



Le Funzioni Split, Join, InStr e InStrRev

■ La Funzione Split

La funzione **Split** divide una stringa in un numero specificato di sottostringhe e usando un carattere separatore in essa inclusa restituisce tutte le sottostringhe di cui la stringa originale è composta in una matrice unidimensionale in base zero. Sintassi:

Split (expression, delimiter, limit, compare).

È necessario specificare solo l'argomento *expression* mentre tutti gli altri argomenti sono facoltativi. L'argomento *expression* è la stringa che verrà divisa in sotto stringhe e delimitata da un carattere contenuto all'interno. Per una stringa di lunghezza zero ("") la funzione restituisce un array vuoto senza elementi.

L'argomento *delimiter* è il carattere utilizzato per delimitare e separare le sottostringhe e identifica i limiti delle sottostringhe, e se viene omesso, verrà assunto il carattere spazio (" ") di default come delimitatore, mentre se è una stringa di lunghezza zero (""), la funzione restituisce l'intera espressione come una matrice a elemento singolo.

L'argomento *limit* specifica il numero di sottostringhe da restituire e il valore di -1 indica che tutte le sottostringhe vengano restituite.

L'argomento *compare* specifica il tipo di confronto da utilizzare per valutare le stringhe

È possibile specificare i seguenti argomenti per l'argomento *compare*:

- *vbUseCompareOption* (valore: -1) esegue un confronto utilizzando l'impostazione di Option Compare.
- *vbBinaryCompare* (valore: 0) esegue un confronto binario - confronti tra stringhe basato su un ordinamento
- *vbTextCompare* (valore: 1) esegue un confronto testuale - confronti tra stringhe che non si basano su un ordinamento testuale case-sensitive
- *vbDatabaseCompare* (valore: 2) esegue un confronto basato sui dati del database

Se non si specifica l'argomento *compare*, il confronto viene fatto sulla base Option Compare definita *Option Compare Statement*, cioè Option Compare Binary oppure Option Compare Text che può essere utilizzato per impostare il metodo di confronto che è necessario specificare a livello di modulo, prima di qualsiasi procedura e se l'Istruzione Option Compare non è specificata, il metodo di confronto testo predefinito è Binary.

Esempio: Contare e ritornare le parole all'interno di una stringa



Fig. 1



```
Sub Split2()  
Dim testo1 As Variant, varE As Variant, varD As Variant, varP As Variant  
Dim i As Integer  
'stringa che sarà suddivisa in sotto stringhe - ogni parola è separata da uno spazio  
varE = " Ho visto un re anche lui piangeva. "  
varD = " " 'Indicare lo spazio come delimitatore  
'con TRIM si rimuovono tutti gli spazi dal testo ad eccezione dei singoli spazi tra le parole  
varE = Application.Trim(varE)  
testo1 = Split(varE, varD)  
'Restituisce il numero di parole (7) nella stringa  
MsgBox UBound(testo1) + 1  
'mettere ogni parola della stringa su righe diverse  
For i = 0 To UBound(testo1)  
If i = 0 Then  
varP = testo1(i)  
Else  
varP = varP & vbCrLf & testo1(i)  
End If  
Next i  
MsgBox varP 'Restituisce ogni parola in una riga separata  
End Sub
```

Esempio: Estrarre le sotto stringhe utilizzando la funzione Split e riportare la lunghezza e il numero di occorrenze del carattere delimitatore all'interno di una stringa

```
Sub split1()  
Dim test As Variant, varE As Variant, varD As Variant  
Dim i As Integer, lungE As Long, lungEx As Long  
varE = "le belle vie del paese" 'stringa che sarà suddivisa in sotto stringhe  
varD = "e" 'delimitatore della stringa  
test = Split(varE, varD)  
'Restituisce il numero di elementi nella matrice  
MsgBox UBound(test) + 1  
'Restituisce 8, il n° di occorrenze del delimitatore all'interno della stringa  
MsgBox UBound(test)  
For i = LBound(test) To UBound(test)  
'riporta ogni elemento della matrice in cui la stringa è divisa  
MsgBox test(i)  
'ritorna la lunghezza di ogni elemento della matrice  
MsgBox Len(test(i))  
'riporta la lunghezza totale di tutti gli elementi della matrice  
lungE = lungE + Len(test(i))  
Next i  
lungEx = Len(varE) 'lunghezza della stringa divisa  
'calcolare la lunghezza di expression  
If lungEx = UBound(test) * Len(varD) + lungE Then  
MsgBox "Uguale"  
Else  
MsgBox "Diverso"  
End If  
End Sub
```



Esempio: Estrarre un elemento di un array, il nome del sito da un indirizzo web o il nome del file.

```
Sub split3()  
Dim testo1 As Variant, varE As Variant, varD As Variant  
Dim n As Integer  
varE = "E la luna bussò alle porte del sole" 'stringa da estrarre  
varD = " " 'Indicare lo spazio come delimitatore di stringhe  
testo1 = Split(varE, varD)  
n = 3 'Estrarre il terzo elemento della stringa precedente  
MsgBox testo1(n - 1)  
MsgBox Split("22,456,7,9824,0", ",")(n - 1)  
varE = http://forum.wintricks.it/showthread.php?t=155252 'indicare sito web  
varD = "/"  
testo1 = Split(varE, varD)  
n = 3  
MsgBox testo1(n - 1)  
varE = "C:\User\Alex\Documents\Excel\VBA\#39.xls" 'Specificare il percorso di un file  
varD = "\"  
testo1 = Split(varE, varD)  
n = UBound(testo1) + 1 'Estrarre l'ultimo elemento - il nome del file  
MsgBox testo1(n - 1)  
MsgBox testo1(UBound(testo1))  
End Sub
```

Esempio: Sostituire tutte le occorrenze di una stringa in un'espressione stringa con un'altra stringa

```
Function Rep_1(var As Variant, varF As Variant, varR As Variant, opt1 As Integer) As Variant  
Dim contaF As Integer, arr As Variant  
arr = Split(var, varF, , opt1)  
If UBound(arr) < 1 Then 'Se varF non è stato trovato in var l'array avrà un solo elemento  
Rep_1 = var 'Ritorno la stringa var ed esco dalla procedura  
Exit Function  
Else  
var = "" 'Inizio con una stringa di lunghezza zero  
For contaF = 1 To UBound(arr) 'Ciclo per il n° di occorrenze  
var = var & arr(contaF - 1) & varR 'Aggiunge ogni elemento (tranne l'ultimo) della matri. con varR  
Next contaF  
var = var & arr(UBound(arr)) 'Aggiungere l'ultimo elemento dell'array dopo tutte le sostituzioni  
End If  
Rep_1 = var 'Ritorno la stringa finale  
End Function  
Sub cambia1()  
Dim var As Variant, varF As Variant, varR As Variant, opt1 As Integer  
varR = "?" 'var è la stringa all'interno della quale varF viene cercato e sostituito da varR  
opt1 = 1 'valore per eseguire il confronto di testo  
If IsNull(var) Then 'Se var è Null, si esce  
MsgBox "var è nullo, esco dalla procedura"  
Exit Sub  
Else 'Se var non è Null  
If IsNull(varF) Or IsNull(varR) Or varF = "" Then  
MsgBox var  
Exit Sub  
Else  
MsgBox Rep_1(var, varF, varR, opt1)  
End If  
End If  
End Sub
```



■ La Funzione Join

La funzione **Join** unisce le sottostringhe contenute in una matrice, e restituisce una stringa con le sottostringhe separate da un carattere delimitatore. Sintassi:

Join (sourceArray, delimiter).

È necessario specificare l'argomento sourceArray mentre l'argomento delimiter è facoltativo, ricordare che *sourceArray* è un array che contiene le sottostringhe che devono essere unite per restituire una stringa e *delimiter* è il carattere stringa utilizzato per separare le sottostringhe, e se viene omissso, verrà assunto il carattere di spazio (" ") di default per essere usato come delimiter, se invece delimiter è una stringa di lunghezza zero (""), la funzione unisce le stringhe senza delimitatore.

Esempio: Utilizzo della funzione JOIN

```
Sub join1()  
Dim arr As Variant, varJ As Variant, varC As Variant  
Dim i As Integer  
'Definire l'array  
arr = Array("America", "Europa", "Africa", "Asia")  
'unire sottostringhe contenute in una matrice  
varJ = Join(arr, "&")  
'Ritorna la stringa unita  
MsgBox varJ  
'Concatenare ogni elemento della matrice  
For i = 0 To UBound(arr)  
varC = varC & "&" & arr(i)  
Next i  
'Rimuovere la "&" prima del primo elemento  
varC = Mid(varC, 2)  
'String ritorno dopo il concatenamento  
MsgBox varC  
End Sub
```

Esempio: Unire i valori delle celle in un intervallo del foglio di lavoro

	A	B	C	D	E
1	Nome	Città	Età	Sesso	Stato
2	Gino Primo	Milano	50	M	Celibe
3	Marco Secondo	Roma	42	M	Sposato
4	Teresa Terzo	Venezia	36	F	Sposata
5	Elena Quinto	Palermo	75	F	Celibe

Fig. 2



```
Sub join2()  
    Dim rng As Range, riga1 As Integer, colonna1 As Integer, i As Integer  
    Set rng = ActiveSheet.Range("A2:E4")  
    Dim varC As Variant  
    For riga1 = 1 To rng.Rows.Count  
        For colonna1 = 1 To rng.Columns.Count  
            If Not rng(riga1, colonna1).Value = vbNullString Then  
                varC = varC & "," & rng(riga1, colonna1).Value  
            End If  
        Next colonna1  
        'Se l'array è vuoto  
        If varC = vbNullString Then MsgBox "Array Vuoto": GoTo skip1  
        'restituisce un record per riga  
        MsgBox Mid(varC, 2)  
        varC = ""  
  
skip1:  
    Next riga1  
    'Dichiarare una matrice dinamica  
    Dim varA() As Variant  
    Icoll = rng.Columns.Count  
    i = 0  
    'Ridimensionare la matrice dinamica  
    ReDim varA(Icoll - 1) As Variant  
    For riga1 = 1 To rng.Rows.Count  
        For Icoll = 1 To rng.Columns.Count  
            If Not rng(riga1, Icoll).Value = vbNullString Then  
                'Per ogni vbNullString, diminuire il valore dell'indice di matrice  
                varA(Icoll - 1 - i) = rng(riga1, Icoll).Value  
            Else  
                'Contare il numero di vbNullString  
                i = i + 1  
                'Se l'array è vuoto  
                If i = rng.Columns.Count Then MsgBox "Array Vuoto": GoTo skip2  
            End If  
        Next Icoll  
        'Diminuire la dimensione della matrice per numero di vbNullString  
        ReDim Preserve varA(rng.Columns.Count - 1 - i) As Variant  
        'restituisce un record per riga  
        MsgBox Join(varA, ",")  
  
skip2:  
        'Diminuire la dimensione della matrice per numero di vbNullString  
        ReDim varA(rng.Columns.Count - 1) As Variant  
        i = 0  
    Next riga1  
End Sub
```



Esempio: Usare la funzione Split

```
Sub split2()  
Dim newT As Variant, varS As Variant, varD As Variant, varE As Variant, varJ As Variant  
Dim i As Integer  
varS = http://forum.wintricks.it/showthread.php?t=155252 'indirizzo web  
varD = "/" 'delimitatore  
newT = Split(varS, varD) 'Restituisce un array  
For i = 0 To UBound(newT) 'mettere ogni elemento della matrice su una riga separata  
If i = 0 Then  
varE = newT(i) 'Non inserire una interruzione di linea prima del primo elemento  
Else  
varE = varE & vbLf & newT(i)  
End If  
Next i  
MsgBox varE 'Restituisce ogni elemento su una riga separata  
varJ = Join(newT, varD) 'restituisce l'espressione stringa originale  
MsgBox varJ  
End Sub
```

Esempio: Utilizzo delle funzioni stringa

```
Sub demo1()  
Dim newT As Variant, varE As Variant, varSE As Variant, varD As Variant, varJ As Variant  
Dim Nfile As String, Fdir As String  
'stringa da cui si desidera estrarre un elemento  
varE = "Estrarre una sotto espressione dopo aver escluso un elemento da un'espressione"  
varD = " "  
'restituire una matrice unidimensionale in base zero  
newT = Split(varE, varD)  
'Escludere un elemento assegnare il n° dell'elemento (2) a una variabile  
n = 2  
For i = 0 To UBound(newT)  
If i = n - 1 Then  
varSE = varSE  
Else  
varSE = varSE & "," & newT(i)  
End If  
Next i  
'Rimuovere il primo ","  
varSE = Mid(varSE, 2)  
MsgBox varSE  
'Ridimensionare la matrice per ridurre gli elementi di 1 in modo da escludere l'ultimo elemento  
ReDim Preserve newT(UBound(newT) - 1)  
'Unire tutti gli elementi dell'array tranne l'ultimo e aggiungere il . alla fine  
varJ = Join(newT, varD) & "."  
MsgBox varJ 'Stringa estratta, escluso l'ultimo elemento  
varE = "C:\User\Alex\Documents\Excel\VBA\pippo.xls" 'indicare il percorso del file  
Nfile = Mid(varE, InStrRev(varE, "\") + 1) 'Estrarre il nome del file, dal percorso completo del file  
MsgBox Nfile  
Fdir = Left(varE, Len(varE) - Len(Nfile)) 'Estrarre il percorso della cartella, escluso il nome del file  
MsgBox Fdir  
End Sub
```




■ La Funzione InStr e InStrRev

La funzione **InStr** restituisce la posizione (numero di caratteri) in cui una stringa prima si verifica all'interno di un'altra stringa. Sintassi:

InStr (start, string, substring, compare)

ed è necessario specificare gli argomenti *string* e *substring*, mentre gli argomenti *start* e *compare* sono opzionali.

L'argomento *start* specifica la posizione (numero di caratteri) all'interno della stringa da cui si desidera iniziare la ricerca per *substring*, è necessario specificare l'argomento *start*, se l'argomento di confronto è da specificare e se viene omesso, per impostazione predefinita assumerà il valore 1 (cioè la ricerca partirà dalla prima posizione del carattere).

Specificando una posizione di partenza che è maggiore della lunghezza di string verrà restituito il valore 0 (zero), e se *start* contiene un valore *Null* si verificherà un errore. L'argomento *string* è l'espressione stringa all'interno della quale cercare *substring*, la funzione restituisce 0 se la stringa è di lunghezza zero, e restituisce *Null* se la stringa è *Null*. L'argomento *substring* è l'espressione stringa che viene cercata all'interno della stringa e la cui posizione verrà restituito dalla funzione che restituisce 0 se stringa non viene trovata, oppure restituisce il valore iniziale se la stringa è di lunghezza zero, o restituisce *Null* se la stringa è *Null*. L'argomento *compare* specifica il tipo di confronto da utilizzare per valutare le stringhe.

È possibile specificare i seguenti argomenti per l'argomento *compare*:

- *vbUseCompareOption* (valore: -1) esegue un confronto utilizzando l'impostazione di Option Compare.
- *vbBinaryCompare* (valore: 0) esegue un confronto binario
- *vbTextCompare* (valore: 1) esegue un confronto testuale - confronti tra stringhe che non si basano su un ordinamento testuale case-sensitive
- *vbDatabaseCompare* (valore: 2) esegue un confronto basato sui dati del database

Se non si specifica l'argomento *compare*, il confronto viene fatto sulla base Option definito nella dichiarazione, ricordare che l'istruzione Option Compare (cioè Option Compare Binary o Option Compare Text) può essere utilizzato per impostare il metodo di confronto ed è necessario specificare 'Option Compare Binary' o 'Option Compare Text' a livello di modulo, prima di qualsiasi altra procedura. Se l'Istruzione Option compare non è specificata, il metodo di confronto predefinito è Binary.

La **funzione InStrRev** Restituisce la posizione della prima occorrenza di una stringa inclusa in un'altra a partire dalla destra della stringa con la seguente sintassi:

InStrRev (string, substring, start, compare)

mentre si utilizzare la funzione *InStrRev* invece di *InStr* per cercare nella direzione opposta. È necessario specificare gli argomenti di stringa e sottostringa, mentre gli argomenti *start* e *compare* sono opzionali. Se viene omesso l'argomento *start*, viene utilizzato -1, che significa che la ricerca inizierà dalla posizione dell'ultimo carattere. Tutte le altre spiegazioni e sintassi rimangono invariati rispetto alla funzione *InStr*.



Esempio: Utilizzo della funzione InStr.

```
Sub InStrFunc()  
    Dim str1 As String, str2 As String  
    str1 = "Alice vince sempre"  
    str2 = "e"  
    MsgBox InStr(str1, str2) 'restituisce 5  
    str1 = "Alice vince sempre"  
    str2 = "e"  
    MsgBox InStr(4, str1, str2) 'restituisce 5  
    str1 = "Alice vince sempre"  
    str2 = "e"  
    MsgBox InStr(24, str1, str2) 'restituisce 0  
    str1 = ""  
    str2 = "e"  
    MsgBox InStr(str1, str2) 'restituisce 0  
    str1 = "Alice vince sempre"  
    str2 = "i"  
    MsgBox InStr(str1, str2) 'restituisce 3  
    Dim str3 As Variant  
    str1 = "Alice vince sempre"  
    str3 = Null  
    MsgBox VarType(InStr(str1, str3)) 'restituisce 1  
    str1 = "Alice vince sempre"  
    str2 = "s"  
    MsgBox InStr(2, str1, str2) 'restituisce 13  
    MsgBox InStr(2, str1, str2, 1) 'restituisce 13  
End Sub
```




Esempio: Sostituire tutte le occorrenze di una stringa in un'espressione stringa con un'altra stringa

```
Function cambia1(var As Variant, varF As Variant, varR As Variant) As Variant
Dim Lfind As Integer, Pfind As Integer
Pfind = InStr(var, varF) 'Posizione della prima occorrenza di varF, in var
Lfind = Len(varF) 'Lunghezza varF, che sarà sostituito con varR
Rplen = Len(varR) 'Lunghezza Rplen, che sarà sostituito da varF
'Se varF non viene trovato all'interno di var
If Pfind < 1 Then
'Si restituisce la stringa var stringa e si esce dalla procedura
cambia1 = var
Exit Function
'Se varF viene trovato all'interno di var
Else
Do
'Sostituire varF con varR in var
var = Left(var, Pfind - 1) & varR & Mid(var, Pfind + Lfind)
'Posizione della prima occorrenza di varF all'interno di var aggiornato, iniziando
' la ricerca dal primo carattere dopo l'ultima sostituzione
Pfind = InStr(Pfind + Rplen, var, varF)
'Se varF non è stato trovato all'interno aggiornato di var, si esce dal ciclo
If Pfind = 0 Then Exit Do
Loop
End If
cambia1 = var 'Ritorno stringa finale
End Function

Sub cambia2()
Dim var As Variant, varF As Variant, varR As Variant
'var è la stringa all'interno della quale varF viene cercato e sostituito da varR
var = "Alice vince sempre"
'varF è la stringa da cercare all'interno di var e che sarà sostituito da varR
varF = "e"
'varR è la stringa che sostituisce tutte le istanze di varF contenute in var
varR = "?"
'Se var è Null, si esce
If IsNull(var) Then
MsgBox "var è Null, esco dalla procedura"
Exit Sub
'Se var non è Null
Else
'Se uno tra varF o varR sono Null o varF è una stringa di lunghezza zero
If IsNull(varF) Or IsNull(varR) Or varF = "" Then
'Ritorno var senza sostituzioni e si esce dalla procedura
MsgBox var
Exit Sub
Else
'Se nessuno tra var, varF e varR sono Null, si esegue la funzione e si sostituiscono tutte le istanze
di varF
MsgBox cambia1(var, varF, varR)
End If
End If
End Sub
```