Functions (Visual Basic)

Visual Studio 2015

The topics in this section contain tables of the Visual Basic run-time member functions.

Note
You can also create functions and call them. For more information, see Function Statement (Visual Basic) and How to: Create a Procedure that Returns a Value (Visual Basic).

In This Section

Conversion Functions (Visual Basic)

Math Functions (Visual Basic)

String Functions (Visual Basic)

Type Conversion Functions (Visual Basic)

CType Function (Visual Basic)

Related Sections

Visual Basic Language Reference

Visual Basic

Math Functions (Visual Basic)

Visual Studio 2015

The methods of the System.Math class provide trigonometric, logarithmic, and other common mathematical functions.

Remarks

The following table lists methods of the System.Math class. You can use these in a Visual Basic program.

.NET Framework method	Description
Abs	Returns the absolute value of a number.
Acos	Returns the angle whose cosine is the specified number.
Asin	Returns the angle whose sine is the specified number.
Atan	Returns the angle whose tangent is the specified number.
Atan2	Returns the angle whose tangent is the quotient of two specified numbers.
BigMul	Returns the full product of two 32-bit numbers.
Ceiling	Returns the smallest integral value that's greater than or equal to the specified **Decimal** or **Double**.
Cos	Returns the cosine of the specified angle.
Cosh	Returns the hyperbolic cosine of the specified angle.
DivRem	Returns the quotient of two 32-bit or 64-bit signed integers, and also returns the remainder in an output parameter.
Exp	Returns *e* (the base of natural logarithms) raised to the specified power.
Floor	Returns the largest integer that's less than or equal to the specified **Decimal** or **Double** number.
IEEERemainder	Returns the remainder that results from the division of a specified number by another specified number.

Log	Returns the natural (base *e*) logarithm of a specified number or the logarithm of a specified number in a specified base.
Log10	Returns the base 10 logarithm of a specified number.
Max	Returns the larger of two numbers.
Min	Returns the smaller of two numbers.
Pow	Returns a specified number raised to the specified power.
Round	Returns a **Decimal** or **Double** value rounded to the nearest integral value or to a specified number of fractional digits.
Sign	Returns an **Integer** value indicating the sign of a number.
Sin	Returns the sine of the specified angle.
Sinh	Returns the hyperbolic sine of the specified angle.
Sqrt	Returns the square root of a specified number.
Tan	Returns the tangent of the specified angle.
Tanh	Returns the hyperbolic tangent of the specified angle.
Truncate	Calculates the integral part of a specified **Decimal** or **Double** number.

To use these functions without qualification, import the System.Math namespace into your project by adding the following code to the top of your source file:

```
Imports System.Math
```

Example

This example uses the Abs method of the Math class to compute the absolute value of a number.

```
' Returns 50.3.
Dim MyNumber1 As Double = Math.Abs(50.3)
' Returns 50.3.
Dim MyNumber2 As Double = Math.Abs(-50.3)
```

Example

This example uses the Atan method of the Math class to calculate the value of pi.

```
Public Function GetPi() As Double
    ' Calculate the value of pi.
    Return 4.0 * Math.Atan(1.0)
End Function
```

Example

This example uses the Cos method of the Math class to return the cosine of an angle.

```
Public Function Sec(ByVal angle As Double) As Double
    ' Calculate the secant of angle, in radians.
    Return 1.0 / Math.Cos(angle)
End Function
```

Example

This example uses the Exp method of the Math class to return *e* raised to a power.

```
Public Function Sinh(ByVal angle As Double) As Double
    ' Calculate hyperbolic sine of an angle, in radians.
    Return (Math.Exp(angle) - Math.Exp(-angle)) / 2.0
End Function
```

Example

This example uses the Log method of the Math class to return the natural logarithm of a number.

```
Public Function Asinh(ByVal value As Double) As Double
    ' Calculate inverse hyperbolic sine, in radians.
    Return Math.Log(value + Math.Sqrt(value * value + 1.0))
End Function
```

Example

This example uses the Round method of the Math class to round a number to the nearest integer.

```
' Returns 3.
Dim MyVar2 As Double = Math.Round(2.8)
```

Example

This example uses the Sign method of the Math class to determine the sign of a number.

```
' Returns 1.
Dim MySign1 As Integer = Math.Sign(12)
' Returns -1.
Dim MySign2 As Integer = Math.Sign(-2.4)
' Returns 0.
Dim MySign3 As Integer = Math.Sign(0)
```

Example

This example uses the Sin method of the Math class to return the sine of an angle.

```
Public Function Csc(ByVal angle As Double) As Double
    ' Calculate cosecant of an angle, in radians.
    Return 1.0 / Math.Sin(angle)
End Function
```

Example

This example uses the Sqrt method of the Math class to calculate the square root of a number.

```
' Returns 2.
Dim MySqr1 As Double = Math.Sqrt(4)
' Returns 4.79583152331272.
Dim MySqr2 As Double = Math.Sqrt(23)
' Returns 0.
Dim MySqr3 As Double = Math.Sqrt(0)
' Returns NaN (not a number).
Dim MySqr4 As Double = Math.Sqrt(-4)
```

Example

This example uses the Tan method of the Math class to return the tangent of an angle.

```
Public Function Ctan(ByVal angle As Double) As Double
```

```
    ' Calculate cotangent of an angle, in radians.
    Return 1.0 / Math.Tan(angle)
End Function
```

Requirements

Class: Math

Namespace: System

Assembly: mscorlib (in mscorlib.dll)

See Also

Rnd
Randomize
NaN
Derived Math Functions (Visual Basic)
Arithmetic Operators (Visual Basic)

String Functions (Visual Basic)

Visual Studio 2015

The following table lists the functions that Visual Basic provides to search and manipulate strings.

.NET Framework method	Description
Asc, AscW	Returns an **Integer** value representing the character code corresponding to a character.
Chr, ChrW	Returns the character associated with the specified character code.
Filter	Returns a zero-based array containing a subset of a **String** array based on specified filter criteria.
Format	Returns a string formatted according to instructions contained in a format **String** expression.
FormatCurrency	Returns an expression formatted as a currency value using the currency symbol defined in the system control panel.
FormatDateTime	Returns a string expression representing a date/time value.
FormatNumber	Returns an expression formatted as a number.
FormatPercent	Returns an expression formatted as a percentage (that is, multiplied by 100) with a trailing % character.
InStr	Returns an integer specifying the start position of the first occurrence of one string within another.
InStrRev	Returns the position of the first occurrence of one string within another, starting from the right side of the string.
Join	Returns a string created by joining a number of substrings contained in an array.
LCase	Returns a string or character converted to lowercase.
Left	Returns a string containing a specified number of characters from the left side of a string.
Len	Returns an integer that contains the number of characters in a string.
LSet	Returns a left-aligned string containing the specified string adjusted to the specified length.

LTrim	Returns a string containing a copy of a specified string with no leading spaces.
Mid	Returns a string containing a specified number of characters from a string.
Replace	Returns a string in which a specified substring has been replaced with another substring a specified number of times.
Right	Returns a string containing a specified number of characters from the right side of a string.
RSet	Returns a right-aligned string containing the specified string adjusted to the specified length.
RTrim	Returns a string containing a copy of a specified string with no trailing spaces.
Space	Returns a string consisting of the specified number of spaces.
Split	Returns a zero-based, one-dimensional array containing a specified number of substrings.
StrComp	Returns -1, 0, or 1, based on the result of a string comparison.
StrConv	Returns a string converted as specified.
StrDup	Returns a string or object consisting of the specified character repeated the specified number of times.
StrReverse	Returns a string in which the character order of a specified string is reversed.
Trim	Returns a string containing a copy of a specified string with no leading or trailing spaces.
UCase	Returns a string or character containing the specified string converted to uppercase.

You can use the Option Compare statement to set whether strings are compared using a case-insensitive text sort order determined by your system's locale (**Text**) or by the internal binary representations of the characters (**Binary**). The default text comparison method is **Binary**.

Example

This example uses the **UCase** function to return an uppercase version of a string.

VB

```
' String to convert.
Dim LowerCase As String = "Hello World 1234"
' Returns "HELLO WORLD 1234".
Dim UpperCase As String = UCase(LowerCase)
```

Example

This example uses the **LTrim** function to strip leading spaces and the **RTrim** function to strip trailing spaces from a string

variable. It uses the **Trim** function to strip both types of spaces.

VB

```
' Initializes string.
Dim TestString As String = "  <-Trim->  "
Dim TrimString As String
' Returns "<-Trim->  ".
TrimString = LTrim(TestString)
' Returns "  <-Trim->".
TrimString = RTrim(TestString)
' Returns "<-Trim->".
TrimString = LTrim(RTrim(TestString))
' Using the Trim function alone achieves the same result.
' Returns "<-Trim->".
TrimString = Trim(TestString)
```

Example

This example uses the **Mid** function to return a specified number of characters from a string.

VB

```
' Creates text string.
Dim TestString As String = "Mid Function Demo"
' Returns "Mid".
Dim FirstWord As String = Mid(TestString, 1, 3)
' Returns "Demo".
Dim LastWord As String = Mid(TestString, 14, 4)
' Returns "Function Demo".
Dim MidWords As String = Mid(TestString, 5)
```

Example

This example uses **Len** to return the number of characters in a string.

VB

```
' Initializes variable.
Dim TestString As String = "Hello World"
' Returns 11.
Dim TestLen As Integer = Len(TestString)
```

Example

This example uses the **InStr** function to return the position of the first occurrence of one string within another.

VB

```
' String to search in.
Dim SearchString As String = "XXpXXpXXPXXP"
```

```
' Search for "P".
Dim SearchChar As String = "P"

Dim TestPos As Integer
' A textual comparison starting at position 4. Returns 6.
TestPos = InStr(4, SearchString, SearchChar, CompareMethod.Text)

' A binary comparison starting at position 1. Returns 9.
TestPos = InStr(1, SearchString, SearchChar, CompareMethod.Binary)

' If Option Compare is not set, or set to Binary, return 9.
' If Option Compare is set to Text, returns 3.
TestPos = InStr(SearchString, SearchChar)

' Returns 0.
TestPos = InStr(1, SearchString, "W")
```

Example

This example shows various uses of the **Format** function to format values using both **String** formats and user-defined formats. For the date separator (**/**), time separator (**:**), and the AM/PM indicators (**t** and **tt**), the actual formatted output displayed by your system depends on the locale settings the code is using. When times and dates are displayed in the development environment, the short time format and short date format of the code locale are used.

Note

For locales that use a 24-hour clock, the AM/PM indicators (**t** and **tt**) display nothing.

VB

```
Dim TestDateTime As Date = #1/27/2001 5:04:23 PM#
Dim TestStr As String
' Returns current system time in the system-defined long time format.
TestStr = Format(Now(), "Long Time")
' Returns current system date in the system-defined long date format.
TestStr = Format(Now(), "Long Date")
' Also returns current system date in the system-defined long date
' format, using the single letter code for the format.
TestStr = Format(Now(), "D")

' Returns the value of TestDateTime in user-defined date/time formats.
' Returns "5:4:23".
TestStr = Format(TestDateTime, "h:m:s")
' Returns "05:04:23 PM".
TestStr = Format(TestDateTime, "hh:mm:ss tt")
' Returns "Saturday, Jan 27 2001".
TestStr = Format(TestDateTime, "dddd, MMM d yyyy")
' Returns "17:04:23".
TestStr = Format(TestDateTime, "HH:mm:ss")
' Returns "23".
```

```
TestStr = Format(23)

' User-defined numeric formats.
' Returns "5,459.40".
TestStr = Format(5459.4, "##,##0.00")
' Returns "334.90".
TestStr = Format(334.9, "###0.00")
' Returns "500.00%".
TestStr = Format(5, "0.00%")
```

See Also

Keywords (Visual Basic)
Visual Basic Runtime Library Members
String Manipulation Summary (Visual Basic)

Dates and Times Summary (Visual Basic)

Visual Studio 2015

Visual Basic language keywords and run-time library members are organized by purpose and use.

Action	Language element
Get the current date or time.	Now, Today, TimeOfDay
Perform date calculations.	DateAdd, DateDiff, DatePart
Return a date.	DateSerial, DateValue, MonthName, WeekdayName
Return a time.	TimeSerial, TimeValue
Set the date or time.	DateString, TimeOfDay, TimeString, Today
Time a process.	Timer

See Also

Keywords (Visual Basic)
Visual Basic Runtime Library Members

DateAndTime Class

.NET Framework (current version)

The **DateAndTime** module contains the procedures and properties used in date and time operations.

Namespace: Microsoft.VisualBasic
Assembly: Microsoft.VisualBasic (in Microsoft.VisualBasic.dll)

Inheritance Hierarchy

System.Object
  Microsoft.VisualBasic.DateAndTime

Syntax

VB

```
<StandardModuleAttribute>
Public NotInheritable Class DateAndTime
```

Properties

	Name	Description
	DateString	Returns or sets a **String** value representing the current date according to your system.
	Now	Returns a **Date** value containing the current date and time according to your system.
	TimeOfDay	Returns or sets a **Date** value containing the current time of day according to your system.
	Timer	Returns a **Double** value representing the number of seconds elapsed since midnight.
	TimeString	Returns or sets a **String** value representing the current time of day according to your system.
	Today	Returns or sets a **Date** value containing the current date according to your system.

Methods

	Name	Description
	DateAdd(DateInterval, Double, DateTime)	Returns a **Date** value containing a date and time value to which a specified time interval has been added.
	DateAdd(String, Double, Object)	Returns a **Date** value containing a date and time value to which a specified time interval has been added.
	DateDiff(DateInterval, DateTime, DateTime, FirstDayOfWeek, FirstWeekOfYear)	Returns a **Long** value specifying the number of time intervals between two **Date** values.
	DateDiff(String, Object, Object, FirstDayOfWeek, FirstWeekOfYear)	Returns a **Long** value specifying the number of time intervals between two **Date** values.
	DatePart(DateInterval, DateTime, FirstDayOfWeek, FirstWeekOfYear)	Returns an **Integer** value containing the specified component of a given **Date** value.
	DatePart(String, Object, FirstDayOfWeek, FirstWeekOfYear)	Returns an **Integer** value containing the specified component of a given **Date** value.
	DateSerial(Int32, Int32, Int32)	Returns a **Date** value representing a specified year, month, and day, with the time information set to midnight (00:00:00).

S	DateValue(String)	Returns a **Date** value containing the date information represented by a string, with the time information set to midnight (00:00:00).
S	Day(DateTime)	Returns an **Integer** value from 1 through 31 representing the day of the month.
	Equals(Object)	Determines whether the specified object is equal to the current object.(Inherited from Object.)
	GetHashCode()	Serves as the default hash function. (Inherited from Object.)
	GetType()	Gets the Type of the current instance.(Inherited from Object.)
S	Hour(DateTime)	Returns an **Integer** value from 0 through 23 representing the hour of the day.
S	Minute(DateTime)	Returns an **Integer** value from 0 through 59 representing the minute of the hour.
S	Month(DateTime)	Returns an **Integer** value from 1 through 12 representing the month of the year.
S	MonthName(Int32, Boolean)	Returns a **String** value containing the name of the specified month.
S	Second(DateTime)	Returns an **Integer** value from 0 through 59 representing the second of the minute.
S	TimeSerial(Int32, Int32, Int32)	Returns a **Date** value representing a specified hour, minute, and second, with the date information set relative to January 1 of the year 1.
S	TimeValue(String)	Returns a **Date** value containing the time information represented by a string, with the date information set to January 1 of the year 1.
	ToString()	Returns a string that represents the current object. (Inherited from Object.)
S	Weekday(DateTime, FirstDayOfWeek)	Returns an **Integer** value containing a number representing the day of the week.
S	WeekdayName(Int32, Boolean, FirstDayOfWeek)	Returns a **String** value containing the name of the specified weekday.

	Year(DateTime)	Returns an **Integer** value from 1 through 9999 representing the year.

Remarks

This module supports the Visual Basic language keywords and run-time library members that get the current date or time, perform date calculations, return a date or time, set the date or time, or time the duration of a process.

Examples

This example uses the **Today** property to return the current system date.

```
Dim thisDate As Date
thisDate = Today
```

Version Information

.NET Framework
Available since 1.1
Silverlight
Available since 2.0

Thread Safety

Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

See Also

Microsoft.VisualBasic Namespace
Dates and Times Summary (Visual Basic)
Keywords (Visual Basic)
Visual Basic Runtime Library Members

Return to top

Math Summary (Visual Basic)

Visual Studio 2015

Visual Basic language keywords and run-time library members are organized by purpose and use.

Action	Language element
Derive trigonometric functions.	Atan, Cos, Sin, Tan
General calculations.	Exp, Log, Sqrt
Generate random numbers.	Randomize, Rnd
Get absolute value.	Abs
Get the sign of an expression.	Sign
Perform numeric conversions.	Fix, Int

See Also

Derived Math Functions (Visual Basic)
Keywords (Visual Basic)
Visual Basic Runtime Library Members

Derived Math Functions (Visual Basic)

Visual Studio 2015

The following table shows non-intrinsic math functions that can be derived from the intrinsic math functions of the System.Math object. You can access the intrinsic math functions by adding `Imports System.Math` to your file or project.

Function	Derived equivalents
Secant (Sec(x))	*1 / Cos(x)*
Cosecant (Csc(x))	*1 / Sin(x)*
Cotangent (Ctan(x))	*1 / Tan(x)*
Inverse sine (Asin(x))	*Atan(x / Sqrt(-x * x + 1))*
Inverse cosine (Acos(x))	*Atan(-x / Sqrt(-x * x + 1)) + 2 * Atan(1)*
Inverse secant (Asec(x))	*2 * Atan(1) – Atan(Sign(x) / Sqrt(x * x – 1))*
Inverse cosecant (Acsc(x))	*Atan(Sign(x) / Sqrt(x * x – 1))*
Inverse cotangent (Acot(x))	*2 * Atan(1) - Atan(x)*
Hyperbolic sine (Sinh(x))	*(Exp(x) – Exp(-x)) / 2*
Hyperbolic cosine (Cosh(x))	*(Exp(x) + Exp(-x)) / 2*
Hyperbolic tangent (Tanh(x))	*(Exp(x) – Exp(-x)) / (Exp(x) + Exp(-x))*
Hyperbolic secant (Sech(x))	*2 / (Exp(x) + Exp(-x))*
Hyperbolic cosecant (Csch(x))	*2 / (Exp(x) – Exp(-x))*
Hyperbolic cotangent (Coth(x))	*(Exp(x) + Exp(-x)) / (Exp(x) – Exp(-x))*
Inverse hyperbolic sine (Asinh(x))	*Log(x + Sqrt(x * x + 1))*
Inverse hyperbolic cosine (Acosh(x))	*Log(x + Sqrt(x * x – 1))*
Inverse hyperbolic tangent (Atanh(x))	*Log((1 + x) / (1 – x)) / 2*
Inverse hyperbolic secant (AsecH(x))	*Log((Sqrt(-x * x + 1) + 1) / x)*

Inverse hyperbolic cosecant (Acsch(x))	*Log((Sign(x) * Sqrt(x * x + 1) + 1) / x)*
Inverse hyperbolic cotangent (Acoth(x))	*Log((x + 1) / (x – 1)) / 2*

See Also

Math Functions (Visual Basic)