Some Modules, Structures, and Classes and their functions, properties and methods

Public Module **Strings** **In general don't use these functions – use the String class instead**
Member of: **Microsoft**.**VisualBasic**
Summary: The Strings module contains procedures used to perform string operations.

Public Function **FormatCurrency**(ByVal *Expression* As **Object**, Optional ByVal *NumDigitsAfterDecimal* As **Integer** = -1, ...) As **String**
Summary: Returns an expression formatted as a currency value using the currency symbol defined in the system control panel.
Parameters:
Expression: Required. Expression to be formatted.
NumDigitsAfterDecimal: Optional. Numeric value indicating how many places are displayed to the right of the decimal. Default value is –1, which indicates that the computer's regional settings are used.

Public Function **FormatNumber**(ByVal *Expression* As **Object**, Optional ByVal *NumDigitsAfterDecimal* As **Integer** = -1, ...) As **String**
Summary: Returns an expression formatted as a number.
Parameters:
Expression: Required. Expression to be formatted.
NumDigitsAfterDecimal: Optional. Numeric value indicating how many places are displayed to the right of the decimal. The default value is –1, which indicates that the computer's regional settings are used.

Public Function **Replace**(ByVal *Expression* As **String**, ByVal *Find* As **String**, ByVal *Replacement* As **String**, Optional ByVal *Start* As **Integer** = 1, Optional ByVal *Count* As **Integer** = -1, ...) As **String**
Summary: Returns a string in which a specified substring has been replaced with another substring a specified number of times.
Parameters:
Start: Optional. Position within Expression where substring search is to begin. If omitted, 1 is assumed.
Count: Optional. Number of substring substitutions to perform. If omitted, the default value is –1, which means "make all possible substitutions."
Expression: Required. String expression containing substring to replace.
Replacement: Required. Replacement substring.
Find: Required. Substring being searched for.
Return Values: If Find is zero-length or Nothing, Replace returns copy of Expression. If Replace is zero-length, Replace returns copy of Expression with no occurrences of Find. If Expression is zero-length or Nothing, or Start is greater than length of Expression, Replace returns Nothing. If Count is 0, Replace returns copy of Expression.

Public Module **Information**
Member of: **Microsoft**.**VisualBasic**
Summary: The Information module contains the procedures used to return, test for, or verify information.

Public Function **IsDate**(ByVal *Expression* As **Object**) As **Boolean**
Summary: Returns a Boolean value indicating whether an expression represents a valid Date value.
Parameters:
Expression: Required. Object expression.
Public Function **IsNumeric**(ByVal *Expression* As **Object**) As **Boolean**
Summary: Returns a Boolean value indicating whether an expression can be evaluated as a number.
Parameters:
Expression: Required. Object expression.

Public NotInheritable Class **Convert**
Summary: Converts a base data type to another base data type.

Public Shared Function **ToDateTime**(ByVal *value* As **String**) As **Date**

Summary: Converts the specified System.String representation of a date and time to an equivalent System.DateTime.
Parameters: *value*: A System.String containing a date and time to convert.
Return Values:
A System.DateTime equivalent to the value of value.-or- A System.DateTime equivalent to System.DateTime.MinValue if value is null.

Public Shared Function **ToDouble**(ByVal *value* As **String**) As **Double**
Summary: Converts the specified System.String representation of a number to an equivalent double-precision floating point number.
Parameters: *value*: A System.String containing a number to convert.
Return Values:
A double-precision floating point number equivalent to the value of value.-or- Zero if value is null.

Public Shared Function **ToInt32**(ByVal *value* As **String**) As **Integer**
Summary: Converts the specified System.String representation of a number to an equivalent 32-bit signed integer.
Parameters: *value*: A System.String containing a number to convert.
Return Values: A 32-bit signed integer equivalent to the value of value.-or- Zero if value is null.

Public Structure **Int32**
Summary: Represents a 32-bit signed integer.

Public Shared Function **Parse**(ByVal *s* As **String**) As **Integer**
Summary: Converts the string representation of a number to its 32-bit signed integer equivalent.
Parameters: *s*: A string containing a number to convert.
Return Values: A 32-bit signed integer equivalent to the number contained in s.

Public Shared Function **TryParse**(ByVal *s* As **String**, ByRef *result* As **Integer**) As **Boolean**
Summary: Converts the string representation of a number to its 32-bit signed integer equivalent. A return value indicates whether the operation succeeded.
Parameters:
s: A string containing a number to convert.
result: When this method returns, contains the 32-bit signed integer value equivalent to the number contained in s, if the conversion succeeded, or zero if the conversion failed. The conversion fails if the s parameter is null, is not of the correct format, or represents a number less than System.Int32.MinValue or greater than System.Int32.MaxValue. This parameter is passed uninitialized.
Return Values: true if s was converted successfully; otherwise, false.

(Same functions for structures **Double**, **Single, Integer**, etc.)

Public NotInheritable Class **Math**
Summary: Provides constants and static methods for trigonometric, logarithmic, and other common mathematical functions.

Public Shared Function **Abs**(ByVal *value* As **Double**) As **Double**
Summary: Returns the absolute value of a double-precision floating-point number.
Parameters: *value*: A number in the range System.Double.MinValue≤value≤System.Double.MaxValue.
Return Values: A double-precision floating-point number, x, such that 0 ≤ x ≤System.Double.MaxValue.

Public Shared Function **Ceiling**(ByVal *a* As **Double**) As **Double**
Summary: Returns the smallest integer greater than or equal to the specified double-precision floating-point number.
Parameters: *a*: A double-precision floating-point number.

Public Shared Function **Floor**(ByVal *d* As **Double**) As **Double**
Summary: Returns the largest integer less than or equal to the specified double-precision floating-point number.

Parameters: *d*: A double-precision floating-point number.

Public Shared Function **Max**(ByVal *val1* As **Double**, ByVal *val2* As **Double**) As **Double**
Summary: Returns the larger of two double-precision floating-point numbers.
Parameters:
val1: The first of two double-precision floating-point numbers to compare.
val2: The second of two double-precision floating-point numbers to compare.

Public Shared Function **Min**(ByVal *val1* As **Double**, ByVal *val2* As **Double**) As **Double**
Summary: Returns the smaller of two double-precision floating-point numbers.
Parameters:
val1: The first of two double-precision floating-point numbers to compare.
val2: The second of two double-precision floating-point numbers to compare.

Public Const **PI** As **Double** = 3.1415926535897931
Summary: Represents the ratio of the circumference of a circle to its diameter.
Public Shared Function **Pow**(ByVal *x* As **Double**, ByVal *y* As **Double**) As **Double**
Summary: Returns a specified number raised to the specified power.
Parameters:
y: A double-precision floating-point number that specifies a power.
x: A double-precision floating-point number to be raised to a power.

Public Shared Function **Round**(ByVal *value* As **Double**, ByVal *digits* As **Integer**) As **Double**
Summary: Rounds a double-precision floating-point value to the specified precision.
Parameters:
digits: The number of significant digits (precision) in the return value.
value: A double-precision floating-point number to be rounded.

Public Shared Function **Sqrt**(ByVal *d* As **Double**) As **Double**
Summary: Returns the square root of a specified number.
Parameters: *d*: A number.

Public Class **Random**
Summary: Represents a pseudo-random number generator, a device that produces a sequence of numbers that meet certain statistical requirements for randomness.

Public Sub **New**()
Summary: Initializes a new instance of the Random class, using a time-dependent default seed value.

Public Overridable Function **[Next]**(ByVal *maxValue* As **Integer**) As **Integer**
Summary: Returns a nonnegative random number less than the specified maximum.
Parameters: *maxValue*: The exclusive upper bound of the random number to be generated. maxValue must be greater than or equal to zero.
Return Values: A 32-bit signed integer greater than or equal to zero, and less than maxValue; that is, the range of return values includes zero but not maxValue.

Public Overridable Function **[Next]**(ByVal *minValue* As **Integer**, ByVal *maxValue* As **Integer**) As **Integer**
Summary: Returns a random number within a specified range.
Parameters:
minValue: The inclusive lower bound of the random number returned.
maxValue: The exclusive upper bound of the random number returned. maxValue must be greater than or equal to minValue.
Return Values: A 32-bit signed integer greater than or equal to minValue and less than maxValue; that is, the range of return values includes minValue but not maxValue. If minValue equals maxValue, minValue is returned.

Public Overridable Function **NextDouble**() As **Double**
Summary: Returns a random number between 0.0 and 1.0.
Return Values: A double-precision floating point number greater than or equal to 0.0, and less than 1.0.

Public NotInheritable Class **String**
Member of: **System**
Summary: Represents text as a series of Unicode characters.

Public ReadOnly Default Property **Chars**(ByVal *index* As **Integer**) As **Char**
Summary: Gets the character at a specified character position in the current System.String object.
Parameters:
index: A character position in the current System.String object.
Return Values: A Unicode character.
Exceptions:
System.IndexOutOfRangeException: index is greater than or equal to the length of this object or less than zero.

Public Function **CompareTo**(ByVal *strB* As **String**) As **Integer**
Summary: Compares this instance with a specified System.String object.
Parameters: *strB*: A System.String.
Return Values: A 32-bit signed integer indicating the lexical relationship between the two comparands. Value Condition Less than zero This instance is less than strB. Zero This instance is equal to strB. Greater than zero This instance is greater than strB.-or- strB is null.

Public Function **Contains**(ByVal *value* As **String**) As **Boolean**
Summary: Returns a value indicating whether the specified System.String object occurs within this string.
Parameters: *value*: The System.String object to seek.
Return Values: true if the value parameter occurs within this string, or if value is the empty string (""); otherwise, false.
Exceptions:
System.ArgumentNullException: value is null.

Public Function **IndexOf**(ByVal *value* As **String**, ByVal *startIndex* As **Integer**) As **Integer**
Summary: Reports the index of the first occurrence of the specified System.String in this instance. The search starts at a specified character position.
Parameters:
value: The System.String to seek.
startIndex: The search starting position.
Return Values: The index position of value if that string is found, or -1 if it is not. If value is System.String.Empty, the return value is startIndex.
Exceptions:
System.ArgumentNullException: value is null.
System.ArgumentOutOfRangeException: startIndex is negative. -or- startIndex specifies a position not within this instance.

Public Function **Insert**(ByVal *startIndex* As **Integer**, ByVal *value* As **String**) As **String**
Summary: Inserts a specified instance of System.String at a specified index position in this instance.
Parameters:
value: The System.String to insert.
startIndex: The index position of the insertion.
Return Values: A new System.String equivalent to this instance but with value inserted at position startIndex.

Public ReadOnly Property **Length**() As **Integer**
Summary: Gets the number of characters in this instance.
Return Values: The number of characters in this instance.

Public Function **PadLeft**(ByVal *totalWidth* As **Integer**, ByVal *paddingChar* As **Char**) As **String**
Summary: Right-aligns the characters in this instance, padding on the left with a specified Unicode character for a specified total length.
Parameters:
paddingChar: A Unicode padding character.

totalWidth: The number of characters in the resulting string, equal to the number of original characters plus any additional padding characters.

Return Values: A new System.String that is equivalent to this instance, but right-aligned and padded on the left with as many paddingChar characters as needed to create a length of totalWidth. Or, if totalWidth is less than the length of this instance, a new System.String that is identical to this instance.

Exceptions:

System.ArgumentOutOfRangeException: totalWidth is less than zero.

also…

Public Function **PadRight**(ByVal *totalWidth* As **Integer**, ByVal *paddingChar* As **Char**) As **String**

Public Function **ToLower**() As **String**

Summary: Returns a copy of this System.String converted to lowercase, using the casing rules of the current culture.

Return Values: A System.String in lowercase.

also…

Public Function **ToUpper**() As **String**

Public Function **Trim**() As **String**

Summary: Removes all leading and trailing white-space characters from the current System.String object.

Return Values: The string that remains after all white-space characters are removed from the start and end of the current System.String object.

Public Class **FileStream**

Member of **System.IO**

Summary: Exposes a System.IO.Stream around a file, supporting both synchronous and asynchronous read and write operations.

Public Overridable Sub **Close**()

Summary: Closes the current stream and releases any resources (such as sockets and file handles) associated with the current stream.

Public Sub **Dispose**()

Summary: Releases all resources used by the System.IO.Stream.

Public Sub **New**(ByVal *path* As **String**, ByVal *mode* As **System.IO.FileMode**, ByVal *access* As **System.IO.FileAccess**)

Summary: Initializes a new instance of the System.IO.FileStream class with the specified path, creation mode, and read/write permission.

Parameters:

mode: A System.IO.FileMode constant that determines how to open or create the file. (I.e. Append, Create, CreateNew, Open, OpenOrCreate)

access: A System.IO.FileAccess constant that determines how the file can be accessed by the FileStream object. (I.e. Read, ReadWrite, Write)

path: A relative or absolute path for the file that the current FileStream object will encapsulate.

Public Class **StreamReader**

Member of **System.IO**

Summary: Implements a System.IO.TextReader that reads characters from a byte stream in a particular encoding.

Public Overrides Sub **Close**()

Summary: Closes the System.IO.StreamReader object and the underlying stream, and releases any system resources associated with the reader.

Public Sub **Dispose**()

Summary: Releases all resources used by the System.IO.TextReader object.

Public ReadOnly Property **EndOfStream**() As **Boolean**

Summary: Gets a value that indicates whether the current stream position is at the end of the stream.
Return Values: true if the current stream position is at the end of the stream; otherwise false.

Public Sub **New**(ByVal *stream* As **System.IO.Stream**)
Summary: Initializes a new instance of the System.IO.StreamReader class for the specified stream.
Parameters: *stream*: The stream to be read.

Public Overrides Function **ReadLine**() As **String**
Summary: Reads a line of characters from the current stream and returns the data as a string.
Return Values: The next line from the input stream, or null if the end of the input stream is reached.

Public Overrides Function **ReadToEnd**() As **String**
Summary: Reads the stream from the current position to the end of the stream.
Return Values: The rest of the stream as a string, from the current position to the end. If the current position is at the end of the stream, returns the empty string("").

Public Class **StreamWriter**
Member of **System.IO**
Summary: Implements a System.IO.TextWriter for writing characters to a stream in a particular encoding.

Public Overrides Sub **Close**()
Summary: Closes the current StreamWriter object and the underlying stream

Public Sub **Dispose**()
Summary: Releases all resources used by the System.IO.TextWriter object.

Public Sub **New**(ByVal *stream* As **System.IO.Stream**)
Summary: Initializes a new instance of the System.IO.StreamWriter class for the specified stream, using UTF-8 encoding and the default buffer size.
Parameters: *stream*: The stream to write to.

Public Overrides Sub **Write**(ByVal *value* As **String**)
Summary: Writes a string to the stream.
Parameters: *value*: The string to write to the stream. If value is null, nothing is written.

Public Overridable Sub **WriteLine**(ByVal *value* As **String**)
Summary: Writes a string followed by a line terminator to the text stream.
Parameters: *value*: The string to write. If value is null, only the line termination characters are written.

Public Class **ListBox**
Member of **System.Windows.Forms**
Summary: Represents a Windows control to display a list of items.

Public Function **FindString**(ByVal *s* As **String**) As **Integer**
Summary: Finds the first item in the System.Windows.Forms.ListBox that starts with the specified string.
Parameters: *s*: The text to search for.
Return Values: The zero-based index of the first item found; returns ListBox.NoMatches if no match is found.

Public ReadOnly Property **Items**() As **System.Windows.Forms.ListBox.ObjectCollection**
Summary: Gets the items of the System.Windows.Forms.ListBox.
Return Values: An System.Windows.Forms.ListBox.ObjectCollection representing the items in the System.Windows.Forms.ListBox

---

Public Class **ObjectCollection**
Member of **System.Windows.Forms.ListBox**
Summary: Represents the collection of items in a System.Windows.Forms.ListBox.

Public Function **Add**(ByVal *item* As **Object**) As **Integer**
Summary: Adds an item to the list of items for a System.Windows.Forms.ListBox.
Parameters: *item*: An object representing the item to add to the collection.
Return Values: The zero-based index of the item in the collection.

Public Overridable Sub **Clear**()
Summary: Removes all items from the collection.

Public Function **Contains**(ByVal *value* As **Object**) As **Boolean**
Summary: Determines whether the specified item is located within the collection.
Parameters: *value*: An object representing the item to locate in the collection.
Return Values: true if the item is located within the collection; otherwise, false.

Public ReadOnly Property **Count**() As **Integer**
Summary: Gets the number of items in the collection.
Return Values: The number of items in the collection

Public Function **IndexOf**(ByVal *value* As **Object**) As **Integer**
Summary: Returns the index within the collection of the specified item.
Parameters: *value*: An object representing the item to locate in the collection.
Return Values: The zero-based index where the item is located within the collection; otherwise, negative one (-1).

Public Sub **Insert**(ByVal *index* As **Integer**, ByVal *item* As **Object**)
Summary: Inserts an item into the list box at the specified index.
Parameters:
item: An object representing the item to insert.
index: The zero-based index location where the item is inserted.

Default Public Overridable Property **Item**(ByVal *index* As **Integer**) As **Object**
Summary: Gets or sets the item at the specified index within the collection.
Parameters: *index*: The index of the item in the collection to get or set.
Return Values: An object representing the item located at the specified index within the collection.

Public Sub **Remove**(ByVal *value* As **Object**)
Summary: Removes the specified object from the collection.
Parameters: *value*: An object representing the item to remove from the collection.

Public Sub **RemoveAt**(ByVal *index* As **Integer**)
Summary: Removes the item at the specified index within the collection.
Parameters: *index*: The zero-based index of the item to remove.