of two 64-bit signed integers. | Math. Min(-4,54)=-4 |

# Math.Pow(Double, Double) Method:

|  |  |  |
| --- | --- | --- |
| **Method** | **Definition** | **Example** |
| Min(UInt64,UInt64) | Returns the Smaller of two 64-bit unsigned integers. | Math. Min(1,2)=1 |

| **Parameters** | **Return value** |
| --- | --- |
| x or y = NaN. | NaN |
| x = Any value except NaN; y = 0. | 1 |
| x = Negative Infinity; y < 0. | 0 |
| x = Negative Infinity; y is a positive odd integer. | Negative Infinity |
| x = Negative Infinity; y is positive but not an odd integer. | Positive Infinity |
| x < 0 but not Negative Infinity; y is not an integer, Negative Infinity, or Positive Infinity. | NaN |
| x = -1; y = Negative Infinity or Positive Infinity. | NaN |
| -1 < x < 1; y = Negative Infinity. | Positive Infinity |
| -1 < x < 1; y = Positive Infinity. | 0 |
| x < -1 or x > 1; y = Negative Infinity. | 0 |
| x < -1 or x > 1; y = Positive Infinity. | Positive Infinity |
| x = 0; y < 0. | Positive Infinity |
| x = 0; y > 0. | 0 |
| x = 1; y is any value except NaN. | 1 |
| x = Positive Infinity; y < 0. | 0 |
| x = Positive Infinity; y > 0. | Positive Infinity |

# Math.Round Method:

# 

Rounds a decimal value to a specified number of fractional digits, and uses the specified rounding convention for midpoint values.

|  |  |  |
| --- | --- | --- |
| **Method** | **Definition** | **Example** |
| Round(Decimal) | Rounds a decimal value to the nearest integral value, and rounds midpoint values to the nearest even number.. | Math. Round(4.5m)=4 |
| Round(Decimal, Midpoint Rounding) | Rounds a decimal value to the nearest integer, and uses the specified rounding convention for midpoint values. | Math. Round(12.8m)=13 |

# Math.Sign Method:

|  |  |  |
| --- | --- | --- |
| **Method** | **Definition** | **Example** |
| Sign(Single) | Returns an integer that indicates the sign of a single-precision floating-point number. | Math.Sign(0.0f)=0 |
| Sign(SByte) | Returns an integer that indicates the sign of an 8-bit signed integer. | Math.Sign(-101)<0 |
| Sign(Int 32) | Returns an integer that indicates the sign of a 32-bit signed integer. | Math.Sign(-3)<0 |
| Sign(Int 64) | Returns an integer that indicates the sign of a 64-bit signed integer. | Math.Sign(-4)<0 |
| Sign(Double) | Returns an integer that indicates the sign of a double-precision floating-point number. | Math.Sign(6.0)>0 |
| Sign(Decimal) | Returns an integer that indicates the sign of a decimal number. | Math.Sign(-7m)<0 |
| Sign(Int 16) | Returns an integer that indicates the sign of a 16-bit signed integer. | Math.Sign(-2)<0 |

# Math.Sqrt(Double) Method:

|  |  |  |
| --- | --- | --- |
| **Method** | **Definition** | **Example** |
| Sqrt (double) | Returns the square root of a specified number. | Math. Sqrt (138.8)=11.78 |

| **Parameter** | **Return value** |
| --- | --- |
| Zero or positive | The positive square root of d. |
| Negative | [NaN](https://docs.microsoft.com/en-us/dotnet/api/system.double.nan?view=netframework-4.8) |
| Equals [NaN](https://docs.microsoft.com/en-us/dotnet/api/system.double.nan?view=netframework-4.8) | [NaN](https://docs.microsoft.com/en-us/dotnet/api/system.double.nan?view=netframework-4.8) |
| Equals [PositiveInfinity](https://docs.microsoft.com/en-us/dotnet/api/system.double.positiveinfinity?view=netframework-4.8) | [PositiveInfinity](https://docs.microsoft.com/en-us/dotnet/api/system.double.positiveinfinity?view=netframework-4.8) |

# Math.Truncate Method:

|  |  |  |
| --- | --- | --- |
| **Method** | **Definition** | **Example** |
| Truncate(Decimal) | Calculates the integral part of a specified decimal number. | Math.Truncate (32.7865m;)=32 |
| Truncate(Double) | Calculates the integral part of a specified double-precision floating-point number. | Math.Truncate (-32.9012f)=-32 |

| **Parameter** | **Return value** |
| --- | --- |
| [NaN](https://docs.microsoft.com/en-us/dotnet/api/system.double.nan?view=netframework-4.8) | [NaN](https://docs.microsoft.com/en-us/dotnet/api/system.double.nan?view=netframework-4.8) |
| [NegativeInfinity](https://docs.microsoft.com/en-us/dotnet/api/system.double.negativeinfinity?view=netframework-4.8) | [NegativeInfinity](https://docs.microsoft.com/en-us/dotnet/api/system.double.negativeinfinity?view=netframework-4.8) |
| [PositiveInfinity](https://docs.microsoft.com/en-us/dotnet/api/system.double.positiveinfinity?view=netframework-4.8) | [PositiveInfinity](https://docs.microsoft.com/en-us/dotnet/api/system.double.positiveinfinity?view=netframework-4.8) |