

Date de publication : 13/06/2004

Dernière mise à jour : 01/07/2006

La FAQ VBA fut initialement conçue par **Jean-Marc Rabilloud** et nous l'en remercions vivement.

Nous avons l'espoir qu'elle sera complétée par les contributions de tous les membres de l'équipe de rédaction et des utilisateurs du forum VBA..

Aidez-nous à faire vivre cette FAQ en participant au forum **Vos contributions VBA**.

Pour toutes questions ou tous problèmes concernant cette FAQ, vous pouvez contacter par MP **Khany** ou **ThierryAIM**.

Merci à tous pour votre collaboration passée et à venir.

L'équipe de rédaction de **Developpez.com**

Ont contribué à cette FAQ :

bidou - Etienne Bar - Maxence Hubiche - Romain
Puyfoulhoux - DarkVader - ThierryAIM -
SilkyRoad - bbil - nico-pyright(c) - Bazoom -

1. Généralités (2)	4
2. Toutes Applications Office (4)	5
3. Excel (101)	7
3.1. Application (15)	8
3.1.1. Configuration simple (6)	9
3.1.2. Divers (9)	11
3.2. Classeurs (10)	14
3.3. Feuilles (10)	19
3.4. Plages (25)	23
3.4.1. Généralités (2)	24
3.4.2. Manipuler des plages (13)	25
3.4.3. Mise en forme (3)	30
3.4.4. Valeurs et formules (7)	32
3.5. Graphiques (12)	35
3.6. Userforms, Code et Contrôles (6)	42
3.7. Manipuler WORD (23)	46
4. Word (26)	56
4.1. Application (1)	57
4.2. Document (24)	58
4.2.1. Sélection et Range (8)	61
4.2.2. Tableau et signets (6)	65

Sommaire > Généralités

Règles de notation

Auteurs : bidou ,

Dans cette FAQ, les syntaxes différentes dues à la version seront notées [97], [2000] etc....

Les objets mis en variable se noteront *objNomObjet*. Par exemple *objWorkSheet* pour une feuille Excel, *objDoc* pour un document Word, etc...

Les paramètres en VBA peuvent être indifféremment passés par leurs noms ou leur position ordinale. Vous trouverez dans cette FAQ les deux notations.

Le complément indispensable : la FAQ VBA Access

Auteurs : ThierryAIM ,

Vous trouverez aussi de nombreux exemples de code VBA, souvent adaptables dans les autres applications Office dans

[FAQ La FAQ VBA Access](#)

Sommaire > Toutes Applications Office

Comment créer un certificat local pour les macros VBA ?

Auteurs : **ThierryAIM** ,

Lorsque le niveau de sécurité intrinsèque des documents Office est réglé sur *moyen* ou *élevé*, il est alors impossible d'activer automatiquement les macros d'un document.

La première solution consiste à régler le niveau de sécurité sur *faible* (Menu Outils >> Macros >> Sécurité...), mais n'est pas très recommandée.

L'autre solution consiste à auto-certifier les macros d'un document.

Un certificat local, valable uniquement sur la machine sur laquelle il est installé, va vous permettre d'éviter le désagrément d'avoir à confirmer l'activation des macros à l'ouverture d'un document, ou pire, la désactivation systématique des macros.

Comment procéder :

- Rechercher dans le répertoire d'installation de Microsoft Office (généralement C:\Program Files\Microsoft Office \OfficeXX) le programme SelfCert.exe
- Exécuter SelfCert.exe
- Renseigner le nom du certificat dans la zone de texte et valider
- Fermer SelfCert.exe

- Ouvrir le document Office dont les macros doivent être certifiées
- Dans l'éditeur VBA (Alt-F11), menu Outils >> Signature électronique ...
- Cliquer sur "Choisir..." et sélectionner le certificat précédemment créé
- Enregistrer le document

A la prochaine ouverture du document, une alerte de sécurité va apparaître, cocher alors la case "Toujours faire confiance au macros provenant de...".

Pour supprimer un certificat existant :

- Panneau de configuration >> Options Internet
- Onglet Contenu, cliquer sur Certificats...
- Sélectionner le certificat et cliquer sur Supprimer

Comment Lire et modifier les propriétés des fichiers Office ?

Auteurs : **SilkyRoad** ,

Ce petit tutoriel pour vous montrer quelques exemples pour lire et modifier les propriétés des fichiers Office :

[Lire et modifier les propriétés d'un document Office](#)

Comment connaître les valeurs des twips en VBA ?

Auteurs : **nico-pyright(c)** ,

L'objet Screen n'existant pas en VBA, on peut retrouver ces valeurs en utilisant les fonctions de l'API Win32 :

vba

```
Private Declare Function GetDC Lib "user32" (ByVal hwnd As Long) As Long
Private Declare Function ReleaseDC Lib "user32" (ByVal hwnd As Long, ByVal hdc As Long) As Long
Private Declare Function GetDeviceCaps Lib "gdi32" (ByVal hdc As Long, ByVal nIndex As Long) As Long

Private Const HWND_DESKTOP As Long = 0
Private Const LOGPIXELSX As Long = 88
Private Const LOGPIXELSY As Long = 90

Function TwipsPerPixelX() As Single
Dim lngDC As Long
    lngDC = GetDC(HWND_DESKTOP)
    TwipsPerPixelX = 1440& / GetDeviceCaps(lngDC, LOGPIXELSX)
    ReleaseDC HWND_DESKTOP, lngDC
End Function

Function TwipsPerPixelY() As Single
Dim lngDC As Long
    lngDC = GetDC(HWND_DESKTOP)
    TwipsPerPixelY = 1440& / GetDeviceCaps(lngDC, LOGPIXELSY)
    ReleaseDC HWND_DESKTOP, lngDC
End Function
```

Comment vider le presse-papier en VBA ?

Auteurs : Romain Puyfoulhoux ,

L'objet Clipboard n'existant pas en VBA, il faut utiliser les fonctions de l'API Windows :

vba

```
'-- Déclaration des fonctions API
Private Declare Function OpenClipboard Lib "user32" (ByVal hwnd As Long) As Long
Private Declare Function CloseClipboard Lib "user32" () As Long
Private Declare Function EmptyClipboard Lib "user32" () As Long

Private Sub Commande0_Click()
    OpenClipboard 0
    EmptyClipboard
    CloseClipboard
End Sub
```


Sommaire > Excel > Application > Configuration simple

Comment supprimer la demande de mise à jour des liaisons ?

Auteurs : bidou ,

Il est possible d'utiliser :

```
Application.AskToUpdateLinks = True
```

Néanmoins cela ne supprime que la boîte de dialogue et les liaisons sont mises à jour. Pour empêcher la mise à jour des liaisons sans boîte de dialogue, il faut mettre le paramètre UpdateLinks de la méthode Open de l'objet Workbook à False

Comment contrôler le mode de calcul ?

Auteurs : bidou ,

Le mode de calcul d'Excel appartient à l'objet Application. Cependant l'ordre de calcul peut être réduit à tout ou partie des feuilles.

```
Application.Calculation = xlCalculationManual
```

Peut prendre une des valeurs

- xlCalculationAutomatic
- xlCalculationManual
- xlCalculationSemiautomatic

Le mode de calcul semi-automatique ne calcule pas les tables de données. Il peut être nécessaire de bloquer le calcul lorsqu'on travaille sur des cellules dépendantes pour éviter un recalcul à chaque opération.

Peut-on limiter les actions de l'utilisateur ?

Auteurs : bidou ,

L'application permet trois limitations différentes : Bloquer le glisser/déposer de cellule

```
Application.CellDragAndDrop = True
```

Désactiver les touches d'arrêt

```
Application.EnableCancelKey = xlDisabled
```

- xlDisabled : Interdit l'utilisation des touches d'arrêt
- xlErrorHandler : Envoie une erreur 18 vers le gestionnaire en cours
- xlInterrupt

Bloquer la souris et le clavier

```
Application.Interactive = False
```

Comment dévalider la copie ?

Auteurs : bidou ,

Lorsque l'on copie une plage de cellules, il y a apparition de la zone de copie. Celle ci est maintenue tant qu'une autre valeur n'est pas mise dans le presse-papier. Il est possible de vider celui-ci en faisant

```
Application.CutCopyMode = False
```

Comment supprimer les messages d'alertes ?

Auteurs : bidou ,

Les messages d'avertissements peuvent être supprimés en faisant

```
Application.DisplayAlerts = False
```

N'oubliez pas de rétablir la propriété avant la fin de votre code. Attention : Il ne s'agit pas de messages d'erreurs, mais de messages proposant un choix à l'utilisateur. Désactiver ces messages revient à toujours exécuter la même action que si l'utilisateur avait cliqué sur OUI.

Pourquoi désactiver la mise à jour de l'écran ?

Auteurs : bidou ,

Désactiver la mise à jour n'est pas toujours pertinent. La désactivation se fait par

```
Application.ScreenUpdating = False
```

Celle-ci n'est rentable que si votre code utilise de nombreux appels à la méthode Select. Dans ce cas le gain de temps est énorme.

Sommaire > Excel > Application > Divers

Comment programme-t-on la correction d'orthographe ?

Auteurs : bidou ,

Cela dépend de ce que l'on veut obtenir. Pour démarrer la correction sur une feuille par exemple il suffit de faire :

```
Application.CheckSpelling CustomDictionary:="PERSO.DIC", IgnoreUppercase:=False, AlwaysSuggest:=True
```

Ceci a pour effet d'ouvrir la fenêtre du correcteur. Il est possible de contrôler un seul mot, avec une réponse booléenne.

```
Private Function MotExiste(ByVal strMot As String) As Boolean
    MotExiste = Application.CheckSpelling(strMot, "PERSO.DIC", False)
End Function
```

Peut-on basculer le type de référence par le code ?

Auteurs : bidou ,

Oui. Il faut faire

```
Application.ReferenceStyle = xlR1C1
```

ou xlA1 dans l'autre sens. Néanmoins, il vaut mieux l'éviter en cours d'exécution. Généralement, Excel convertit correctement ses références. Cependant certains codes attendent un type de référence homogène entre le code et la fonction appelée.

A quoi sert la propriété WorksheetFunction ?

Auteurs : bidou ,

Elle renvoie la collection des fonctions intégrées d'Excel. Ceci est très pratique dans de nombreux cas. Par exemple, dans Excel 97, la fonction VBA Replace n'existe pas. Mais Excel fournit sa fonction de feuille SUBSTITUTE qui est équivalente. On peut l'utiliser par le biais de WorksheetFunction.

```
Dim MaChaine As String
MaChaine = "où est l'apostrophe"
MaChaine = Application.WorksheetFunction.Substitute(MaChaine, "'", "'")
```

A quoi correspond la notation [A1] ?

Auteurs : bidou ,

C'est un appel implicite à la méthode Evaluate. Il s'agit en fait d'un interpréteur (le même que celui qui interprète les formules Excel). On peut faire des utilisations très complexes de cette méthode. Par exemple

```
MaValeur = Application.Evaluate("LOG(A1*3+2) ")
```

Cela peut s'écrire aussi

```
MaValeur = [LOG(A1*3+2)]
```

Notez qu'il n'y a pas de guillemets lorsqu'on utilise la notation crochets. L'interprétation des plages, dans notre cas A1, demande que le système de référence soit homogène. Si ma feuille était en référence LC j'obtiendrais une incompatibilité de type.

Quelles différences entre FindFile et GetOpenFilename ?

Auteurs : bidou ,

Principalement FindFile ouvre le fichier alors que GetOpenFilename renvoie le nom du fichier sélectionné. Cependant GetOpenFilename permet d'utiliser un filtrage ce que ne permet pas la méthode FindFile. Par exemple :

```
NomFichier =  
Application.GetOpenFilename("Classeurs Excel(*.xls),*.xls, Macros complémentaires (*.xla),*.xla")
```

Il y a une plus grande sécurité à utiliser la méthode GetOpenFilename.

Comment gérer les événements de l'application ?

Auteurs : bidou ,

Il y a deux approches différentes dans ce concept. L'objet Application donne la possibilité de créer du code sur des méthodes *OnAction*. Par exemple

```
Application.OnKey "^{p}", "Gestion"
```

Déclenche la procédure Gestion sur l'appui des touches CTRL+P Il est aussi possible de faire de l'interruption d'événements sur l'objet Application, mais c'est un peu plus complexe. Dans un module objet (Feuille, classeur, classe), on déclare une variable pour l'application. Dès lors, il est possible de gérer les événements par l'intermédiaire de cette variable.

```
Dim WithEvents App As Application  
  
Private Sub App_NewWorkbook(ByVal Wb As Excel.Workbook)  
  
End Sub
```

Peut-on appeler une macro d'un autre classeur ?

Auteurs : bidou ,

Il faut utiliser Run. Cette méthode permet aussi de passer des paramètres. Supposons que mon classeur "calcul.xls" contienne une fonction Log10 telle que :

```
Public Function Log10(ByVal X As Single) As Single  
    Log10 = Log(X) / Log(10#)  
End Function
```

Il est possible de l'utiliser depuis un autre classeur en faisant

```
Result = Application.Run("calcul.xls!Log10", 1.2)
```

Comment exporter un graphe en un gif ?

Auteurs : **DarkVader**,

La routine ci-dessous te permettra de sauvegarder le graphe actif

```
Sub PrintChart()  
Dim FName As String, NomGraphe As String  
  
NomGraphe = Right(ActiveChart.Name, Len(ActiveChart.Name) - Len(ActiveSheet.Name) - 1)  
With ActiveSheet.ChartObjects(NomGraphe).Chart  
    FName =  
    Application.GetSaveAsFilename("", "Fichier Gif (*.GIF),*.GIF,Fichier JPEG (*.JPG),*.JPG,Tous fichiers (*.*)*,*.*)",  
    .Export Filename:=FName, FilterName:=TypeImg, Interactive:=True  
End With  
End Sub
```

Comment trouver la traduction anglaise d'une fonction de feuille de calcul ?

Auteurs : **bbil**,

Dans le répertoire d'installation d'Excel, (généralement C:\Program Files\Microsoft Office\Officex\1036 pour une version Office en français), le fichier VBALIST.XLS permet de trouver, entre autres, la traduction des fonctions de feuilles de calcul Excel.

Exemple : pour la fonction EQUIV => MATCH

Ces fonctions peuvent ensuite être utilisées dans le code VBA :

```
Sub FindFirst()  
    myVar = Application.WorksheetFunction _  
        .Match(9, Worksheets(1).Range("A1:A10"), 0)  
    MsgBox myVar  
End Sub
```

lien : [FAQ](#) A quoi sert la propriété WorksheetFunction ?

lien : [FAQ](#) Pourquoi autant de propriétés "Formula" différentes ?

Sommaire > Excel > Classeurs

Comment ajouter un classeur contenant juste une feuille ?

Auteurs : bidou ,

Il y a deux méthodes. On peut restreindre le nombre de feuilles de calcul créées dans un nouveau classeur

```
Application.SheetsInNewWorkbook = 1  
Application.Workbooks.Add
```

Seulement cela est un paramétrage de l'application et il faudrait restaurer l'ancienne valeur. On peut aussi créer un classeur contenant une feuille de type spécifique

```
Application.Workbooks.Add xlWBATWorksheet
```

contient une feuille de calcul

```
Application.Workbooks.Add xlWBATChart
```

Contient une feuille graphique

Comment ouvrir un classeur ?

Auteurs : bidou ,

On utilise la méthode `Open` de la collection `WorkBooks` Elle accepte plusieurs paramètres dont quelques-uns sont utiles à connaître, `FileName` est le seul paramètre obligatoire. Il doit contenir le chemin complet

```
Application.Workbooks.Open "d:\tuto.xls"
```

On peut aussi passer par la méthode `GetOpenFilename` pour laisser l'utilisateur sélectionner le fichier

```
Application.Workbooks.Open Application.GetOpenFilename()
```

Les paramètres intéressants supplémentaires sont : `UpdateLinks`

- 0--> Ne met à jour aucune référence
- 1--> Met à jour des références externes mais pas des références distantes
- 2--> Met à jour des références distantes mais pas des références externes
- 3--> Met à jour des références distantes et des références externes

Une référence externe appartient à un autre classeur, une référence distante appartient à un autre programme. `ReadOnly` Booléen qui définit si le classeur doit être ouvert en lecture seule `Password` Représente le mot de passe pour un classeur à ouverture protégée

Comment travailler sur deux classeurs ?

Auteurs : bidou ,

Dans ce cas on utilise des variables

```
Dim objWorkbookSource As Workbook, objWorkbookCible As Workbook
```

```
set objWorkbookSource = Application.Workbooks.Open(Application.GetOpenFilename)
set objWorkbookCible = Application.Workbooks.Add
```

Il ne reste plus qu'à basculer entre les deux variables

Comment fermer un classeur sans enregistrer les changements ?

Auteurs : bidou ,

Il suffit de faire

```
ActiveWorkbook.Close False
```

La méthode Close permet de gérer un grand nombre de cas. Sa syntaxe générale est :

```
Close(SaveChanges, FileName, RouteWorkbook)
```

Si SaveChanges est vrai et que le classeur n'a pas encore été enregistré, il utilisera la valeur de filename. Si celle-ci n'est pas précisée, il y aura appel de la boîte de dialogue SaveAs.

Je ne comprends pas la protection du classeur !

Auteurs : bidou ,

Dans Excel, la protection d'un objet ne joue que sur les descendants directs. Je m'explique : la protection d'un classeur ne protège que ses fenêtres et sa structure de feuille. Si je fais :

```
ActiveWorkbook.Protect Password:="monpasse", Structure:=True, Windows:=False
```

Je ne protège que la structure. Cela veut dire qu'il n'est pas possible d'ajouter, de supprimer ou de déplacer des feuilles. En aucun cas cela ne protège les cellules. Pour cela il faut protéger la ou les feuilles.

Comment dupliquer une plage sur plusieurs feuilles du classeur ?

Auteurs : bidou ,

On utilise la méthode FillAcrossSheet de l'objet Workbook

```
ThisWorkbook.Worksheets.FillAcrossSheets Range("A1:C10"), xlFillWithAll
```

Notez que le deuxième paramètre permet de préciser si on veut dupliquer le contenu, le format ou les deux.

Quelle différence entre ActiveWorkbook et ThisWorkbook ?

Auteurs : bidou , Etienne Bar ,

ThisWorkbook représente le classeur qui contient le code faisant appel à **thisWorkbook**. **Active Workbook** représente le classeur actif de l'application. De manière générale, il vaut mieux éviter les objets actifs et privilégier des variables, car la programmation des objets actifs est assez piègeuse : ceux-ci tendent à ne plus l'être (actif) quand on en a besoin. Prenons un exemple.

```
Set objSheet = ThisWorkbook.Sheets("recup")
Application.Workbooks.Open "d:\user\tuto1.xls"
ActiveWorkbook.OpenLinks Name:=ActiveWorkbook.LinkSources(xlExcelLinks)
```

Après l'appel de OpenLinks plus de possibilité de savoir aisément quel classeur est désigné par ActiveWorkbooks

Comment écrit-on une fonction de feuille de calcul ?

Auteurs : bidou ,

Il s'agit d'une fonction normale. Par exemple

```
Function InvChaine(Cellule As Range) As Variant

Dim MaChaine As String, cpt As Long

If Len(Cellule.Value) = 0 Then
    InvChaine = CVErr(xlErrValue)
Else
    MaChaine = CStr(Cellule.Value)
    For cpt = Len(MaChaine) To 1 Step -1
        InvChaine = InvChaine & Mid(MaChaine, cpt, 1)
    Next cpt
End If

End Function
```

Si dans votre feuille vous écrivez la formule =invchaine(E2) La cellule contiendra la chaîne retournée ou #VALEUR

Peut-on supprimer les liaisons d'un classeur ?

Auteurs : bidou ,

Ce n'est pas toujours évident. Pour supprimer les liaisons il y a globalement deux méthodes. Une recherche complète permettant la récupération de ce qui est récupérable

```
Sub ChercheLiaison()
Dim NomFichier As String, MonClasseur As Workbook, Liaisons As Variant
Dim compteur As Long, comptCar As Long, Cible As Range
Dim FirstAddress As String, PlageLieu As Range, comptFeuille As Long, Reponse As Integer
Dim MaFeuille As Worksheet, MonGraphe As Chart, MonGraphe1 As ChartObject, MaSerie As Series

NomFichier = Application.GetOpenFilename
Workbooks.Open NomFichier, False
Set MonClasseur = ActiveWorkbook
Liaisons = MonClasseur.LinkSources
If IsEmpty(Liaisons) Then Exit Sub
'parcours les feuilles
For Each MaFeuille In MonClasseur.Worksheets
    MaFeuille.Activate
    MaFeuille.Cells.Select
    For compteur = 1 To UBound(Liaisons)
        For comptCar = Len(Liaisons(compteur)) To 1 Step -1
            If Mid(Liaisons(compteur), comptCar, 1) = "\" Then
                Liaisons(compteur) = Mid(Liaisons(compteur), comptCar + 1)
            Exit For
        End If
    Next comptCar
    Set Cible = Selection.Find(What:=Liaisons(compteur), After:=ActiveCell,
LookIn:=xlFormulas, _
```

```

LookAt:=xlPart, SearchOrder:=xlByRows, SearchDirection:=xlNext, _
MatchCase:=False)
If Not Cible Is Nothing Then
    FirstAddress = Cible.Address
    Do
        If PlageLieu Is Nothing Then Set PlageLieu = Cible Else Set PlageLieu =
Union(PlageLieu, Cible)
        Set Cible = Selection.FindNext(After:=Cible)
        Loop While Not Cible Is Nothing And Cible.Address <> FirstAddress
    End If
Next compteur
If Not PlageLieu Is Nothing Then
    Reponse = MsgBox("La feuille " & MaFeuille.Name & " contient " & PlageLieu.Cells.Count &
-
                                " cellules avec des liaisons" & vbCrLf & _
                                "voulez-
vous les supprimer ?", vbYesNo + vbQuestion, "Liaisons trouvées")
    If Reponse = 6 Then
        For Each Cible In PlageLieu.Cells
            Cible.Formula = Cible.Value
        Next
    End If
    Set PlageLieu = Nothing
End If
For Each MonGraphe1 In MaFeuille.ChartObjects
    For Each MaSerie In MonGraphe1.SeriesCollection
        For compteur = 1 To UBound(Liaisons)
            If InStr(1, MaSerie.Formula, Liaisons(compteur), vbTextCompare) > 0 Then
                Reponse = MsgBox("le graphe de la feuille " & MonGraphe1.Name & _
                    " contient une série " &
MaSerie.Name & " avec des liaisons" & vbCrLf & _
                                "Voulez-
vous les supprimer ?", vbYesNo + vbQuestion, "Liaisons trouvées")
                If Reponse = 6 Then
                    MaSerie.Delete
                    Exit For
                End If
            End If
        Next compteur
    Next
Next
For Each MonGraphe In MonClasseur.Charts
    For Each MaSerie In MonGraphe.SeriesCollection
        For compteur = 1 To UBound(Liaisons)
            If InStr(1, MaSerie.Formula, Liaisons(compteur), vbTextCompare) > 0 Then
                Reponse = MsgBox("le graphe de la feuille " & MonGraphe.Name & _
                    " contient une série " &
MaSerie.Name & " avec des liaisons" & vbCrLf & _
                                "voulez-
vous les supprimer ?", vbYesNo + vbQuestion, "Liaisons trouvées")
                If Reponse = 6 Then
                    MaSerie.Delete
                    Exit For
                End If
            End If
        Next compteur
    Next
Next
End Sub

```

Une méthode beaucoup plus violente qui consiste à modifier les liaisons par le nom du classeur source. Cela va créer des références circulaires qu'il faudra alors supprimer. Les données anciennement liées seront perdues.

```

Dim Nom As String, objWorkbook As Workbook, TabLiaison As Variant, cmpt As Long
Dim objWorksheet As Worksheet, objRange As Range

```

```
Set objWorkbook = Application.Workbooks.Open(Application.GetOpenFilename, UpdateLinks:=0)
TabLiaison = objWorkbook.LinkSources(xlExcelLinks)
If IsEmpty(TabLiaison) Then Exit Sub
For cmpt = LBound(TabLiaison) To UBound(TabLiaison)
    objWorkbook.ChangeLink TabLiaison(cmpt), objWorkbook.FullName, xlLinkTypeExcelLinks
Next cmpt
For Each objWorksheet In objWorkbook.Worksheets
    Do
        Set objRange = objWorksheet.CircularReference
        If Not objRange Is Nothing Then objRange.Value = objRange.Value
    Loop While Not objRange Is Nothing
```

Comment empêcher la fermeture de mon classeur ?

Auteurs : bidou ,

Dans le module ThisWorkbook on met

```
Private Sub Workbook_BeforeClose(Cancel As Boolean)
    Cancel=true
End Sub
```

Comme à un moment il faudra fermer le classeur on désactivera la gestion des événements

```
Application.EnableEvents = False
ThisWorkbook.Close True
```

Sommaire > Excel > Feuilles

Quelles différences entre Sheet et Worksheet ?

Auteurs : bidou ,

Attention, il n'existe pas d'objet Sheet. Un classeur renvoie deux collections, Sheets et Worksheets. La collection Sheets représente l'ensemble des feuilles d'un classeur quelque soit leurs types. La collection Worksheets représente l'ensemble des feuilles de calcul d'un classeur. De fait, on utilise la collection Sheets que lorsqu'on doit manipuler un classeur ayant des feuilles mixtes (graphiques et calculs), généralement pour ajouter une feuille graphique vierge.

```
ActiveWorkbook.Sheets.Add After:=ActiveWorkbook.Sheets.Item(ActiveWorkbook.Sheets.Count),  
Type:=xlChart
```

Je n'obtiens pas les propriétés et méthodes de l'objet ActiveSheet !

Auteurs : bidou ,

En effet, sur Excel 97, certains objets ne donnent pas accès aux fonctionnalités Intellisense. Pour contourner le problème, utiliser une variable

```
Dim objFeuille As Worksheet  
  
Set objFeuille = ActiveWorkbook.ActiveSheet
```

A partir de là, objFeuille vous fournira ses propriétés / méthodes

Comment ajouter un numéro de page à l'impression ?

Auteurs : bidou ,

La gestion de la mise en page de votre feuille pour l'impression passe par l'intermédiaire de la propriété PageSetup. Celle-ci renvoie un objet PageSetup qu'il faut manipuler. Par exemple

```
With objFeuille.PageSetup  
    .CenterFooter = "&P"  
    .CenterHeader = "&F"  
    .FirstPageNumber = 3  
    .FitToPagesWide = 1  
    .Orientation = xlLandscape  
    .PrintGridlines = False  
    .PrintHeadings = False  
End With  
objFeuille.PrintOut 1, 1, 1, False
```

Ce code imprime la feuille en mettant le nom du fichier dans l'en-tête, le numéro de page dans le pied, celui-ci commençant à 3, force l'impression sur une page en largeur en mode paysage. Ni les lignes, ni les numéros de lignes/colonnes ne seront imprimés.

Comment protéger ma feuille ?

Auteurs : bidou ,

La méthode Protect permet de gérer plusieurs protections selon les paramètres. Elle est de la forme

```
Protect(Password, DrawingObjects, Contents, Scenarios, UserInterfaceOnly)
```

- **DrawingObjects** : protège les contrôles et les formes dessinées de la feuille
- **Contents** : Protège le contenu (graphique ou cellules)
- **Scenarios** : Protège les cellules variables du scénario
- **UserInterfaceOnly** : Protège les UserForm sans protéger le code

Tout ces paramètres sont facultatifs. C'est pour cela qu'on trouve souvent des codes ressemblant à :

```
ActiveSheet.Protect Password:="MonPasse", Contents:=True
```

A quoi sert l'événement Calculate de la feuille ?

Auteurs : bidou ,

Je serais tenté de répondre "à rien". Celui-ci intervient après le recalcul, ce qui fait qu'il a peu d'utilisations concrètes. Néanmoins il permet de déclencher une opération nécessitant d'attendre la fin du recalcul

Comment utiliser le paramètre Target des événements de feuilles ?

Auteurs : bidou ,

Target représente la plage des cellules affectées par l'événement. A ce titre, c'est un objet Range. Pour savoir si cette plage contient un élément d'une plage particulière, on teste l'intersection des deux plages.

```
Private Sub Worksheet_SelectionChange(ByVal Target As Excel.Range)
    If Not Application.Intersect("Target", Range("A1")) Is Nothing Then
        .....
    End If
End Sub
```

Comment ajouter un CommandButton à une feuille ?

Auteurs : bidou ,

On passe par la méthode Add de la collection Shapes

```
Dim objWorksheet As Worksheet
Set objWorksheet = ThisWorkbook.Worksheets(1)
objWorksheet.Shapes.AddOLEObject "Forms.CommandButton.1", , , , , 141, 28.5, 153, 27.75
```

Le bouton sera créé ici en (141;28,5) et d'une taille de 153x27,75

Comment ajouter une commande au menu contextuel d'une cellule Excel ?

Auteurs : bbil ,

Pour ajouter une commande au menu contextuel (apparaissant sur click droit) d'une cellule, il suffit d'ajouter une entrée à la *CommandeBar* "Cell" (menu contextuel cellule feuille de calcul).

vba

```

'
' Rajout d'une entrée dans menu contextuel
'
Function MenuCell(stCde As String, stMess As String)
Dim mc As CommandBarControls
Dim bo As CommandBarButton
    Set mc = CommandBars("Cell").Controls
    Set bo = mc.Add(msoControlButton, Temporary:=True)
    bo.Caption = stMess
    bo.OnAction = stCde
End Function

```

Exemple d'utilisation :

Ajout d'une entrée Ajouter Prefixe, permettant, sur click droit sur une cellule, de remplacer la valeur de chaque cellule sélectionnée par la valeur courante précédée d'une chaîne de caractère saisie dans une inputBox.

vba

```

'Initialisation dans l'évènement Workbook_open
'
Private Sub Workbook_Open()
    MenuCell "Ajouter Prefixe", "AjoutePrefixe"
End Sub

```

vba

```

' Fonction exemple..
'
Sub AjoutePrefixe()
Dim Pre As String
Dim s as range, c As Object
    Pre = InputBox("prefixe")
    Set s = Selection
    For Each c In s
        c.Value = Pre & c.Value
    Next
End Sub

```

Comment insérer une image sans donner sa taille et dans une cellule définie ?

Auteurs : bidou ,

C'est un peu complexe. Pour insérer une image on utilise habituellement :

```

Dim objFeuille As Worksheet, objShape As Shape

Set objFeuille = ActiveSheet
Set objShape = objFeuille.Shapes.AddPicture("c:\developpez\logo_developpez.gif", msoFalse,
    msoCTrue, 100, 100, 70, 70)

```

Cette méthode oblige de passer les valeurs de hauteur et de largeur (dans notre cas 70,70). Cependant, pour faire ce que vous voulez, il faut passer par la manipulation des objets Pictures

```

Dim objFeuille As Worksheet, objPict As Picture

Set objFeuille = ActiveSheet
Set objPict = objFeuille.Pictures.Insert("C:\developpez\logo_developpez.gif")
With objPict

```

```
.Left = Range("G2").Left  
.Top = Range("G2").Top  
End With
```

Comment appliquer une couleur à l'onglet d'une feuille (à partir de Excel 2002) ?

Auteurs : SilkyRoad ,

Cette option n'est disponible qu'à partir de la version Excel 2002.

En mode feuille de calcul : Click droit sur l'onglet >> Couleur d'onglet...

Par le code VBA :

vba

```
'-- Appliquer une couleur jaune à l'onglet de la Feuille  
Sheets("Feuille").Tab.ColorIndex = 6
```

Sommaire > Excel > Plages

Sommaire > Excel > Plages > Généralités

Que représentent les objets Range et Cells ?

Auteurs : bidou ,

Toute la difficulté de la programmation Excel se retrouve dans ces concepts. Il n'existe pas d'objet Cell. Cells donne accès à une collection de cellules représentant les cellules de tout objet contenant une ou plusieurs cellules. L'objet Range représente un ensemble de cellules. Il peut en contenir une ou plusieurs, contiguës ou non. De ce fait un objet Range peut contenir plusieurs objets Range. De nombreuses propriétés / méthodes de l'objet Range renvoient un Objet Range. Un objet Range contient toujours une collection Cells. La plupart des Q/R de cette FAQ portent sur la manipulation des objets Range

Comment fonctionnent les références Excel ?

Auteurs : bidou ,

Excel gère deux systèmes de références. La notation A1 est telle que les lignes sont représentées par des nombres croissants, du haut vers le bas, et les colonnes par des lettres croissantes, de gauche à droite. La notation L1C1 est telle que les lignes sont des nombres précédés de 'L' et les colonnes des nombres précédés de 'C'. Utiliser l'un ou l'autre de ces systèmes ne changent rien. Il faut juste utiliser le même que celui dans lequel Excel est. C'est pour cela que généralement on force le mode d'Excel par le code. Quelle que soit le type de référence, Excel gère aussi une notation absolue et une notation relative. Celle-ci peut être différente pour les lignes et les colonnes.

Sommaire > Excel > Plages > Manipuler des plages

Comment décaler une plage ?

Auteurs : bidou ,

On utilise généralement la méthode `Offset`. Ainsi

```
Range("A1:C1").Offset(1, 1)
```

renvoie la plage (Range) "B2:D2". Dans certains cas particuliers, on peut utiliser les méthodes `Next` ou `Previous` pour décaler vers des cellules adjacentes. Par exemple

```
MsgBox Range("A1").Next.Address(True, True, xlA1, False)
```

Renvoie `B1`

Comment redimensionner une plage ?

Auteurs : bidou ,

On utilise la méthode `Resize` pour redimensionner une plage quand on connaît le nombre de lignes et de colonnes de la nouvelle plage.

```
MsgBox Range("A1").Resize(3, 2).Address(True, True, xlA1)
```

Renvoie `A1:B3`. Notons que comme dans le cas de `Offset` l'objet renvoyé est un Range. Il est donc possible de combiner plusieurs de ces ordres.

```
Range("A1").Resize(3, 2).Offset(,1).ClearContents
```

Que représente l'objet Area ?

Auteurs : bidou ,

L'objet `Area` n'existe pas. Il existe une collection `Areas` qui renvoie l'ensemble des plages continues d'un objet Range. En lui-même, tout objet Range possède au moins un élément dans sa collection `Areas`, qui peut être lui-même si Range est une plage continue. En générale on énumère la collection.

```
Dim objRange As Range

For Each objRange In Range("A3:A12,D10:E21,I1:I16").Areas
    objRange.Merge False
Next
```

Quelles différences entre `UsedRange` et `CurrentRegion` ?

Auteurs : bidou ,

`UsedRange` est la plage des cellules utilisées dans une feuille. `CurrentRegion` est la plage rectangulaire de cellules non vides qui entourent l'objet Range sur lequel on fait l'appel de `CurrentRegion`. `CurrentRegion` doit être utilisé avec

précaution car il a parfois un comportement piègeux. Imaginons qu'il existe un tableau rempli en A1:C3 et que toutes les autres cellules soit vides, alors

```
MsgBox Cells(2, 4).CurrentRegion.Address(True, True, xlR1C1)
```

Renvoie R1C1:R3C4

Comment sélectionner les cellules d'une colonne ?

Auteurs : bidou ,

Plusieurs cas : Sélectionner toute la colonne

```
Range("A1").EntireColumn.Select
```

Sélectionner les cellules non vides de la colonne

```
Range("A1").EntireColumn.SpecialCells(xlCellTypeConstants).Select
```

Attention, si vous voulez compter dans les cellules non vides, celles qui contiennent une formule il faut

```
Application.Union(Range("A1").EntireColumn.SpecialCells(xlCellTypeConstants),  
Range("A1").EntireColumn.SpecialCells(xlCellTypeFormulas)).Select
```

Sélectionner une plage de valeurs continues

```
Range("A1", Range("A1").End(xlDown)).Select
```

Sélectionner de la première à la dernière cellule non vide en prenant les cellules vides incluses

```
Range("A1", Range("A1").EntireColumn.Find(What:="*", SearchDirection:=xlPrevious)).Select
```

Peut-on récupérer une plage de cellules contenant les cellules en erreurs ?

Auteurs : bidou ,

C'est assez simple :

```
Dim objWorksheet As Worksheet, objRange As Range  
Set objWorksheet = ThisWorkbook.Worksheets(1)  
Set objRange = objWorksheet.UsedRange.SpecialCells(xlCellTypeFormulas, xlErrors)
```

C'est assez logique puisque seules les cellules contenant une formules peuvent être en erreur.

Comment trier une plage ?

Auteurs : bidou ,

En supposant que je veuille trier la plage J1:L20 par ordre croissant dans la colonne J puis dans la colonne L

```
Range("J1:L20").Sort Range("J1"), xlAscending, Range("L1"), , xlAscending
```

Comment savoir si une plage fait référence à une cellule ou à plusieurs, voire à plusieurs pages ?

Auteurs : Etienne Bar , bidou ,

Il faut utiliser la propriété Count des collections visées. C'est la collection Cells pour le nombres de cellules, Areas pour le nombre de plages. Un code Exemple pourrait être.

```
Dim Msg As String, objRange As Range

With ThisWorkbook.Worksheets(1).Range("A3:A11,D7:E14,G1:G4,I15:J23,C22:F22")
    Msg = "La plage contient " & .Cells.Count & " cellules dans " &
    .Areas.Count & " plages continues" & vbCrLf
    For Each objRange In .Areas
        Msg = Msg & "La plage " & objRange.AddressLocal(False, False, xlA1) & " contient " &
        objRange.Cells.Count & " cellules" & vbCrLf
    Next
End With
MsgBox Msg
```

Comment récupérer la plage filtrée par AutoFilter ?

Auteurs : bidou ,

Excel masque les lignes ne répondant pas au filtre, il suffit donc de récupérer la plage des cellules visibles.

```
Range("A2:C21").AutoFilter Field:=1, Criteria:=">10",
Range("A2:C21").SpecialCells(xlCellTypeVisible).Copy Destination:=Range("A24")
```

Pourquoi ai-je une erreur lors d'une recherche si la valeur n'existe pas ?

Auteurs : bidou ,

Le problème ne vient pas de la recherche mais de l'appel d'une propriété / méthode sur l'objet renvoyé. Je prends un exemple standard produit par l'enregistreur de macro.

```
Range("A2:C21").Select
Selection.Find(What:="170", After:=ActiveCell, LookIn:=xlValues, LookAt _
:=xlPart, SearchOrder:=xlByRows, SearchDirection:=xlNext, MatchCase:= _
False).Activate
```

Ce code fonctionnera sans problème pour peu que la valeur cherchée existe dans la plage. En effet la méthode Find renvoie un objet Range désignant la cellule contenant la valeur ou Nothing si celle-ci n'existe pas. Dans le cas de ce code elle va donc tenter de faire Nothing.Activate d'où l'erreur. Pour utiliser correctement la recherche, utilisez le type de code suivant :

```
Dim objCell As Range, PremAdresse As String, PlageResult As Range

With Range("A2:C21")
    Set objCell = .Find(What:="170", After:=ActiveCell, LookIn:=xlValues, LookAt:=xlPart,
SearchOrder:=xlByRows, SearchDirection:=xlNext)
    If Not objCell Is Nothing Then
        PremAdresse = objCell.Address
    Do
```

```
    If PlageResult Is Nothing Then
        Set PlageResult = objCell
    Else
        Set PlageResult = Application.Union(objCell, PlageResult)
    End If
    Set objCell = .FindNext(objCell)
Loop While Not objCell Is Nothing And objCell.Address <> PremAdresse
End If
End With
```

L'objet `plageResult` contiendra toutes les cellules contenant la valeur, ou `nothing` si cette valeur n'existe pas.

Comment effacer les valeurs d'un tableau sans effacer les formules et les titres ?

Auteurs : bidou ,

On travaille sur la méthode `SpecialCells`

```
Range("A1:F21").SpecialCells(xlCellTypeConstants, xlNumbers).ClearContents
```

Je veux retrouver toutes les cellules ayant un fond particulier, mais le code est très long, que faire ?

Auteurs : bidou ,

comme je l'ai dit dans une autre réponse, il faut éviter les longues énumérations. Supposons que je veuille trouver toutes les cellules ayant un fond bleu.

```
For Each objCell In ThisWorkbook.Worksheets(1).Cells
    If objCell.Interior.ColorIndex = 5 Then
        If objRange Is Nothing Then
            Set objRange = objCell
        Else
            Set objRange = Application.Union(objCell, objRange)
        End If
    End If
End If
Next
```

Ce code est facile à écrire mais très long à exécuter. Pour accélérer la méthode je vais utiliser le regroupement des propriétés. Dans Excel, lorsqu'une propriété est équivalente pour toutes les cellules d'une plage, elle est renvoyée comme propriété de la plage, sinon c'est `Null` qui est renvoyé. Dès lors je vais parcourir les colonnes et ne garder que celles contenant au moins une cellule ciblée. Puis je ferai le parcours des lignes de la même façon. Par intersection, il sera facile de composer ma plage de retour.

```
Dim objRange As Range, PlageRed As Range, objCell As Range, PlageResult As Range

'reduction de la plage
For Each objRange In ThisWorkbook.Worksheets(1).Columns
    If IsNull(objRange.Interior.ColorIndex) Then
        If PlageRed Is Nothing Then
            Set PlageRed = objRange
        Else
            Set PlageRed = Application.Union(objRange, PlageRed)
        End If
    End If
End If
Next
'travail en ligne
For Each objRange In ThisWorkbook.Worksheets(1).Rows
    If IsNull(objRange.Interior.ColorIndex) Then
```

```
For Each objCell In Application.Intersect(objRange, PlageRed).Cells
    If objCell.Interior.ColorIndex = 5 Then
        If PlageResult Is Nothing Then
            Set PlageResult = objCell
        Else
            Set PlageResult = Application.Union(objCell, PlageResult)
        End If
    End If
Next
PlageResult.Select
```

Comment demander à l'utilisateur de sélectionner une plage de cellule ?

Auteurs : **Bazoom** ,

Vous connaissez sûrement la *fonction* `InputBox`, mais connaissez-vous la *méthode* `InputBox` de l'objet `Application` ? Pour tous les détails sur cette méthode, reportez-vous à l'aide en ligne de VBA Excel, mais voici un exemple pour générer une boîte de dialogue qui attend une cellule ou un groupe de cellules :

vba

```
Dim P As Range
On Error Resume Next
Set P = Application.InputBox("Sélectionnez une cellule ou une plage :", Type:=8)
On Error GoTo 0
If P Is Nothing Then MsgBox "Sélection annulée"
```

Pour information, les autres types disponibles pour la méthode `InputBox` :

- 0 : Formule
- 1 : Nombre
- 2 : String
- 4 : Booléen
- 8 : Range
- 16 : Erreur
- 64 : Tableau

Sommaire > Excel > Plages > Mise en forme

Comment gérer les encadrements par le code ?

Auteurs : bidou ,

Il existe deux approches. On veut encadrer le contour externe de la plage, on utilise `BordersAround`.

```
objRange.BorderAround xlContinuous, xlMedium, xlColorIndexAutomatic
```

on souhaite mettre des bordures particulières, on passe par l'élément de la collection `Borders`

```
Dim objWorksheet As Worksheet, objRange As Range

Set objWorksheet = ThisWorkbook.ActiveSheet
Set objRange = objWorksheet.Range(objWorksheet.Cells(1, 1), objWorksheet.Cells(10, 10))
With objRange.Borders
    With .Item(xlInsideHorizontal)
        .ColorIndex = 5
        .LineStyle = xlContinuous
        .Weight = xlThin
    End With
End With
```

On veut tout encadrer

```
With objRange.Borders
    .ColorIndex = 5
    .LineStyle = xlContinuous
    .Weight = xlThin
End With
```

Comment utilise-t-on le format conditionnel ?

Auteurs : bidou ,

Le code suivant va mettre un fond vert à toutes les cellules dont la valeur est doublonnée dans la première colonne.

```
Dim ObjRange As Range

Set ObjRange = Range("A1").EntireColumn
With ObjRange.FormatConditions
    .Add Type:=xlExpression, Formula1:="=NB.SI(C1;LC)>1"
    .Item(.Count).Interior.Color = vbGreen
End With
```

Comment ajouter un lien hypertext ?

Auteurs : bidou ,

On peut ajouter un lien indifféremment à un objet ou à une plage.

```
Dim objLink As Hyperlink

Set objLink = ActiveSheet.Hyperlinks.Add(Range("F2"), "d:\user\tuto.xls")
With objLink
    .SubAddress = "A1:C10"
    .Follow NewWindow:=True
End With
```

End With

Ce code crée le lien est ouvre une nouvelle fenêtre contenant la cible. Astuce : Lorsqu'on modifie la cible d'un lien existant on ne modifie pas le texte du lien. Pour que celui-ci change, il faut modifier la propriété value de la cellule contenant le lien :

```
objLink.Address = "d:\jmarc.xls"  
objLink.Range.Value = "d:\jmarc.xls"
```

Sommaire > Excel > Plages > Valeurs et formules

Peut on cacher les valeurs nulles sans écrire de formules ?

Auteurs : bidou ,

Il faut utiliser les formats de cellules.

```
objSheet.UsedRange.NumberFormat = "0;-0;;@"
```

Pourquoi passer par un tableau pour manipuler des plages de valeurs ?

Auteurs : bidou ,

Ce n'est pas toujours une nécessité, mais cela permet d'accélérer le code. Sur les grandes collections, il convient d'éviter l'énumération qui est très lente. Comparez la rapidité d'exécution du code entre

```
Dim ObjCell As Range

For Each ObjCell In Range("A1:J65535").Cells
    ObjCell.Value = ObjCell.Value * 2 + 3
Next
```

et

```
Dim Montab As Variant, cmpt1 As Long, cmpt2 As Long
Montab = Range("A1:J65535").Value
For cmpt1 = LBound(Montab, 1) To UBound(Montab, 1)
    For cmpt2 = LBound(Montab, 2) To UBound(Montab, 2)
        Montab(cmpt1, cmpt2) = Montab(cmpt1, cmpt2) * 2 + 3
    Next cmpt2
Next cmpt1
Range("A1:J65535").Value = Montab
```

Le deuxième code s'exécute environ 20 fois plus vite.

Pourquoi autant de propriétés "Formula" différentes ?

Auteurs : bidou , Etienne Bar ,

Regardons ensemble. Un objet Range possède des propriétés Formula, FormulaArray, FormulaLocal, FormulaR1C1, FormulaLocalR1C1 La propriété FormulaArray sert à entrer des formules matricielles. Par exemple

```
Range("E1:F11").FormulaArray = "=LINEST(R1C3:R20C3,R1C2:R20C2,TRUE,TRUE)"
```

Ce code renvoie la matrice de réponses de la formule DROITEREG d'Excel. Notez que ce n'est pas DROITEREG mais LINEST qui est utilisé. Pourquoi ? Intrinsèquement, Excel utilise les formules internationales (donc anglaise). Mais il n'est pas toujours évident de connaître la syntaxe internationale. On peut donc toujours utiliser la syntaxe locale en appelant la propriété locale équivalente. Ainsi les deux formules suivantes sont équivalentes

```
Range("E18").Formula = "=OFFSET(C1,MATCH(16,A1:A20,0)-1,0)"
Range("E18").FormulaLocal = "=DECALER(C1;EQUIV(16;A1:A20;0)-1;0)"
```

Je peux aussi entrer la formule avec une référence LC (RC international) et j'utilise alors la version R1C1 des propriétés

```
Range("E18").FormulaR1C1 = "=OFFSET(R1C3,MATCH(16,R1C1:R20C1,0)-1,0)"  
Range("E18").FormulaR1C1Local = "=DECALER(L1C3;EQUIV(16;L1C1:L20C1;0)-1;0)"
```

Comment fonctionnent les recopies de cellules ?

Auteurs : bidou ,

Il existe globalement deux types de recopies. La recopie incrémentée (AutoFill) et la non-incrémentée. Une recopie incrémentée consiste à recopier une plage en incrémentant une des propriétés suivantes

- **xlFillDefault** : laisse excel choisir l'incrément
- **xlFillSeries** : Recopie avec les paramètres de la série si ceux-ci existent
- **xlFillCopy** : Sans incrément (utilisez plutôt la méthode Fill)
- **xlFillFormats** : Recopie le format sans les valeurs
- **xlFillValues** : Utilise l'incrément par défaut, mais ne recopie pas les formules
- **xlFillDays** : Incrémente les jours
- **xlFillWeekdays** : Incrémente les noms de jours
- **xlFillMonths** : Incrémente les mois
- **xlFillYears** : Incrémente les années
- **xlLinearTrend** : Tendence linéaire
- **xlGrowthTrend** : Tendence géométrique

Evidemment la recopie doit pouvoir trouver la valeur de l'incrément, il faut donc que la source contiennent au moins deux cellules.

```
Range("F29:F30").AutoFill Destination:=Range("F29:F43"), Type:=xlGrowthTrend  
Cells(29, 6).Resize(2).AutoFill Destination:=Cells(29, 6).Resize(10), Type:=xlFillDefault
```

La recopie non incrémentée utilise une des méthodes Fill directionnelles. FillDown, FillUp, FillLeft, FillRight. Il faut que la cellule opposée au sens contienne une valeur ou une formule.

```
Cells(29, 6).Resize(10).FillDown
```

Peut on effacer les formules pour ne garder que les valeurs ?

Auteurs : bidou ,

Généralement lorsqu'on utilise Excel on utilise un collage spécial pour cela. Mais par le code il vaut mieux passer par une astuce toute simple.

```
Range("F1:H20").Value = Range("F1:H20").Value
```

Pourquoi cela fonctionne ? Car l'affectation explicite d'une valeur à la propriété value revient à supprimer la valeur de la propriété Formula. En effet on ne peut pas affecter une valeur à une cellule contenant une formule sans écraser celle-ci afin d'éviter un conflit entre la valeur calculée et la valeur affectée.

Comment limiter les saisies par le code ?

Auteurs : bidou ,

On utilise les validations comme par le menu. L'exemple suivant restreint les valeurs saisissables de G1 entre 10 et 100

```
With Range("G1").Validation
    .Add Type:=xlValidateWholeNumber, AlertStyle:=xlValidAlertStop, _
    Operator:=xlBetween, Formula1:="10", Formula2:="100"
    .IgnoreBlank = True
    .InCellDropdown = True
    .ErrorMessage = "La valeur doit être comprise entre 10 et 100"
    .ShowInput = True
    .ShowError = True
End With
```

Comment obtenir la valeur en lettre de la colonne ?

Auteurs : bidou ,

La formule générique est

```
NumCol = Cells(1, 72).Column
MsgBox IIf(NumCol > 26, Chr(64 + NumCol \ 26) & Chr(64 + NumCol Mod 26), Chr(64 + NumCol))
```

Sommaire > Excel > Graphiques

Quelles sont les méthodes pour créer un graphique ?

Auteurs : bidou ,

Il en existe vraiment plusieurs, je ne vais vous donner ici que les principales. Par interprétation de la sélection. Ce type de création est peu souple et souvent source d'erreur.

```
Range("A1").CurrentRegion.Select  
ThisWorkbook.Charts.Add
```

Le graphique créera des abscisses par défaut. Cela veut dire aussi qu'une plage sélectionnée avant la création d'un graphique sera tracée à la création de celui-ci. Avec SourceData. Si votre tableau est continu avec les abscisses à gauche, c'est la méthode la plus rapide

```
Dim objChart As Chart, objRange As Range  
  
Set objRange = Worksheets("Feuil1").Range(Worksheets("Feuil1").Cells(1, 1),  
Worksheets("Feuil1").Cells(21, 3))  
Set objChart = ThisWorkbook.Charts.Add  
objChart.ChartType = xlXYScatter  
objChart.SetSourceData objRange, xlColumns
```

Il existe une autre syntaxe équivalente

```
Dim objChart As Chart, objRange As Range  
  
Set objRange = Worksheets("Feuil1").Range(Worksheets("Feuil1").Cells(1, 1),  
Worksheets("Feuil1").Cells(21, 3))  
Set objChart = ThisWorkbook.Charts.Add  
objChart.ChartType = xlXYScatter  
objChart.SeriesCollection.Add objRange, xlColumns, True, True
```

Par manipulation de séries

```
Dim objChart As Chart, objRange As Range, MaSerie As Series, compteur As Long  
  
Set objRange = Worksheets("Feuil1").Range(Worksheets("Feuil1").Cells(1, 1),  
Worksheets("Feuil1").Cells(21, 3))  
Set objChart = ThisWorkbook.Charts.Add  
objChart.ChartType = xlXYScatter  
objChart.SeriesCollection.Add objRange, xlColumns, True, True  
For compteur = 2 To objRange.Columns.Count  
Set MaSerie = objChart.SeriesCollection.NewSeries  
MaSerie.Values = "=" & objRange.Columns(compteur).Address(True, True, xlR1C1, True)  
MaSerie.XValues = "=" & objRange.Columns(1).Address(True, True, xlR1C1, True)  
Next compteur
```

Comment créer un graphique composé ?

Auteurs : bidou ,

Il faut d'abord créer le graphique puis manipuler les séries

```
Public Sub CreationGraphe1()  
  
Dim MonGraphe As Chart, MaPlage As Range
```

```

Set MaPlage = Worksheets("donnees").Range(Cells(2, 7), Cells(14, 12))
Set MonGraphe = ThisWorkbook.Charts.Add
MonGraphe.ChartType = xlColumnStacked100
MonGraphe.SetSourceData MaPlage, xlColumns
With MonGraphe.SeriesCollection(5)
    .ChartType = xlXYScatterSmoothNoMarkers
    .AxisGroup = 2
    With .Border
        .Weight = xlMedium
        .LineStyle = xlAutomatic
        .ColorIndex = 4
    End With
End With
With MonGraphe
    .HasTitle = True
    With .ChartTitle
        .Characters.Text = "ANNEE 2001"
        .Shadow = True
        .Border.Weight = xlHairline
    End With
    With .Axes(xlValue, xlPrimary)
        .HasTitle = True
        .AxisTitle.Characters.Text = "Proportion"
    End With
    With .Axes(xlValue, xlSecondary)
        .HasTitle = True
        .AxisTitle.Characters.Text = "Total (hrs)"
    End With
End With
End Sub

```

Comment ajouter du texte sur un graphique ?

Auteurs : DarkVader ,

Il est possible de manipuler des objets Shapes sur une feuille graphique

```

ActiveChart.Shapes.AddTextbox(msoTextOrientationHorizontal, 100, 100, 100#, 100#).Select
Selection.Characters.Text = "Le Texte"

```

Comment lister tous les graphiques d'un classeur ?

Auteurs : SilkyRoad ,

Cette procédure permet de boucler sur tous les onglets du classeur et de lister les feuilles graphiques (Charts) ainsi que les graphiques incorporés (ChartObjects).

vba

```

Private Sub CommandButton1_Click()
    Dim j As Byte, i As Byte
    Dim Ch As Chart

    'Recherche les onglets graphiques
    For Each Ch In Charts
        Debug.Print "Feuille graphique " & vbTab & Ch.Name
    Next Ch

    'Boucle sur tous les onglets du classeur
    For j = 1 To Sheets.Count

```

vba

```
'Recherche les graphiques incorporés dans la feuille
For i = 1 To Sheets(j).ChartObjects.Count
    Debug.Print "Nom: " & Sheets(j).ChartObjects(i).Name & vbTab & _
        " dans " & Sheets(j).Name
Next i
Next j
End Sub
```

Comment vérifier si un graphique existe dans une feuille ?

Auteurs : **SilkyRoad** ,

Cet exemple vérifie si un graphique nommé "Graphique 1" existe dans la Feuil2

vba

```
Dim Grph As ChartObject
On Error Resume Next
Set Grph = Sheets("Feuil2").ChartObjects("Graphique 1")
If Not Grph Is Nothing Then MsgBox "Le graphique existe"
```

Comment créer un graphique à partir de tableaux dynamiques ?

Auteurs : **SilkyRoad** ,

La première partie de la procédure permet de remplir deux tableaux.

Ces tableaux sont ensuite utilisés pour alimenter les valeurs d'ordonnées et d'abscisses.

vba

```
Sub creationGraphiqueParTableau()
    Dim i As Byte
    Dim Tableau(10) As Integer, Tableau2(10) As Integer

    'Création du tableau pour les Abscisses
    For i = 1 To 10
        Tableau(i) = i * 2
    Next i

    'Création d'un tableau pour les Ordonnées
    For i = 1 To 10
        'Le tableau est rempli par des valeurs aléatoires pour
        'cet exemple
        Tableau2(i) = Int((50 * Rnd) + 1)
    Next i

    'Création graphique
    Charts.Add
    'Définit la localisation du graphique:
    'dans la feuille de calcul Feuil1 pour cet exemple
    ActiveChart.Location _
    Where:=xlLocationAsObject, Name:="Feuil1"

    'Ajoute une série dans le graphique
    With ActiveChart
        .SeriesCollection.NewSeries
        .SeriesCollection(1).XValues = Tableau() 'Abscisses
        .SeriesCollection(1).Values = Tableau2() 'Ordonnées
        'Définit le type (Courbe)
```

```
vba
        .ChartType = xlLine
    End With
End Sub
```

Remarque:

Cette méthode est limitée par le nombre de caractères que vous pourrez insérer dans la barre de formules (La fonction qui s'affiche lorsque vous sélectionnez une série dans le graphique)

Par exemple:

=SERIE(;{0.1.2.3.4.5.6.7.8.9.10};{36.13.10.45.21.44.40.19.49.44};1)

Cette limite est d'environ 450 caractères.

Comment insérer l'image d'un graphique dans un commentaire ?

Auteurs : SilkyRoad ,

Cet exemple exporte un graphique au format image GIF pour ensuite l'insérer comme image de fond dans le commentaire d'une cellule.

Cette solution prend un peu d'espace mémoire mais permet de stocker avantageusement un graphique (ou plutôt son image) quand on manque de place dans une feuille de calcul.

```
vba
Sub ImageGraphiqueDansCommentaire_CelluleA1()
    Dim nomImage As String
    Dim Grph As ChartObject
    Dim Hauteur As Single, Largeur As Single

    nomImage = "C:\imageTemp.gif"

    'Définit le 1er graphique dans la Feuille
    Set Grph = Feuille.ChartObjects(1)
    'Exporte le graphique au format image GIF
    Grph.Chart.Export nomImage, "GIF"

    'récupère la dimension du graphique pour l'appliquer au commentaire
    Hauteur = Grph.Height
    Largeur = Grph.Width

    'Vérifie s'il existe déjà un commentaire dans la cellule A1
    'et le supprime si c'est le cas
    If Not Feuille.Range("A1").Comment Is Nothing Then _
        Feuille.Range("A1").Comment.Delete

    'Creation du nouveau commentaire dans la cellule A1
    With Feuille.Range("A1")
        .AddComment
        .Comment.Visible = False
        'Définit la hauteur du commentaire
        'le graphique
        .Comment.Shape.Height = Hauteur
        'Définit la largeur du commentaire
        .Comment.Shape.Width = Largeur
        'Insère l'image dans le commentaire
        .Comment.Shape.Fill.UserPicture nomImage
    End With

    'Supprime l'image exportée
    Kill nomImage

    'Supprime le graphique
```

```
vba
Grph.Delete
End Sub
```

Comment appliquer un format particulier pour une des séries du graphique ?

Auteurs : **SilkyRoad** ,

Si vous utilisez un graphique de type histogramme composé de plusieurs séries, vous pouvez spécifier qu'une des séries doit être de type courbe (Pour par exemple créer une ligne de repère).

```
vba
'Définit un type d'affichage spécifique pour la 1ere série du graphique
'xlLine = Courbe
Feuill1.ChartObjects(1).Chart.SeriesCollection(1).ChartType = xlLine
```

Comment extraire les valeurs de chaque série d'un graph ?

Auteurs : **SilkyRoad** ,

Lorsqu'un graphique a perdu ses liaisons vers la source de données, la question peut se poser de savoir comment extraire les différentes valeurs contenues dans chaque série.

Une solution consiste à boucler sur tous les points afin d'en récupérer les valeurs.

Les étiquettes doivent impérativement être affichées (HasDataLabel = True) pour que les données puissent être récupérées.

```
vba
Sub extractionValeursOrdonnees_Series()
Dim Cible As ChartObject
Dim i As Integer, j As Integer

'Définit le 1er graphique de la Feuill
Set Cible = Feuill1.ChartObjects(1)

'Boucle sur toutes les séries du graphique
For j = 1 To Cible.Chart.SeriesCollection.Count

'Recupère le nom de la série
Debug.Print Cible.Chart.SeriesCollection(j).Name & ":"

'boucle sur tous les points de la série
For i = 1 To Cible.Chart.SeriesCollection(j).Points.Count

With Cible.Chart.SeriesCollection(j).Points(i)
'Affiche les étiquettes
.HasDataLabel = True
'Extrait la valeur du point
Debug.Print .DataLabel.Characters.Text
'Masque les étiquettes
.HasDataLabel = False
End With

Next i

Debug.Print "-----"
Next j
```

```
vba
End Sub
```

Comment afficher l'évolution d'une série en fonction du point précédent ?

Auteurs : **SilkyRoad** ,

Cette procédure permet d'afficher (en pourcentage) l'évolution des valeurs d'une série en fonction du point précédent. La macro boucle sur tous les points d'une série, calcule le rapport entre 2 points et ajoute le résultat dans une étiquette (DataLabel).

```
vba

Sub afficherEvolutionPourcentage_enFonctionDuPointPrecedent()
'Remarque : les labels ne seront pas mis à jour automatiquement
'si les données sont modifiées dans le tableau
Dim j As Integer
Dim X As Single, Y As Single
Dim Resultat As String
Dim Grph As ChartObject

'Définit le graphique cible dans la Feuille
Set Grph = Feuille1.ChartObjects(1)

On Error Resume Next
'suppression des Labels existants dans la 1ere série
Grph.Chart.SeriesCollection(1).DataLabels.Delete
On Error GoTo 0

'affiche les Labels (Ordonnées) de la 1ere série
'pour en extraire les valeurs
Grph.Chart.SeriesCollection(1).ApplyDataLabels _
    Type:=xlDataLabelsShowValue

'boucle sur les points de la 1ere série (en dehors du premier point)
For j = 2 To Grph.Chart.SeriesCollection(1).Points.Count

    X = Grph.Chart.SeriesCollection(1).Points(j). _
        DataLabel.Characters.Text 'valeur du point
    Y = Grph.Chart.SeriesCollection(1).Points(j - 1). _
        DataLabel.Characters.Text 'valeur du point précédent

    'Insère le pourcentage pour chaque point de la série.
    'Cette ligne est placée avant la ligne
    '"Resultat = Format((X / Y) - 1, "0.00%")" et permet ainsi
    'de ne rien afficher pour le 1er point.
    Grph.Chart.SeriesCollection(1).Points(j - 1). _
        DataLabel.Characters.Text = Resultat

    'calcul le pourcentage entre les 2 points
    Resultat = Format((X / Y) - 1, "0.00%")
Next j

'affiche le pourcentage pour le dernier point de la série
Grph.Chart.SeriesCollection(1).Points(j - 1).DataLabel. _
    Characters.Text = Resultat

'Mise en forme des étiquettes
With Grph.Chart.SeriesCollection(1).DataLabels
    .Font.ColorIndex = 5 'couleur bleue
    .Position = xlLabelPositionAbove 'position au dessus du point
    .Orientation = xlUpward 'orientation verticale (vers le haut)
End With
```

```
vba  
End Sub
```

Remarques:

La procédure ne gère pas les cellules vides (division par 0).

Les étiquettes ne seront pas mises à jour automatiquement si les données sont modifiées dans les cellules.

Comment imprimer un graphique ?

Auteurs : SilkyRoad ,

Cet exemple imprime le 2eme graphique de la Feuil1 :

```
vba  
Feuil1.ChartObjects(2).Chart.PrintOut
```

Pour imprimer une feuille graphique, utilisez :

```
vba  
Charts("Graph1").PrintOut
```

Comment empêcher le copier/coller d'une feuille graphique ?

Auteurs : SilkyRoad ,

Cette procédure événementielle doit être placée au niveau du classeur "ThisWorkBook"

La macro boucle les feuilles graphiques et annule le mode Couper ou Copier si l'onglet qui vient d'être désactivé est trouvé dans la collection.

```
vba  
  
Private Sub Workbook_SheetDeactivate(ByVal Sh As Object)  
    Dim Ch As Chart  
  
    'Boucle sur la collection de feuilles graphiques  
    For Each Ch In Charts  
        If Ch.Name = Sh.Name Then Application.CutCopyMode = False  
    Next Ch  
End Sub
```

Une autre possibilité, en utilisant l'événement Deactivate du graphique :

```
vba  
  
Private Sub Chart_Deactivate()  
    Application.CutCopyMode = False  
End Sub
```

Sommaire > Excel > Userforms, Code et Contrôles

Comment bloquer le code en utilisant une UserForm non modale ?

Auteurs : bbil ,

Il suffit, après l'affichage de la Userform, d'effectuer une boucle tant que celle-ci est visible :

```
vba
'-- Afficher la Form en non modale
UserForm1.Show False

'-- Boucle d'attente
While UserForm1.Visible
    DoEvents
Wend

'-- Suite du code ...
' ...
```

Comment trouver le Handle d'une Userform ?

Auteurs : bbil ,

De nombreuses fonctions de l'API Win32 agissant sur les fenêtres réclament en paramètre le handle (ou identificateur) de la dite fenêtre.

Malheureusement certains applicatifs tels Excel, Word... ne permettent pas de récupérer cette valeur.

la fonction FindWindowA, de l'API Win32, permet de connaître le handle d'une fenêtre à partir de son titre.

```
vba
Private Declare Function FindWindowA Lib "user32" _
    (ByVal lpClassName As String, ByVal lpWindowName As String) As Long

Private Sub UserForm_Initialize()
Dim MeHwnd As Long
    MeHwnd = FindWindowA(vbNullString, Me.Caption)
    MsgBox "le Handle de l'Userform " & Me.Caption & " est : 0x" & Hex(MeHwnd)
End Sub
```

Comment inhiber ou masquer le bouton "Fermer" d'une UserForm ?

Auteurs : bbil , ThierryAIM ,

Vous pouvez avoir besoin d'empêcher la fermeture d'une userform par la croix système.

2 solutions vous sont proposées ci-dessous :

1) La plus simple : inhiber l'action de la croix de fermeture dans l'évènement *QueryClose* de la UserForm :

```
vba
Private Sub UserForm_QueryClose(Cancel As Integer, CloseMode As Integer)
    If CloseMode = vbFormControlMenu Then Cancel = True
End Sub
```

2) La plus jolie : masquer le bouton de fermeture de la UserForm :

vba

```
'-- Dans la partie Déclaration de la Form :
Private Const SC_CLOSE = &HF060&
Private Const MF_BYCOMMAND = &H0&

Private Declare Function GetSystemMenu Lib "user32" _
    (ByVal hwnd As Long, ByVal bRevert As Long) As Long

Private Declare Function RemoveMenu Lib "user32" _
    (ByVal hMenu As Long, ByVal nPosition As Long, ByVal wFlags As Long) As Long

Private Declare Function FindWindowA Lib "user32" _
    (ByVal lpClassName As String, ByVal lpWindowName As String) As Long

'-- Dans l'évènement Initialise de la Form :
Private Sub UserForm_Initialize()
Dim hSysMenu As Long
Dim MeHwnd As Long
MeHwnd = FindWindowA(vbNullString, Me.Caption)
If MeHwnd > 0 Then
    hSysMenu = GetSystemMenu(MeHwnd, False)
    RemoveMenu hSysMenu, SC_CLOSE, MF_BYCOMMAND
Else
    MsgBox "Handle de " & Me.Caption & " Introuvable", vbCritical
End If
End Sub
```

Comment supprimer la barre de titre d'une UserForm ?

Auteurs : bbil ,

Il faut utiliser les fonctions de l'API Windows : SetWindowLong, SetWindowPos, GetWindowRect, GetWindowLong et hSysMenu
et, pour le cas d'Excel ou Word qui ne donne pas accès au handle de la fenêtre : FindWindowA...

Déclarations à placer dans un module standard :

vba

```
Public Type RECT
    Left As Long
    Top As Long
    Right As Long
    Bottom As Long
End Type

Const GWL_STYLE = (-16)
Const WS_CAPTION = &HC0000
Const SWP_FRAMECHANGED = &H20

Public Declare Function FindWindowA Lib "user32" _
    (ByVal lpClassName As String, ByVal lpWindowName As String) As Long

Public Declare Function GetWindowRect Lib "user32" _
    (ByVal hwnd As Long, lpRect As RECT) As Long

Public Declare Function GetWindowLong Lib "user32" Alias _
    "GetWindowLongA" (ByVal hwnd As Long, ByVal nIndex As Long) As Long

Public Declare Function SetWindowLong Lib "user32" Alias _
    "SetWindowLongA" (ByVal hwnd As Long, ByVal nIndex As Long, _
```

vba

```

        ByVal dwNewLong As Long) As Long

Public Declare Function SetWindowPos Lib "user32" _
    (ByVal hwnd As Long, ByVal hWndInsertAfter As Long, ByVal x As Long, _
    ByVal y As Long, ByVal cx As Long, ByVal cy As Long, _
    ByVal wFlags As Long) As Long
    
```

Fonction pour afficher ou masquer la barre de titre d'une UserForm (vous pouvez aussi la placer dans le module) :

vba

```

Sub AfficheTitleBarre(stCaption As String, pbVisible As Boolean)
Dim vrWin As RECT
Dim style As Long
Dim lHwnd As Long
'- Recherche du handle de la fenêtre par son Caption
lHwnd = FindWindowA(vbNullString, stCaption)
If lHwnd = 0 Then
    MsgBox "Handle de " & stCaption & " Introuvable", vbCritical
    Exit Sub
End If

GetWindowRect lHwnd, vrWin
style = GetWindowLong(lHwnd, GWL_STYLE)
If pbVisible Then
    SetWindowLong lHwnd, GWL_STYLE, style Or WS_CAPTION
Else
    SetWindowLong lHwnd, GWL_STYLE, style And Not WS_CAPTION
End If
SetWindowPos lHwnd, 0, vrWin.Left, vrWin.Top, vrWin.Right - vrWin.Left, _
    vrWin.Bottom - vrWin.Top, SWP_FRAMECHANGED
End Sub
    
```

Dans l'évènement Initialize de l'UserForm concernée :

vba

```

Private Sub UserForm_Initialize()
'On passe en arguments :
' - le titre de la fenêtre
' - False pour masquer la barre de titre
AfficheTitleBarre Me.Caption, False
End Sub
    
```

lien : [FAQ Afficher/Masquer la barre de titre d'une fenêtre \(FAQ VBA/ Access\)](#)

Comment supprimer un module de code dans un classeur externe ?

Auteurs : bbil ,

Vous devez ajouter la référence "*Microsoft visual Basic For Application Extensibility x.x*" à votre projet :

vba

```

Sub Macro1(ByVal nom As String)
Dim w As Workbook
Dim d As VbComponent

Set w = Workbooks.Open(FileName:="Classeur1.xls")
For Each d In w.VBProject.VBComponents
    If d.Type = vbext_ct_StdModule And d.Name = nom Then
    
```

vba

```
w.VBProject.VBComponents.Remove w.VBProject.VBComponents(d.Name)
End If
Next
w.Close True
```

lien : [FAQ](#) Comment supprimer une macro par le code ?

Comment supprimer une macro par le code ?

Auteurs : [SilkyRoad](#) ,

Vous devez connaître le nom de la macro et le nom du module dans laquelle elle se trouve.

Vous devez ajouter la référence "*Microsoft visual Basic For Application Extensibility x.x*" à votre projet :

vba

```
'Supprimer la macro nommée "MaMacro" dans le "module3"
Sub supprimerUneMacroPrecise()
Dim Debut As Integer, Lignes As Integer

With ThisWorkbook.VBProject.VBComponents("Module3").CodeModule
Debut = .ProcStartLine("MaMacro", 0)
Lignes = .ProcCountLines("MaMacro", 0)
.DeleteLines Debut, Lignes
End With
End Sub
```

lien : [FAQ](#) Comment supprimer un module de code dans un classeur externe ?

Sommaire > Excel > Manipuler WORD

Comment créer un nouveau document Word ?

Auteurs : [SilkyRoad](#) ,

Nécessite d'activer la référence "Microsoft Word xx.x Object Library".

vba

```
Dim WordApp As Word.Application
Dim WordDoc As Word.Document

Set WordApp = CreateObject("Word.Application")    '-- ouvre une session Word
WordApp.Visible = True
Set WordDoc = WordApp.Documents.Add    '-- crée un nouveau document
WordDoc.SaveAs "C:\monDocument.doc"    '-- enregistre le nouveau doc
```

Comment ouvrir un document Word existant ?

Auteurs : [SilkyRoad](#) ,

Nécessite d'activer la référence "Microsoft Word xx.x Object Library".

vba

```
Dim WordApp As Word.Application
Dim WordDoc As Word.Document

Set WordApp = CreateObject("Word.Application")
WordApp.Visible = True
Set WordDoc = WordApp.Documents.Open("monDocument.doc", ReadOnly:=True)
```

Comment piloter un document Word déjà ouvert ?

Auteurs : [SilkyRoad](#) ,

Nécessite d'activer la référence "Microsoft Word xx.x Object Library".

Pour piloter un document Word déjà ouvert , utilisez la fonction getObject :

vba

```
Dim WordDoc As Word.Document

Set WordDoc = GetObject("monDocument.doc")
MsgBox WordDoc.Paragraphs.Count
```

Comment modifier les marges d'un document Word ?

Auteurs : [SilkyRoad](#) ,

Nécessite d'activer la référence "Microsoft Word xx.x Object Library" :

vba

```
Dim WordApp As Word.Application
```

vba

```
Dim WordDoc As Word.Document

Set WordApp = CreateObject("Word.Application")
Set WordDoc = WordApp.Documents.Add
WordApp.Visible = True

With WordDoc.PageSetup
    .LeftMargin = CentimetersToPoints(1)
    .RightMargin = CentimetersToPoints(1)
    .TopMargin = CentimetersToPoints(1.5)
    .BottomMargin = CentimetersToPoints(2)
End With
```

Comment modifier l'entête et/ou le pied de page d'un document Word ?

Auteurs : **SilkyRoad** ,

Nécessite d'activer la référence "Microsoft Word xx.x Object Library" :

vba

```
Dim WordApp As Word.Application
Dim WordDoc As Word.Document

Set WordApp = CreateObject("Word.Application")
WordApp.Visible = True
Set WordDoc = WordApp.Documents.Open("monDocument.doc") 'adapter le chemin
With WordDoc.Sections(1)
    .Headers(wdHeaderFooterPrimary).Range.Text = "Le titre"
    .Headers(wdHeaderFooterPrimary).Range.Paragraphs.Alignment = wdAlignParagraphCenter
    .Footers(wdHeaderFooterPrimary).PageNumbers.Add
End With
```

Comment accéder aux paragraphes d'un document Word ?

Auteurs : **SilkyRoad** ,

Nécessite d'activer la référence "Microsoft Word xx.x Object Library" :

L'exemple ci-dessous montre comment boucler sur les paragraphes d'un document Word et les supprimer s'ils débutent par le mot "Test"

vba

```
Dim WordApp As Word.Application
Dim WordDoc As Word.Document
Dim Cible As Paragraph

Set WordApp = New Word.Application
WordApp.Visible = True

Set WordDoc = WordApp.Documents.Open(ThisWorkbook.Path & "\Doc1.doc")
WordDoc.Bookmarks("\StartOfDoc").Select

For Each Cible In WordDoc.Paragraphs
    Cible.Range.Select
    If Trim(Cible.Range.Words(1)) = "Test" Then Cible.Range.Delete
Next Cible
```

Un autre exemple qui supprime les paragraphes de façon conditionnelle :

vba

```
'boucle sur les 3 premiers paragraphes du document Word :
'si la cellule A1<>1 alors suppression du paragraphe 1
'si la cellule A2<>1 alors suppression du paragraphe 2
'si la cellule A3<>1 alors suppression du paragraphe 3
Dim WordApp As Word.Application
Dim WordDoc As Word.Document
Dim i As Integer

Set WordApp = CreateObject("Word.Application")
WordApp.Visible = True
Set WordDoc = WordApp.Documents.Open("C:\monDocument.doc")

For i = 3 To 1 Step -1
    If Cells(i, 1) <> 1 Then _
        WordDoc.Paragraphs.Item(i).Range.Delete
Next i
```

Comment vérifier si la première cellule d'un tableau est "vide" ?

Auteurs : SilkyRoad ,

Nécessite d'activer la référence "Microsoft Word xx.x Object Library".

vba

```
'Chr(13) & Chr(7) sont des caractères qui apparaissent par défaut dans les cellules lors de la création du tableau
If WordDoc.Tables(1).Columns(1).Cells(1).Range.Text = Chr(13) & Chr(7) Then
    MsgBox "Cellule vide"
Else
    MsgBox "Cellule non vide"
End If
```

Comment fusionner des cellules dans un document Word ?

Auteurs : SilkyRoad ,

Nécessite d'activer la référence "Microsoft Word xx.x Object Library" :

vba

```
Dim WordApp As Word.Application
Dim WordDoc As Word.Document

Set WordApp = CreateObject("word.application") 'Word Session
Set WordDoc = WordApp.Documents.Open("monDocument.doc") 'Ouverture Document Word
WordApp.Visible = True

'fusionner les Cells(2,3) à Cells(3,5) dans le premier tableau du document Word
WordDoc.Tables(1).Cell(Row:=2, Column:=3).Merge _
```

vba

```
mergeTo:=wordDoc.Tables(1).Cell(Row:=3, Column:=5)
```

Comment importer la totalité d'un document Word dans une feuille Excel ?

Auteurs : **SilkyRoad** ,

Nécessite d'activer la référence "Microsoft Word xx.x Object Library" :

vba

```
Dim WordDoc As Word.Document
Dim WordApp As Word.Application
Dim Wb As Workbook

Set Wb = Workbooks.Add(1)
Set WordApp = New Word.Application
WordApp.Visible = False
Set WordDoc = WordApp.Documents.Open("monDocument.doc", ReadOnly:=True)

With WordApp
    .Selection.WholeStory
    .Selection.Copy
End With

Wb.ActiveSheet.Range("A1").Select
Wb.ActiveSheet.Paste
WordApp.Application.Quit
Application.CutCopyMode = False
Wb.SaveAs "C:\copieDocument.xls"
```

Comment importer un tableau Word en intégrant les retours à la ligne ?

Auteurs : **SilkyRoad** ,

Nécessite d'activer la référence "Microsoft Word xx.x Object Library" :

Les retours à la ligne dans les cellules d'un tableau Word génèrent autant de cellules supplémentaires lors du collage dans Excel. Pour y remédier, cet exemple montre comment importer le premier tableau d'un document Word "C:\monFichier.doc" (déjà ouvert), en conservant le format des cellules :

vba

```
Dim WordDoc As Object
Dim i As Integer, j As Integer
Dim Cible As Variant

Set WordDoc = GetObject("C:\monFichier.doc")

For i = 1 To WordDoc.Tables(1).Rows.Count
    For j = 1 To WordDoc.Tables(1).Columns.Count
        Cible = WordDoc.Tables(1).Columns(j).Cells(i)

        Sheets(1).Cells(i, j) = _
            Application.WorksheetFunction.Substitute(Cible, vbCr, vbLf)
        Sheets(1).Cells(i, j) = _
            Left(Sheets(1).Cells(i, j), Len(Sheets(1).Cells(i, j)) - 1)
    Next j
Next i
```

vba

```
Next i
```

lien : [FAQ](#) Comment ouvrir un document Word existant ?

Comment importer les données provenant de plusieurs tableaux Word ?

Auteurs : [SilkyRoad](#) ,

Nécessite d'activer la référence "Microsoft Word xx.x Object Library" :

vba

```
Dim WordApp As Word.Application
Dim WordDoc As Word.Document
Dim i As Byte, j As Byte

Set WordApp = CreateObject("word.application")
wWrdApp.Visible = False
Set WordDoc = WordApp.Documents.Open("monFichier.doc")
'dans 3 tables Word du document , importer 5 valeurs de la premiere colonne
'importer les données de chaque table dans une feuille différente
For i = 1 To 3
    For j = 1 To 5
        ActiveWorkbook.Sheets(i).Cells(j, 1) = WordDoc.Tables(i).Columns(1).Cells(j)
    Next j
Next i
WordDoc.Close
WordApp.Quit
```

Comment copier une ligne précise d'un tableau Word et la coller dans Excel ?

Auteurs : [SilkyRoad](#) ,

Nécessite d'activer la référence "Microsoft Word xx.x Object Library" :

vba

```
Dim WordApp As Word.Application
Dim WordDoc As Word.Document

Set WordApp = CreateObject("word.application")
WordApp.Visible = False 'Word reste masqué pendant l'opéraion
Set WordDoc = WordApp.Documents.Open("monDocument.doc") 'ouvre le document Word

'copies la 3eme ligne de la 1ere table Word
WordDoc.Tables(1).Rows(3).Range.Copy

'collage dans Excel
Range("A1").PasteSpecial xlPasteValues

WordDoc.Close 'fermeture document Word
```

vba

```
WordApp.Quit 'fermeture session Word
```

Comment exporter un tableau Excel vers Word et l'adapter à la largeur de la page ?

Auteurs : SilkyRoad ,

Nécessite d'activer la référence "Microsoft Word xx.x Object Library"

vba

```
Dim WordApp As Word.Application
Dim WordDoc As Word.Document

Set WordApp = New Word.Application
WordApp.Visible = True
Set WordDoc = WordApp.Documents.Add

Range("A1:H10").Copy

WordApp.Selection.Paste
WordDoc.Tables(1).AutoFitBehavior wdAutoFitWindow

Application.CutCopyMode = False
```

Comment exporter des données Excel dans des cellules précises d'un tableau Word ?

Auteurs : SilkyRoad ,

Nécessite d'activer la référence "Microsoft Word xx.x Object Library" :

vba

```
Dim WordApp As Word.Application
Dim WordDoc As Word.Document

Set WordApp = CreateObject("word.application")
WordApp.Visible = True 'mettre False pour garder Word masqué
Set WordDoc = WordApp.Documents.Open("monDocument.doc") 'ouvre le document Word

'Tables(2) correspond au 2eme tableau du document Word
'transfert la donnée de la cellule A1 dans la 3eme cellule de la 1ere colonne
WordDoc.Tables(2).Columns(1).Cells(3).Range.Text = Range("A1")

'transfert la donnée de la cellule A2 dans la 2eme cellule de la 3eme colonne
WordDoc.Tables(2).Columns(3).Cells(2).Range.Text = Range("A2")

'WordDoc.Close True 'ferme le document Word en enregistrant les modifications
'WordApp.Quit 'ferme l'application Word
```

Comment exporter des données Excel vers plusieurs signets d'un document Word ?

Auteurs : SilkyRoad ,

Nécessite d'activer la référence "Microsoft Word xx.x Object Library" :

L'exemple ci-dessous permet d'exporter les valeurs des cellules A1 à A3 vers des signets d'un document Word nommé "monfichier.doc"

Les signets positionnés aux endroits de votre choix doivent être nommés "Signet1 , Signet2 et Signet3"

vba

```

Dim WordApp As Word.Application
Dim WordDoc As Word.Document
Dim i As Byte

Set WordApp = CreateObject("word.application") 'ouvre une session Word
Set WordDoc = WordApp.Documents.Open("monDocument.doc") 'ouvre le document Word
WordApp.Visible = False 'Word est masqué pendant l'opération

For i = 1 To 3
    'les signets du document Word sont nommés Signet1 , Signet2 , Signet3
    WordDoc.Bookmarks("Signet" & i).Range.Text = Cells(i, 1)
Next i

WordApp.Visible = True 'affiche le document Word
'WordDoc.PrintOut 'Pour imprimer le doc obtenu

'WordDoc.Close True 'ferme le document word en sauvegardant les données
'WordApp.Quit 'ferme la session Word
    
```

Comment inserer la date du jour dans un signet Word ?

Auteurs : **SilkyRoad** ,

Nécessite d'activer la référence "Microsoft Word xx.x Object Library" :

Dans l'exemple ci-dessous le signet à renseigner doit être nommé "SignetDate"

vba

```

Dim WordApp As Word.Application
Dim WordDoc As Word.document

Set WordApp = New Word.Application
WordApp.Visible = True
Set WordDoc = WordApp.Documents.Open("monDocument.doc")

WordDoc.Bookmarks("SignetDate").Range.Text = Format(Now, "dd/mm/yyyy")
    
```

Comment insérer, redimensionner et positionner une image dans un document Word existant ?

Auteurs : **SilkyRoad** ,

Nécessite d'activer la référence "Microsoft Word xx.x Object Library" :

vba

```

Dim WordApp As Word.Application
Dim WordDoc As Word.Document
Dim Fichier As String

On Error Resume Next

Set WordApp = CreateObject("Word.Application")
    
```

vba

```

WordApp.Visible = True
Set WordDoc = WordApp.Documents.Open("monDocument.doc") 'ouvrir le document Word

WordDoc.InlineShapes.AddPicture Filename:= _
                                "C:\Image1.JPG" 'adapter le chemin de l'image à insérer

On Error GoTo 0

With WordDoc.InlineShapes(1) 'adapter l'index si d'autres images existantes dans le document
    .Height = 190.75 'changement dimension image insérée : hauteur
    .Width = 254# 'largeur
    .ConvertToShape
End With

With WordDoc.Shapes(1)
    .Top = 200 'position verticale de l'image dans le document
    .Left = 150 'position horizontale de l'image dans le document
    .ZOrder msoBringToFrontOfText 'image au premier plan devant le texte
    '.ZOrder msoSendBehindText 'option pour image en arriere plan derriere le texte
End With

'WordDoc.Close 'fermer le document Word
'WordApp.Quit 'fermer l'application Word
    
```

Comment coller dans Word une selection de cellules au format image Bitmap ?

Auteurs : SilkyRoad ,

Nécessite d'activer la référence "Microsoft Word xx.x Object Library" :

vba

```

Dim WordApp As Word.Application

Selection.Copy

Set WordApp = CreateObject("Word.Application")
On Error Resume Next
WordApp.Documents.Add
WordApp.Visible = True

WordApp.Selection.PasteSpecial DataType:=wdPasteBitmap
Application.CutCopyMode = False
    
```

Comment ajouter une colonne dans un tableau d'un document Word ?

Auteurs : SilkyRoad ,

Nécessite d'activer la référence "Microsoft Word xx.x Object Library" :

L'exemple ci-dessous insère une nouvelle colonne en 3eme position dans le 2e tableau d'un document Word. La premiere cellule de cette nouvelle colonne est coloriée en bleu et un texte y est inséré.

vba

```

Dim WordApp As Word.Application
Dim WordDoc As Word.Document

'le document Word est supposé fermé avant le lancement de la macro
Set WordApp = CreateObject("Word.Application") 'creation session Word
    
```

vba

```

WordApp.Visible = False 'pour que word reste masqué pendant l'opération
Set WordDoc = WordApp.Documents.Open("monDocument.doc") 'ouverture du fichier Word

'insertion d'une colonne en 3eme position dans le 2eme tableau du document Word
'colorie en bleu la 1ere cellule dans la colonne insérée
With WordDoc.Tables(2)
    .Columns.Add BeforeColumn:=WordDoc.Tables(2).Columns(3)
    .Columns(3).Cells(1).Shading.BackgroundPatternColorIndex = wdBlue
'fond de cellule en bleu
    .Columns(3).Cells(1).Range.Text = "le forum dvp.com" 'texte dans la cellule
    .AutoFitBehavior wdAutoFitWindow ' adapte la dimension du tableau à la feuille
End With

WordDoc.Close True 'ferme le document Word en sauvegardant les modifications
WordApp.Quit 'ferme l'application Word
    
```

Comment insérer une image dans une cellule d'un tableau d'un document Word ?

Auteurs : SilkyRoad ,

Nécessite d'activer la référence "Microsoft Word xx.x Object Library" :

L'exemple ci-dessous insère une image dans la 3e cellule de la 2e colonne du 1er tableau d'un document Word.

vba

```

Dim WordApp As Word.Application
Dim WordDoc As Word.Document

Set WordApp = CreateObject("word.application") 'ouvrir une session Word
Set WordDoc = WordApp.Documents.Open("monDocument.doc") 'ouvrir un document
'insérer une image dans la 3eme Cellule de la 2eme colonne (dans le
'1er tableau d'un document Word )
WordDoc.Tables(1).Columns(2).Cells(3).Range.InlineShapes.AddPicture _
    Filename:="C:\image1.wmf", linkToFile:=False, saveWithDocument:=True

With WordDoc.InlineShapes(WordDoc.InlineShapes.Count)
    .Height = 150 'redimensionne hauteur image
    .Width = 150 'redimensionne largeur image
End With
WordApp.Visible = True 'affichier le document Word
    
```

Comment récupérer la donnée d'un champ de fusion dans un document Word ouvert ?

Auteurs : SilkyRoad ,

Nécessite d'activer la référence "Microsoft Word xx.x Object Library"

Le document Word doit être ouvert :

vba

```

Dim WordApp As Word.Application
Dim WordDoc As Word.Document

On Error Resume Next
Set WordApp = GetObject(, "Word.Application")
Set WordDoc = WordApp.Documents("monDocument.doc")

If WordDoc Is Nothing Then
    
```

vba

```
MsgBox "Le document est fermé"  
Else  
MsgBox WordDoc.MailMerge.DataSource.DataFields("Nom_Champ").Value  
End If
```

lien : [FAQ](#) Comment ouvrir un document Word existant ?

Comment exécuter une macro d'un document Word ?

Auteurs : [SilkyRoad](#) ,

Nécessite d'activer la référence "Microsoft Word xx.x Object Library".

vba

```
Dim WordApp As Word.Application  
  
Set WordApp = CreateObject("Word.Application")  
WordApp.Visible = True  
  
WordApp.Documents.Open ("C:\monDocument.dot")  
WordApp.Run "laMacro"
```

Comment vérifier si la première cellule d'un tableau est "vide" ?

Auteurs : [SilkyRoad](#) ,

Nécessite d'activer la référence "Microsoft Word xx.x Object Library".

vba

```
'Chr(13) & Chr(7) sont des caractères qui apparaissent par défaut dans les cellules lors de la création du tabl  
If WordDoc.Tables(1).Columns(1).Cells(1).Range.Text = Chr(13) & Chr(7) Then  
MsgBox "Cellule vide"  
Else  
MsgBox "Cellule non vide"  
End If
```

Sommaire > Word**Préambule-généralités****Auteurs : bidou ,**

Le pilotage de Word est plutôt plus simple que celui d'Excel. Le problème qui fait penser l'inverse et le déplacement dans le document. La plage que l'on cible est un objet Range. Le document se subdivise Globalement comme suit: Document -> StoryRanges -> Paragraphs -> Sentences -> Words -> Characters StoryRanges représentent les zones de votre document

- **wdMainTextStory : texte principal**
- **wdEvenPagesHeaderStory : en-tête**
- **wdEvenPagesFooterStory : pied de page**

Paragraphs représente une zone terminée par un retour chariot Sentences est une zone terminée par un point. Words et characters sont suffisamment explicites. Il existe d'autres collections particulières tel que : Les signets (Bookmarks) Les listes (ListParagraphs) Les tableaux (Tables) Les champs (Fields) Les contrôles de formulaire (FormFields) etc#

Sommaire > Word > Application

Peut-on ajouter des raccourcis-clavier par le code ?

Auteurs : bidou ,

C'est assez facile. L'exemple suivant affecte la mise en exposant à CTRL+MAJ+E

```
CustomizationContext = NormalTemplate
KeyBindings.Add KeyCode:=BuildKeyCode(wdKeyE, wdKeyControl, wdKeyShift),
KeyCategory:=wdKeyCategoryCommand, Command:="Superscript"
```

Sommaire > Word > Document

Comment ouvrir un document ?

Auteurs : bidou ,

Trois méthodes sont utilisables Ouvrir un document existant

```
Set objDoc = Application.Documents.Open("c:\tutoriel\tuto.doc")
```

Ouvrir un nouveau document (basé sur le modèle normal)

```
Set objDoc = Application.Documents.Add
```

Ouvrir un document basé sur un autre modèle

```
Set objDoc = Application.Documents.Add(Template:="C:\Program Files\Microsoft Office\Templates\Memos\Professional Memo.dot")
```

Peut on récupérer les propriétés du document (auteur, titre) ?

Auteurs : bidou ,

On peut accéder aux propriétés par la collection **BuiltInDocumentProperties** de l'objet document

```
Dim objDoc As Word.Document, msg As String

Set objDoc = Application.Documents.Open("d:\tuto.doc")
msg = "Titre : " & objDoc.BuiltInDocumentProperties(wdPropertyTitle) & vbCrLf
msg = msg & "Auteur : " & objDoc.BuiltInDocumentProperties(wdPropertyAuthor) & vbCrLf
msg = msg & "Commentaire : " & objDoc.BuiltInDocumentProperties(wdPropertyComments) & vbCrLf
MsgBox msgDim objDoc As Word.Document, msg As String

Set objDoc = Application.Documents.Open("d:\tuto.doc")
msg = "Titre : " & objDoc.BuiltInDocumentProperties(wdPropertyTitle) & vbCrLf
msg = msg & "Auteur : " & objDoc.BuiltInDocumentProperties(wdPropertyAuthor) & vbCrLf
msg = msg & "Commentaire : " & objDoc.BuiltInDocumentProperties(wdPropertyComments) & vbCrLf
MsgBox msg
```

Peut-on générer les statistiques par le code ?

Auteurs : bidou ,

Oui, il faut par contre demander le calcul pour une statistique

```
Dim objDoc As Word.Document, msg As String

Set objDoc = Application.Documents.Open("d:\tuto.doc")
MsgBox "Le documents contient " & objDoc.ComputeStatistics(wdStatisticPages) & " pages et " & objDoc.ComputeStatistics(wdStatisticWords) & " mots"
```

Vous pouvez obtenir les statistiques pour

- **wdStatisticCharacters** : nombre de caractères
- **wdStatisticCharactersWithSpaces** : nombre de caractères avec les espaces
- **wdStatisticLines** : nombre de lignes
- **wdStatisticPages** : nombre de pages

- wdStatisticParagraphs : nombre de paragraphes (retour chariot)
- wdStatisticWords : nombre de mots

Je ne trouve pas la propriété visible d'un document Word 97 ?

Auteurs : bidou ,

C'est normal, elle n'existe pas. Pour gérer un document invisible dans word 97, on utilise :

```
Set objDoc = GetObject("d:\user\tuto.doc")
```

Comment fonctionnent les protections Word ?

Auteurs : bidou ,

Elles sont de plusieurs sortes. Mot de passe d'accès

```
objDoc.Password= "passe"
```

Ce code doit être passé dans la méthode open pour ouvrir le document par le code. Protection spécifique

```
ActiveDocument.Protect Password="monpasse", NoReset=False, Type=wdAllowOnlyComments
```

Dans ce cas, seuls les commentaires sont autorisés. Pour ôter la protection, utiliser le type wdNoProtection ou la méthode UnProtect

Comment mettre en page le document ?

Auteurs : bidou ,

On utilise pour cela l'objet PageSetup de document. Par exemple

```
With ActiveDocument.PageSetup
    .Orientation = wdOrientLandscape
    .FirstPageTray = wdPrinterDefaultBin
    .OtherPagesTray = wdPrinterDefaultBin
    .SectionStart = wdSectionNewPage
    .OddAndEvenPagesHeaderFooter = True
    .DifferentFirstPageHeaderFooter = True
    .VerticalAlignment = wdAlignVerticalTop
    .SuppressEndnotes = False
    .MirrorMargins = True
End With
```

Comment insérer une table des matières ?

Auteurs : bidou ,

On utilise la méthode Add de la collection TableOfContents Set objDoc = Application.Documents.Open("D:\tuto.doc")

```
objDoc.TablesOfContents.Add _
    Range:=objDoc.Range(objDoc.Paragraphs(2).Range.Start, objDoc.Paragraphs(2).Range.Start), _
```

```
UseHeadingStyles:=True, _  
LowerHeadingLevel:=4, _  
UpperHeadingLevel:=1
```

Faite très attention à ne pas sélectionner de texte car celui-ci serait écrasé.

Comment créer et mettre en forme un en-tête ?

Auteurs : bidou ,

Le code suivant crée l'en-tête, centre le texte et le met en gras

```
ActiveDocument.Sections(1).Headers(wdHeaderFooterPrimary).Range.Text = "En-tête"  
With ActiveDocument.StoryRanges(wdPrimaryHeaderStory)  
    .Bold = True  
    .ParagraphFormat.Alignment = wdAlignParagraphCenter  
End With
```

Comment ajouter un lien hypertext à mon document ?

Auteurs : bidou ,

On utilise la méthode Add de la collection Hyperlinks

```
Dim objDoc As Document  
  
Set objDoc = Application.Documents.Open("D:\tuto.doc")  
objDoc.Hyperlinks.Add objDoc.Paragraphs(1).Range.Words(objDoc.Paragraphs(1).Range.Words.Count), "d:  
\pilo.doc"
```

Que représente la collection Sentences ?

Auteurs : bidou ,

Sentences est une collection renvoyant les "phrases" de son conteneur (document, paragraphe ...). Par phrase on entend tout ce qui est compris entre des points et/ou des retours chariots.

```
Notez que c'est donc différent des lignes word  
Dim objDoc As Document, Phrase As Range  
  
Set objDoc = Application.Documents.Open("D:\tuto.doc")  
MsgBox objDoc.Sentences.Count & vbCrLf & objDoc.BuiltInDocumentProperties(wdPropertyLines)  
For Each Phrase In objDoc.Sentences  
    Phrase.Select  
    MsgBox "suite"  
Next
```

Sommaire > Word > Document > Sélection et Range

Comment sélectionne-t-on avec VBA Word

Auteurs : bidou ,

Quelques exemples standards. Sélectionner tout le document

```
objDoc.Range(0, 0).Select  
Selection.MoveEnd wdStory
```

Aller au début du document

```
Selection.HomeKey Unit:=wdStory
```

Aller à la fin du document

```
Selection.EndKey Unit:=wdStory
```

Sélectionner une ligne

```
Selection.EndKey Unit:=wdLine, Extend:=wdExtend
```

Sélectionner le troisième paragraphe

```
objDoc.Paragraphs(3).Range.Select
```

Pourquoi ai-je toujours un texte sélectionné ?

Auteurs : bidou ,

Il y a beaucoup de confusion possible entre range et selection or les deux objets sont différents. Pour pouvoir basculer facilement de l'un à l'autre il faut utiliser une variable Range. Prenons un exemple

```
Dim objDoc As Document  
  
Set objDoc = Application.Documents.Open("c:\jmarc\tuto.doc")  
objDoc.Paragraphs(3).Range.Select  
objDoc.Paragraphs(3).Range.EndOf wdWord, wdMove
```

Là, malgré le EndOf, le paragraphe est encore sélectionné

```
Dim objDoc As Document, objRange As Range  
  
Set objDoc = Application.Documents.Open("c:\jmarc\tuto.doc")  
objDoc.Paragraphs(3).Range.Words(objDoc.Paragraphs(3).Range.Words.Count - 2).Select  
Set objRange = objDoc.Paragraphs(3).Range  
objRange.EndOf wdWord, wdMove  
objRange.Select
```

Place le point d'insertion après le troisième paragraphe

Pourquoi je n'arrive pas à sélectionner le dernier mot de mon paragraphe ?

Auteurs : bidou ,

L'utilisation directe de la collection Words sur le paragraphe peut être trompeuse.

```
Dim objDoc As Document, objRange As Range

Set objDoc = Application.Documents.Open("c:\jmarc\tuto.doc")
Set objRange = objDoc.Paragraphs(8).Range
objRange.Words(objRange.Words.Count).Select
```

Ce code va sélectionner le retour chariot du paragraphe qui stricto sensu est le dernier mot du paragraphe. Accessoirement, l'utilisation de count-1 ne serait pas meilleure puisque selon que la phrase est un point ou non, celui-ci serait sélectionné. Il faut donc faire une vérification arrière

```
Dim objDoc As Document, objRange As Range

Set objDoc = Application.Documents.Open("c:\jmarc\tuto.doc")
Set objRange = objDoc.Paragraphs(8).Range.Sentences(objDoc.Paragraphs(8).Range.Sentences.Count)
Set objRange = objRange.Words.Last
Do
    Set objRange = objRange.Previous(wdWord, 1)
Loop While StrComp(objRange.Text, vbCrLf, vbBinaryCompare) = 0 Or StrComp(objRange.Text, ".", vbBinaryCompare) = 0
objRange.Select
```

Comment faire une recherche avec Range qui modifie le format du mot cherché ?

Auteurs : bidou ,

Normalement, la sélection ne se fait que si Selection est la cible de la recherche ; Néanmoins, vous pouvez procéder à la sélection en désignant le parent de l'objet Find. Le code suivant encadre le texte GNU

```
Dim objDoc As Document, objRange As Range

Set objDoc = Application.Documents.Open("c:\jmarc\tuto.doc")
With objDoc.Content.Find
    .ClearFormatting
    Do While .Execute(FindText:="GNU", Forward:=True, Format:=True) = True
        With .Parent
            .StartOf Unit:=wdWord, Extend:=wdExtend
            .Borders.Enable = True
        End With
    Loop
End With
```

Comment faire un remplacement de toutes les occurrences d'un mot ?

Auteurs : bidou ,

Avec la méthode find et les bons paramètres

```
Set objDoc = Application.Documents.Open("c:\jmarc\tuto.doc")
```

```
objDoc.Content.Find.Execute FindText:="GNU", ReplaceWith:="GPL", Replace:=wdReplaceAll
```

Pourquoi travailler avec l'objet Range et l'objet Selection ?

Auteurs : bidou ,

Cela permet de gérer des opérations sur une plage sans modifier l'emplacement du point d'insertion. Supposons que je veuille insérer deux signets, l'un sur un mot, l'autre sur le point d'insertion, je peux faire :

```
Selection.TypeText Text:="Monsieur "  
Set objRange = Selection.Words.First.Previous(wdWord)  
objRange.Bookmarks.Add "genre"  
With ActiveDocument.Bookmarks  
    .Add Range:=Selection.Range, Name:="Nom"  
    .DefaultSorting = wdSortByName  
    .ShowHidden = True  
End With
```

Comment insérer et modifier une image ?

Auteurs : bidou ,

Les images insérées dans le texte sont membres de la collection InlineShapes

```
Dim objDoc As Document, objShape As InlineShape  
  
Set objDoc = Application.Documents.Add  
Set objShape = Selection.InlineShapes.AddPicture(FileName:="C:\developpez\logo_developpez.gif",  
    LinkToFile:=False, SaveWithDocument:=True)  
With objShape  
    .LockAspectRatio = msoTrue  
    .Height = .Height * 0.5  
    .Width = .Width * 0.5  
End With
```

Ce code insère une image est la réduit de moitié

Lorsque je change le texte de mon paragraphe la mise en page change ?

Auteurs : bidou ,

Il s'agit là d'une erreur classique. Vous confondez le paragraphe et l'objet Range le désignant. Prenons un exemple :

```
Dim objDoc As Document, objRange As Range  
  
Set objDoc = Application.Documents.Open("c:\jmarc\tuto.doc")  
Set objRange = objDoc.Paragraphs(8).Range  
objRange.Text = "Ceci est un nouveau paragraphe"
```

Ma phrase va s'insérer comme la première phrase du paragraphe suivant, car la marque de paragraphe est incluse dans l'objet Range. La syntaxe correcte est :

```
Dim objDoc As Document, objRange As Range  
  
Set objDoc = Application.Documents.Open("c:\jmarc\tuto.doc")  
Set objRange = objDoc.Paragraphs(8).Range
```

```
objRange.Text = "Ceci est un nouveau paragraphe"  
objRange.InsertParagraphAfter
```

Sommaire > Word > Document > Tableau et signets

Pourquoi mon signet est-il détruit lorsque je modifie son texte ?

Auteurs : bidou ,

Il y a toujours destruction d'un signet portant sur une plage. Le signet fait partie de la plage.

```
Dim objRange As Range
Set objRange = ActiveDocument.Bookmarks.Item("genre").Range
objRange.Text = "madame"
```

Le signet 'genre' est détruit dans cette opération. Il faut toujours recréer le signet après, du style

```
ActiveDocument.Bookmarks("genre").Range.Select
Selection.Text = "Madame"
objRange.Bookmarks.Add "genre"
```

Pour éviter cela, il vaut mieux utiliser des signets d'insertion (défini sur un point)

Comment insérer un signet qui ne contient pas de texte ?

Auteurs : bidou ,

On appelle la méthode collapse de la sélection avant l'insertion

```
selection.Collapse Direction:=wdCollapseEnd
With ActiveDocument.Bookmarks
    .Add Range:=Selection.Range, Name:="Nom"
    .DefaultSorting = wdSortByName
    .ShowHidden = True
End With
```

Comment créer et remplir un tableau ?

Auteurs : bidou ,

Utiliser une variable objet Table.

```
Dim objTable As Table
Set objTable = objDoc.Tables.Add(Range:=Selection.Range, NumRows:=5, NumColumns:=3)
Dim cmpt As Long
For cmpt = 1 To objTable.Rows.Count
    objTable.Cell(cmpt, 2).Range.Text = "montext" & cmpt
Next cmpt
```

La navigation dans les tableaux est assez similaire à celle d'Excel avec des objets Cells, Columns etc.... Néanmoins, pour accéder au contenu d'une cellule, vous devez passer par l'objet Range de l'objet Cell.

Comment gère-t-on la mise en forme d'un tableau ?

Auteurs : bidou ,

Deux possibilités. On applique un format prédéfini

```
objTable.AutoFormat wdTableFormatGrid8
```

On gère soit même

```
objTable.AutoFitBehavior (wdAutoFitContent)
With objTable
    .Borders(wdBorderLeft).LineStyle = wdLineStyleSingle
    .Borders(wdBorderRight).LineStyle = wdLineStyleSingle
    .Borders(wdBorderTop).LineStyle = wdLineStyleSingle
    .Borders(wdBorderBottom).LineStyle = wdLineStyleSingle
    .Borders(wdBorderHorizontal).LineStyle = wdLineStyleDot
    .Borders(wdBorderVertical).LineStyle = wdLineStyleDot
    .Borders.Shadow = False
End With
```

Ce code ajuste la largeur des colonnes à leur contenu, encadre le tableau avec une ligne continue et les cellules en pointillé.

Comment scinder un tableau qui fait plus de 10 lignes en dupliquant les titres ?

Auteurs : bidou ,

La manipulation est un peu particulière. Si on travaille uniquement avec les collections exposées par le tableau, on n'accède pas à des méthodes comme Copy ou Paste. Par contre, l'objet Selection expose ces méthodes.

```
Set objTable = ThisDocument.Tables(1)
If objTable.Rows.Count > 10 Then
    objTable.Rows(1).Select
    Selection.Copy
    objTable.Rows(11).Select
    Selection.Paste
    objTable.Rows(11).Select
    Selection.SplitTable
End If
```

Comment atteindre une plage comprise entre deux signets ?

Auteurs : Maxence Hubiche , bidou ,

On utilise la décomposition du range en Start et End

```
ActiveDocument.Range(ActiveDocument..Bookmarks("sig1").Range.Start,
    ActiveDocument..Bookmarks("sig2").Range.End).Select
```