



db2000 VB like functions

© 2008 Massimo Mascalchi

massimo.mascalchi@db2000web.net



DESCRIZIONE DELLE FUNZIONI

Exp

restituisce un valore double che specifica 'e' (la base dei logaritmi naturali) elevato alla potenza indicata

sintassi: `r = fVB.Exp([potenza])`

esempio: `r = fVB.Exp(10) ' 22026.465794857`

Fix

restituisce la parte intera del numero indicato

sintassi: `r = fVB.Fix([numero])`

esempio: `r = fVB.Fix(2.75) ' 2`

Hex

restituisce una stringa che rappresenta il valore esadecimale del numero indicato

sintassi: `r = fVB.Hex([numero])`

esempio: `r = fVB.Hex(255) ' "FF"`

InStr

restituisce la posizione della prima occorrenza della stringa indicata all'interno del testo indicato

sintassi: `r = fVB.InStr([testo], [stringa da trovare])`

esempio: `r = fVB.InStr("Ciao Mondo!", "Mondo") ' 6`

InStrRev

restituisce la posizione della prima occorrenza della stringa indicata all'interno del testo indicato iniziando la ricerca dalla fine di questo ultimo

sintassi: `r = fVB.InStrRev([testo], [stringa da trovare])`

esempio: `r = fVB.InStrRev("Mondo Ciao Mondo!", "Mondo") ' 12`

LCase

converte tutti i caratteri del testo indicato in minuscolo

sintassi: `r = fVB.LCase([testo])`

esempio: `r = fVB.LCase("CIAO MONDO!") ' "ciao mondo!"`

Left

restituisce una stringa a partire da sinistra del testo indicato che contiene il numero dei caratteri indicati

sintassi: `r = fVB.Left([testo], [numero caratteri])`

esempio: `r = fVB.Left("Ciao Mondo!", 4) ' "Ciao"`



Len

restituisce la lunghezza in caratteri del testo indicato

sintassi: `r = fVB.Len([testo])`

esempio: `r = fVB.Len("Ciao Mondo!") ' 11`

LSet

restituisce una stringa allineata a sinistra contenente il testo indicato adattato alla lunghezza in caratteri indicata

sintassi: `r = fVB.LSet([testo], [lunghezza in caratteri])`

esempi: `r = fVB.LSet("Ciao Mondo!", 4) ' "Ciao"`

`r = fVB.LSet("Ciao Mondo!", 20) ' "Ciao Mondo! "`

LTrim

restituisce una stringa contenente una copia del testo indicato ma privo degli eventuali spazi a sinistra

sintassi: `r = fVB.LTrim([testo])`

esempio: `r = fVB.LTrim(" Ciao Mondo! ") ' "Ciao Mondo! "`

Mid

restituisce una stringa estratta dal testo indicato dalla posizione e per il numeri di caratteri indicati

sintassi: `r = fVB.Mid([testo], [posizione di inizio estrazione], [numero caratteri da estrarre])`

esempio: `r = fVB.Mid("Ciao Mondo!", 6, 5) ' "Mondo"`

MkDir

crea la cartella indicata, restituisce **True** se la cartella è stata creata correttamente altrimenti **False**

sintassi: `r = fVB.MkDir([nome cartella])`

esempio: `r = fVB.MkDir("c:\db2000\tmp")`

New1

inizializza gli oggetti della libreria

sintassi: `fVB.New1`

esempio: `fVB.New1`



QBColor

restituisce un valore che rappresenta il codice in decimale del colore **RGB** corrispondente al numero di colore indicato

```
sintassi: r = fVB.QBColor([colore])
```

[colore] =	0	=	nero
	1	=	blu
	2	=	verde
	3	=	ciano
	4	=	rosso
	5	=	magenta
	6	=	giallo
	7	=	bianco
	8	=	grigio
	9	=	blu chiaro
	10	=	verde chiaro
	11	=	ciano chiaro
	12	=	rosso chiaro
	13	=	magenta chiaro
	14	=	giallo chiaro
	15	=	bianco brillante

```
esempio: r = fVB.QBColor(12) ' 255
```

Replace

restituisce il testo indicato aver operato la sostituzione della stringa indicata con la stringa di ricerca indicata

```
sintassi: r = fVB.Replace([testo], [stringa da ricercare], [stringa da sostituire])
```

```
esempio: r = fVB.Replace("Ciao Terra!", "Terra", "Mondo") ' "Ciao Mondo!"
```

Right

restituisce una stringa a partire da destra del testo indicato che contiene il numero dei caratteri indicati

```
sintassi: r = fVB.Right([testo], [numero caratteri])
```

```
esempio: r = fVB.Right("Ciao Mondo!", 6) ' "Mondo!"
```

Rmdir

rimuove la cartella indicata, restituisce **True** se la cartella viene eliminata correttamente altrimenti **False**

```
sintassi: r = fVB.Rmdir([nome cartella])
```

```
esempio: r = fVB.Rmdir("c:\db2000\tmp")
```

RSet

restituisce una stringa adattata alla lunghezza in caratteri indicata a partire dalla destra del testo indicato

```
sintassi: r = fVB.RSet([testo], [lunghezza in caratteri])
```

```
esempio: r = fVB.RSet("Ciao Mondo!", 20) ' "Ciao Mondo!"
```

RTrim

restituisce una stringa contenente una copia del testo indicato ma privo degli eventuali spazi a destra

```
sintassi: r = fVB.LTrim([testo])
```

```
esempio: r = fVB.LTrim("Ciao Mondo!") ' "Ciao Mondo!"
```



Space

restituisce una stringa contenente il numero di spazi indicati

sintassi: `r = fVB.Space([numero di spazi])`

esempio: `r = fVB.Space(10) ' " " "`

Split

restituisce una matrice a una dimensione con base zero contenente un numero di sottostringhe come specificato nei parametri estratte dall'espressione stringa indicata

sintassi: `ra() = fVB.Split([espressione, delimitatore, nr. elementi da estrarre, tipo di confronto])`
' se [nr. elementi da estrarre] = -1 nessun limite viene posto
' se [tipo di confronto] = True il confronto è di tipo testo altrimenti (False) è binario

esempio: `ra() = fVB.Split("1,2,3,4,5", ",", -1, True)`

SplitRegex

suddivide l'espressione stringa indicata in una matrice di sottostringhe con base zero in corrispondenza delle posizioni definite dalla corrispondenza dell'espressione regolare indicata

sintassi: `ra() = fVB.SplitRegex([espressione stringa, espressione regolare])`

esempio: `ra() = fVB.SplitRegex("1*-2*-3", "*-*")`

Str

restituisce una stringa contenente il valore dell'oggetto indicato

sintassi: `r = fVB.Str([valore oggetto])`

esempio: `r = fVB.Str(23.45) ' "23.45"`

StrComp

restituisce un numero che indica il risultato del confronto tra le due stringhe indicate

sintassi: `r = fVB.StrComp([stringa 1], [stringa 2])` ' -1 ([stringa 1] > [stringa 2])
' 0 ([stringa 1] = [stringa 2])
' 1 ([stringa 1] < [stringa 2])

esempi: `r = fVB.StrComp("Ciao Mondo!", "Ciao!") ' -1`
`r = fVB.StrComp("Ciao Mondo!", "Ciao Mondo!") ' 0`
`r = fVB.StrComp("Ciao!", "Ciao Mondo!") ' 1`

StrDup

restituisce una stringa che contiene la ripetizione del carattere indicato per il numero delle volte indicato (come in VB.NET, equivale alla funzione **String** del VB6)

sintassi: `r = fVB.StrDup([numero ripetizioni], [carattere])`

esempio: `r = fVB.StrDup(5, "A") ' "AAAAA"`

StringEx

restituisce una stringa che contiene la ripetizione del carattere indicato per il numero delle volte indicato (equivale alla funzione **String** del VB6)

sintassi: `r = fVB.StringEx([numero ripetizioni], [carattere])`

esempio: `r = fVB.StringEx(5, "A") ' "AAAAA"`



StrReverse

restituisce una stringa in cui l'ordine dei caratteri del testo indicato è stato invertito

sintassi: `r = fVB.StrReverse([testo])`

esempio: `r = fVB.StrReverse("Ciao Mondo!") ' "!odnoM oaiC"`

Trim

restituisce una stringa contenente una copia del testo indicato ma privo degli eventuali spazi sia destra che a sinistra

sintassi: `r = fVB.Trim([testo])`

esempio: `r = fVB.Trim(" Ciao Mondo! ") ' "Ciao Mondo!"`

UCase

converte tutti i caratteri del testo indicato in maiuscolo

sintassi: `r = fVB.UCase([testo])`

esempio: `r = fVB.UCase("ciao mondo!") ' "CIAO MONDO!"`

Val

restituisce il valore numerico rappresentato nella stringa indicata

sintassi: `r = fVB.Val([stringa])`

esempi: `r = fVB.Val("34.50") ' 34.5`

`r = fVB.Val("&H" & "0FF") ' 255`

