

# Manipuler les chaînes de caractères en VB6 et VBA Excel

par [SilkyRoad](#)

Date de publication : 23/11/2006

Dernière mise à jour :

Ce document présente quelques fonctions VB6 et/ou VBA Excel qui permettent de manipuler les chaînes de caractères. Tous les exemples ont été testés avec VB6-SP6 et Excel2002.

## I - Manipuler les chaînes de caractères

- I-A - Compter le nombre de caractères dans une chaîne
- I-B - Extraire une portion de caractères en partant de la gauche
- I-C - Extraire une portion de caractères en partant de la droite
- I-D - Extraire une chaîne de caractères à l'intérieur d'une autre chaîne
- I-E - Renvoyer la position de la première occurrence d'une chaîne dans une autre chaîne
- I-F - Inverser l'ordre des caractères dans une chaîne
- I-G - Passer l'ensemble d'une chaîne en majuscule
- I-H - Passer l'ensemble d'une chaîne en minuscule
- I-I - Scinder une chaîne
- I-J - Remplacer des caractères dans une chaîne
- I-K - Convertir une chaîne vers un format spécifié
- I-L - Supprimer les caractères non imprimables dans la feuille active
- I-M - Supprimer tous les espaces superflus à l'exception des espaces simples entre les mots
- I-N - Vérifier si un mot existe dans le dictionnaire Office
- I-O - La fonction Val
- I-P - Extraire toutes les valeurs numériques (entiers et décimales) contenues dans une chaîne.
- I-Q - Utiliser Option compare
  - I-Q-1 - Option compare Text
  - I-Q-2 - Option compare Binary
- I-R - Comparer deux chaînes en utilisant l'opérateur LIKE
- I-S - La fonction ASC
- I-T - La fonction CHR
- I-U - Découper une longue chaîne de caractères sans couper les mots
- I-V - Vérifier la présence d'une chaîne dans un tableau à une dimension
- I-W - Créer une jonction entre les chaînes contenues dans un tableau

## II - Remerciements

## III - Téléchargement

## I - Manipuler les chaînes de caractères

### I-A - Compter le nombre de caractères dans une chaîne

Utilisez la fonction **Len**

VB6-VBA

```
MsgBox Len("Developpez") 'renvoie 10
```

### I-B - Extraire une portion de caractères en partant de la gauche

Utilisez la fonction **Left**

VB6-VBA

```
'Renvoie les 3 premiers caractères d'un texte  
MsgBox Left("Developpez", 3) 'renvoie "Dev"
```

### I-C - Extraire une portion de caractères en partant de la droite

Utilisez la fonction **Right**

VB6-VBA

```
'Renvoie les 3 derniers caractères d'un texte  
MsgBox Right("Developpez", 3) ' renvoie "pez"
```

### I-D - Extraire une chaîne de caractères à l'intérieur d'une autre chaîne

Utilisez la fonction **Mid**

VB6-VBA

```
'5 est la position du caractère qui marque le début de la partie à extraire  
'et 2 correspond au nombre de caractères à renvoyer)  
MsgBox Mid("Vendredi", 5, 2) ' renvoie "re"
```

Si vous souhaitez récupérer toute la chaîne à partir du 5eme caractère, n'indiquez pas le dernier argument

VB6-VBA

```
MsgBox Mid("Vendredi", 5) 'Renvoie "redi"
```

## I-E - Renvoyer la position de la première occurrence d'une chaîne dans une autre chaîne

Utilisez la fonction **InStr**

Remarque:

cette fonction renvoie 0 si aucune occurrence n'est trouvée.

### VB6-VBA

```
'Renvoie la position de la première occurrence d'une chaîne ("e")  
MsgBox InStr("Developpez", "e") 'renvoie 2
```

les arguments de la fonction : **InStr(start, string1, string2, compare)**

Le premier argument "start" (facultatif) permet de définir la position de départ de la recherche.

### VB6-VBA

```
MsgBox InStr(1, "Developpez", "e") 'renvoie 2
```

un autre exemple

### VB6-VBA

```
Dim x As Integer  
x = InStr(1, "Developpez", "e") + 1  
MsgBox InStr(x, "Developpez", "e") 'renvoie 4
```

string1 contient la chaîne de caractères qui va être contrôlée.

String2 contient la chaîne à rechercher.

La recherche est sensible à la casse si le dernier argument "compare" (facultatif) est égal à 0, ou omis.

(Casse: Signifie que la recherche n'est pas sensible aux majuscules et aux minuscules)

### VB6-VBA

```
MsgBox InStr(1, "Developpez", "Z", 0) 'renvoie 0  
MsgBox InStr(1, "Developpez", "z", 0) 'renvoie 10
```

La recherche n'est pas sensible à la casse si le dernier argument "compare" est égal à 1

#### VB6-VBA

```
MsgBox Instr(1, "Developpez", "z", 1) 'renvoie 10  
MsgBox Instr(1, "Developpez", "Z", 1) 'renvoie 10
```

Nota:

La fonction Instr peut aussi servir à contrôler si le contenu d'une chaîne est égal à un des mots "cible" défini dans la procédure.

Cet exemple vérifie si le contenu de la cellule A1 est égal à un des 3 mots indiqués dans la variable "Cible".

#### VB6-VBA

```
Sub RechercheMultiple()  
    Dim Cible As String  
  
    Cible = "engrenage, reducteur, courroie"  
  
    If Instr(Cible, Range("A1")) = 0 Then  
        MsgBox "Non"  
    Else  
        MsgBox "Oui"  
    End If  
End Sub
```

## I-F - Inverser l'ordre des caractères dans une chaîne

Utilisez la fonction **StrReverse**

#### VB6-VBA

```
MsgBox StrReverse("Developpez")
```

#### VB6-VBA

```
'Contrôler si un mot est un palindrome( qui s'épelle de la même façon dans les deux sens)  
Dim leMot As String  
leMot = "radar"  
If leMot = StrReverse(leMot) Then MsgBox "Il s'agit d'un palindrome"
```

## I-G - Passer l'ensemble d'une chaîne en majuscule

Utilisez la fonction **UCase**

#### VB6-VBA

```
MsgBox UCase("Developpez")
```

## I-H - Passer l'ensemble d'une chaîne en minuscule

Utilisez la fonction **LCase**

### VB6-VBA

```
MsgBox LCase("Developpez")
```

## I-I - Scinder une chaîne

Utilisez la fonction **Split**.

Cette fonction est disponible à partir d'Excel2000

### VB6-VBA

```
'Extraire les données séparées par un espace dans une chaîne de caractères
Sub extractionMots()
    Dim Tableau() As String
    Dim i As Integer

    'découpe la chaîne en fonction des espaces " "
    'le résultat de la fonction Split est stocké dans un tableau
    Tableau = Split("Bienvenue sur le site Developpez.com", " ")

    'boucle sur le tableau pour visualiser le résultat
    For i = 0 To UBound(Tableau)
        'Le résultat s'affiche dans la fenêtre d'exécution de l'éditeur de macros
        Debug.Print Tableau(i)
    Next i
End Sub
```

## I-J - Remplacer des caractères dans une chaîne

Utilisez la fonction **Replace**

### VB6

```
'Remplace les points virgules par des points dans la variable monText
Sub remplacerCaracteres()
    Dim monText As String

    monText = Replace(monText, ";", ".")
End Sub
```

### VBA

```
'Remplace les points virgules par des points dans la plage de cellules sélectionnée
Sub remplacerCaracteres()
    Dim Cell As Variant

    For Each Cell In Selection
        Cell.Value = Replace(Cell.Value, ";", ".")
    Next Cell
```

VBA

```
End Sub
```

Un autre exemple

VBA

```
'Supprimer tous les espaces dans la colonne A  
Sheets("Feuill").Columns(1).Replace " ", ""
```

L'*instruction* **MID** permet aussi d'effectuer un remplacement dans une chaîne. La position de la chaîne à modifier doit être préalablement connue.

Merci à [Bbil](#) pour cette astuce.

VB6-VBA

```
Sub ModifChaine()  
    Dim St As String  
  
    St = "deceloppez.com"  
    Debug.Print St  
  
    'Oops... une erreur de frappe sur 3° caractère...  
    'Utilisons Mid pour le modifier  
    Mid(St, 3, 1) = "v"  
    Debug.Print St  
End Sub
```

## I-K - Convertir une chaîne vers un format spécifié

Utilisez la fonction **StrConv**

Les types de format:

vbUpperCase: 1 (Convertit la chaîne en majuscules)

vbLowerCase: 2 (Convertit la chaîne en minuscules)

vbProperCase: 3 (Convertit la première lettre de chaque mot de la chaîne en majuscule)

vbUnicode: 64 (Convertit la chaîne en Unicode à l'aide de la page de code par défaut du système)

vbFromUnicode: 128 (Convertit la chaîne Unicode dans la page de code par défaut du système)

VB6-VBA

```
'Transformer une chaîne en majuscules  
MsgBox StrConv("Developepez", vbUpperCase)
```

Un autre exemple issu de l'aide en ligne Excel:

#### VB6-VBA

```
'Convertit une chaîne Unicode en une chaîne ANSI.
Dim i As Long
Dim x() As Byte

x = StrConv("ABCDEFGF", vbFromUnicode)

For i = 0 To UBound(x)
    Debug.Print x(i)
Next
```

### I-L - Supprimer les caractères non imprimables dans la feuille active

#### VBA

```
Dim Cell As Range
For Each Cell In ActiveSheet.UsedRange
    Cell.Value = Application.WorksheetFunction.Clean(Cell.Value)
Next
```

Il n'y a pas d'équivalent en VB6

### I-M - Supprimer tous les espaces superflus à l'exception des espaces simples entre les mots

L'équivalent de la fonction SUPPRESSESPACE

#### VBA

```
Dim Cell As Range
For Each Cell In ActiveSheet.UsedRange
    Cell = Application.WorksheetFunction.Trim(Cell)
Next
```

#### VB6

```
Dim monText as String
monText = Trim(monText) '- suppression des espaces à gauche et à droite
monText = LTrim(monText) '- suppression des espaces à gauche
monText = RTrim(monText) '- suppression des espaces à droite
```

### I-N - Vérifier si un mot existe dans le dictionnaire Office

#### VBA

```
Sub controleDictionnaireOffice()
    Dim Valeur As Boolean

    Valeur = Application.CheckSpelling("ordinateur", , True)
    MsgBox Valeur
End Sub
```



Le dernier argument (False/True) indique si la vérification doit être sensible à la casse.

## I-O - La fonction Val

Renvoie le nombre contenu dans une chaîne de caractère

### VB6-VBA

```
Dim Chaine As String
Chaine = " 21ieme siècle"
MsgBox Val(Chaine) ' renvoie 21

Chaine = " siècle 21"
MsgBox Val(Chaine) ' renvoie 0

Chaine = "123,456"
MsgBox Val(Chaine) ' renvoie 123

Chaine = "123.456"
MsgBox Val(Chaine) ' renvoie 123.456
```

Remarques:

L'argument "Chaine" doit impérativement être de type String

La fonction Val ne reconnaît que le point comme séparateur décimal.

## I-P - Extraire toutes les valeurs numériques (entiers et décimales) contenues dans une chaîne.

### VB6-VBA

```
Sub extraireValeursNumeriques_DansChaine()
    Dim i As Byte, Nb As Byte
    Dim Cible As String, Resultat As String
    Dim Nombre As Double

    Cible = "12,3azerty23,5 67"
    'Pour que fonction Val puisse reconnaître les décimales: Remplacement des
    'virgules par des points
    Cible = Replace(Cible, ",", ".")
    'Pour gérer deux nombres qui se suivent: remplacement des espaces
    'par un caractère Alpha
    Cible = Replace(Cible, " ", "x")

    For i = 1 To Len(Cible)
        If IsNumeric(Mid(Cible, i, 1)) Then
            Nombre = Val(Mid(Cible, i, Len(Cible) - i + 1))
            Nb = Nb + 1
            Resultat = Resultat & Nombre & vbCrLf
            i = i + Len(Str(Nombre)) - 1
        End If
    Next

    MsgBox "Il y a " & Nb & " valeurs numériques dans la cellule " & vbCrLf & Resultat
End Sub
```

## I-Q - Utiliser Option compare

L'instruction Option Compare définit la méthode de comparaison de chaînes. Si le module ne contient pas d'instruction Option Compare, la méthode de comparaison de texte par défaut est Binary.

Option Compare doit être placé tout en haut du module, avant toute procédure.

### I-Q-1 - Option compare Text

Les comparaisons ne distinguent pas les majuscules des minuscules.

### I-Q-2 - Option compare Binary

Les comparaisons sont basées sur la représentation binaire interne des caractères.

#### VB6-VBA

```
Dim x As Boolean
x = "A" > "a"
MsgBox x 'renvoie Faux

x = "A" < "a"
MsgBox x 'renvoie Vrai

x = "a" < "b"
MsgBox x 'renvoie Vrai

x = "b" < "a"
MsgBox x 'renvoie Faux

'Exemples:
'A < B < E < Z < a < b < e < z < À < Ê < Ø < à < ê < ø
```

### I-R - Comparer deux chaînes en utilisant l'opérateur LIKE

#### VBA

```
'Vérifie si la chaîne DVP existe dans la cellule A1
If Range("A1").Value Like "*DVP*" Then MsgBox "Le mot "DVP" existe dans la cellule A1."
```

#### VB6

```
'Vérifie si la chaîne DVP existe dans la chaîne monText
If monText Like "*DVP*" Then MsgBox "Le mot "DVP" existe dans la chaîne monText."
```

La fonction LIKE permet aussi de contrôler le format d'une chaîne:

Cet exemple Vérifie que le premier caractère est une lettre majuscule suivie de trois caractères numériques (Styles corrects: B124, F456 ....), dans la cellule A1.

La procédure est sensible à la casse.

#### VBA

```
If Range("A1") Like "[A-Z]###" Then MsgBox "OK"
```

## VB6

```
If monText Like "[A-Z]###" Then MsgBox "OK"
```

## I-S - La fonction ASC

Permet de récupérer la valeur d'un caractère dans un jeu de caractères.

## VB6-VBA

```
MsgBox Asc("a") 'Renvoie 97
MsgBox Asc("A") 'renvoie 65
```

## I-T - La fonction CHR

Permet de récupérer un caractère en fonction de sa valeur dans un jeu de caractères.

## VB6-VBA

```
MsgBox Chr(97) 'renvoie a
MsgBox Chr(65) 'renvoie A
```

Un autre exemples pour manipuler des guillemets contenus dans une variable

## VB6-VBA

```
'Remplace le caractère guillemet " par un point d'exclamation dans une chaine
Dim strVariable As String
strVariable = Range("A1")
MsgBox Replace(strVariable, Chr(34), "!")
```

## I-U - Découper une longue chaine de caractères sans couper les mots

Exemple pour scinder une requête SQL dans une variable:

## VB6-VBA

```
'----- procedure pour scinder la requete -----
'La chaine sera scindée si sa longueur est supérieure à 80 caractères

Cible = Requete
Requete = ""

For i = 1 To Len(Cible)
'Recherche le premier espace après le 80ème caractère
X = InStr(80, Cible, " ")

If X = 0 Then
Chaine = Cible
Requete = Requete & "" & Chaine & ""
Exit For
End If

Chaine = Mid(Cible, 1, X)
Requete = Requete & "" & Chaine & " _" & vbCrLf
```

VB6-VBA

```

        Cible = Mid(Cible, Len(Chaine) + 1)
    Next
    Debug.Print Requete
    
```

I-V - Vérifier la présence d'une chaîne dans un tableau à une dimension

Utilisez la fonction **Filter**.

Cette fonction renvoie un tableau correspondant aux critères de filtre spécifiés.

**Filter(sourcesrray, match[, include[, compare]])**

Sourcesrray est le tableau de chaînes à une dimension dans lequel la recherche doit être effectuée.

Match est la chaîne à rechercher.

Include:

Attribuez la valeur True pour que la fonction renvoie les éléments qui contiennent la chaîne match spécifiée.

Attribuez la valeur False pour que la fonction renvoie les éléments qui ne contiennent pas la chaîne match spécifiée.

Compare indique le type de comparaison de chaîne à utiliser:

Constante	Valeur	Description
vbBinaryCompare	0	Les comparaisons distinguent les majuscules des minuscules.
vbTextCompare	1	Les comparaisons ne distinguent pas les majuscules des minuscules: <b>Valeur par défaut si l'argument n'est pas spécifié.</b>
vbDatabaseCompare	2	Microsoft Access seulement. Effectue une comparaison basée sur des informations contenues dans votre base de données.

## VB6-VBA

```

Dim DataString(3) As String
Dim InString() As String
Dim i As Integer

'Remplissage du tableau qui va être testé
DataString(0) = "chaîne 1"
DataString(1) = "string 2"
DataString(2) = "chaîne 3"
DataString(3) = "string 4"

'"DataString" est le tableau dans lequel doit être effectué la recherche.
'"str" est la chaîne à rechercher.
'La fonction Filter renvoie le tableau "InString" contenant les éléments
'qui répondent aux critères de la recherche.
InString = Filter(DataString, "str", True)

'Boucle sur le tableau afin de visualiser les éléments
'qui répondent aux critères de recherche.
For i = 0 To UBound(InString)
    Debug.Print InString(i)
Next i

```

## I-W - Créer une jonction entre les chaînes contenues dans un tableau

Utilisez la fonction **Join**.

**Join(sourcearray[, delimiter])**

Sourcearray est le tableau à une dimension contenant les chaînes à joindre.

Delimiter (Facultatif) spécifie le séparateur à utiliser entre les éléments du tableau, lors de la jonction. Les éléments sont séparés par un espace si l'argument Delimiter n'est pas spécifié.

## VB6-VBA

```

Dim Tableau(2) As String

Tableau(0) = "Chaîne01"
Tableau(1) = "Chaîne02"
Tableau(2) = "Chaîne03"

'Regroupe les chaînes du tableau, séparés par un point virgule ";"
MsgBox Join(Tableau, ";")

'Regroupe les chaînes du tableau sans séparateur
MsgBox Join(Tableau, " ")

'Les éléments sont séparés par un espace si l'argument Delimiter n'est
'pas spécifié.
MsgBox Join(Tableau)

'Regroupe les chaînes du tableau et ajout d'un saut de ligne
'entre chaque élément.
MsgBox Join(Tableau, vbCrLf)

```



## II - Remerciements

Merci à [Bbil](#) et [ThierryAIM](#) pour leur relecture, contribution et conseils avisés.

### III - Téléchargement