

Astuces et code VBA pour Excel

Rubriques

- [Classeurs \(Workbooks\)](#)
- [Contrôles et Formulaires](#)
- [Feuilles \(Worksheets\)](#)
- [Fichiers \(XL ou autres\)](#)
- [Fonctions](#)
- [Formules](#)
- [Graphiques](#)
- [Menus](#)
- [Sélection](#)
- [OLE Automation](#)
- [Excel <--> Access \(DAO\)](#)
- [Commandes diverses](#)
- [Programmes divers ...](#)
- [Téléchargement](#) sur site <http://ericrenaud.free.fr/>

Les classeurs Excel (Workbooks)

Page mise à jour le : 28/03/2001

Vous trouverez dans cette rubrique des exemples de programmes se rapportant aux classeurs.

Ajouter des feuilles dans un classeur et les renommer.
Enregistrer une copie du classeur (sans modifier le classeur ouvert).
Fermer tous les classeurs ouverts sauf le classeur actif.
Renommer les onglets d'un classeur (Ex. Mois1, Mois2, Mois3, etc...).
Trier les onglets d'un classeur.

Ajouter des feuilles dans un classeur et les renommer

```
Sub AjouterRenommerFeuilles()  
Dim cpt As Integer  
cpt = 1  
Do While cpt < 4 ' Ajoute 3 feuilles  
' Ajout d'une feuille  
Application.Sheets.Add After:=Sheets.Item(Sheets.Count), Type:=xlWorksheet  
' Renomme la feuille  
Application.ActiveSheet.Name = "Semaine " & CStr(cpt)  
cpt = cpt + 1  
Loop  
End Sub
```

Enregistrer une copie du classeur

```
Sub SaveCopyAs()  
ActiveWorkbook.SaveCopyAs "C:\excel\MonDouble.xls"  
End Sub
```

Fermer tous les classeurs ouverts (sauf le classeur actif)

```
Sub FermeClasseurs()  
For Each Wk In Workbooks  
If Wk.Name <> ThisWorkbook.Name Then  
Wk.Close savechanges:=True  
End If  
Next Wk  
End Sub
```

Renommer les onglets d'un classeur

```
Sub RenommeOnglets()  
' Renomme les onglets CL1, CL2, CL3, etc ...  
Dim I As Integer  
Application.ScreenUpdating = False  
For I = 1 To 3  
    Worksheets(I).Name = "CL" & I  
Next I  
End Sub
```

Trier les onglets d'un classeur

```
Sub TriNomsOnglets()  
Dim I As Integer, J As Integer  
For I = 1 To Sheets.Count  
    For J = 1 To I - 1  
        If UCase(Sheets(I).Name) < UCase(Sheets(J).Name) Then  
            Sheets(I).Move Before:=Sheets(J)  
        End If  
    Next J  
Next I  
End Sub
```

Contrôles divers en VBA

Page mise à jour le : 14/05/2002

Vous trouverez dans cette rubrique des exemples sur les différents contrôles comme TextBox, ComboBox, ListBox, Bouton d'option, Case à cocher, calendrier, indicateur de progression, etc...

Ces contrôles s'utilisent sur un Userform (ou formulaire)

Quelques notions de base ...

Comment créer un Userform ?

Depuis une feuille Excel, tapez **ALT+F11** ou par le menu **Outils, Macro, Visual Basic Editor**.

Vous affichez l'environnement Visual Basic Editor.

Insérer un Userform à l'aide du menu Insertion, Userform.

La boîte à outils doit s'afficher, sinon utilisez le menu **Affichage, Boîte à outils**.

Pour placer un contrôle sur un Userform :

- 1 - Sélectionner le contrôle désiré dans la fenêtre Boîte à outils
- 2 - Cliquer sur le contrôle choisi avec le bouton gauche de la souris et sans relâcher celui-ci, déplacer le contrôle vers le Userform puis relâcher le bouton de la souris.

Voici les principaux contrôles de la boîte à outils (ci-dessous).

Les boîtes de dialogue

Boite de dialogue avec bouton d'option.

Le Userform (DlgChoix)

Le code

[Code lançant l'ouverture de la boîte de dialogue](#)

```
Sub AfficheBoutonsOption()  
  DlgChoix.Show  
End Sub
```

[Code d'ouverture du formulaire](#)

```
Private Sub UserForm_Activate()  
  DlgChoix.ChoixPaire.Value = True  
End Sub  
Nota : A l'ouverture du formulaire la case France est cochée par défaut.
```

[Code des boutons](#)

1 / Bouton OK

```
Private Sub OK_Click()  
  DlgChoix.Hide  
  If DlgChoix.ChoixPaire Then  
    Msg = "Vous avez choisi l'activité France."  
    Style = vbOKOnly + vbInformation  
    Title = "Activité Commerciale - © E. RENAUD - Novembre 1999"  
    Response = MsgBox(Msg, Style, Title)  
  Else  
    Msg = "Vous avez choisi l'activité Export."  
    Style = vbOKOnly + vbInformation  
    Title = "Activité Commerciale - © E. RENAUD - Novembre 1999"  
    Response = MsgBox(Msg, Style, Title)  
  End If  
End Sub
```

2 / Bouton Annuler

```
Private Sub Annuler_Click()  
  DlgChoix.Hide  
End Sub
```

Boite de dialogue avec case à cocher.

Le Userform (CasesACocher)

Le code

[Code lançant l'ouverture de la boîte de dialogue](#)

```
Sub AfficheCaseACocher()  
  Range("A3:C3").Select  
  Selection.ClearContents  
  CasesACocher.Show  
End Sub
```

[Code d'ouverture du formulaire](#)

```
Private Sub UserForm_Activate()  
' Remplace la propriété TripleState mise à True(CheckBox1)  
CheckBox1.Value = Null  
CheckBox2.Value = False  
CheckBox3.Value = False  
End Sub
```

Nota : A l'ouverture du formulaire la case Français est coché mais en grisée.

[Code des boutons](#)

1 / Bouton OK

```
Private Sub OK_Click()  
Dim Langue1 As String  
Dim Langue2 As String  
Dim Langue3 As String  
If CheckBox1.Value = False Then  
    Langue1 = "Non"  
Else  
    Langue1 = "Oui"  
End If  
If CheckBox2.Value = False Then  
    Langue2 = "Non"  
Else  
    Langue2 = "Oui"  
End If  
If CheckBox3 = False Then  
    Langue3 = "Non"  
Else  
    Langue3 = "Oui"  
End If  
CasesACocher.Hide  
Range("A3").Value = Langue1  
Range("B3").Value = Langue2  
Range("C3").Value = Langue3  
Range("A1").Select  
End Sub
```

2 / Bouton Annuler

```
Private Sub Annuler_Click()  
CasesACocher.Hide  
End Sub
```

[Le résultat](#)

Boite de dialogue avec liste déroulante.

Le Userform (ListeDeroulante)

Le code

[Code lançant l'ouverture de la liste déroulante](#)

```
Sub AfficheListeDeroulante()  
ListeDeroulante.Show  
End Sub
```

[Code d'ouverture du formulaire](#)

```

Private Sub UserForm_Activate()
    DerniereMarque = Range("A1").End(xlDown).Address
    ' Plage de données pour afficher dans liste déroulante
    Marque.RowSource = "A1:" & DerniereMarque
    ' Afficher la première marque de la liste (0 = Fiat)
    Marque.ListIndex = 0
End Sub

```

[Code des boutons](#)

1 / Bouton Valider

```

Private Sub Valider_Click()
    ListeDeroulante.Hide
    ' Marque est le nom donné au contrôle ListBox
    Index = Marque.ListIndex
    ChoixMarque = Marque.List(Index)
    ' Stockage du choix effectué dans la liste déroulante en D2
    Range("D2").Value = ChoixMarque
End Sub

```

2 / Bouton Annuler

```

Private Sub Annuler_Click()
    ListeDeroulante.Hide
End Sub

```

[Les données utilisées pour remplir la liste déroulante](#)

Boite de dialogue avec liste déroulante modifiable.

Le Userform (ListeDeroulanteModifiable)

Les contrôles

```

Nom de la ComboBox = Constructeur
Nom du CommandButton "Valider Choix" = OK
Nom du CommandButton "Annuler" = Annuler

```

Le code

[Code lançant l'ouverture de la boîte de dialogue](#)

```

Sub AfficheListeDeroulanteModifiable()
    ListeDeroulanteModifiable.Show
End Sub

```

[Code d'ouverture du formulaire](#)

```

Private Sub UserForm_Activate()
    ' Utilisation de l'option AddItem
    Dim Arr() As String
    Dim I As Integer, NbSheets As Integer
    Dim Liste As Object
    Set Liste = Worksheets(1).Cells(1, 1).Resize(Worksheets("Feuil1") _
        .Cells(1, 1).CurrentRegion.Rows.Count - 1, 1)
    NbConstructeurs = Application.CountA(Range("A1").EntireColumn)
    ListeDeroulanteModifiable.Constructeur.Clear
    ReDim Arr(1 To NbConstructeurs)
    For I = 1 To NbConstructeurs
        Arr(I) = Liste(I).Value
        ListeDeroulanteModifiable.Constructeur.AddItem Arr(I)
    Next

```

```
Constructeur.ListIndex = 0
End Sub
```

[Code des boutons](#)

1 / Bouton OK

```
Private Sub OK_Click()
    ListeDeroulanteModifiable.Hide
    Application.ScreenUpdating = False
    ' Constructeur est le nom donné au contrôle ComboBox
    Marque = Constructeur.Value
    ' Stockage du choix effectué dans la liste déroulante en D2
    Range("D2").Value = Marque
    Columns("A:A").Select
    On Error GoTo Ajoute
    Selection.Find(What:=Marque, After:=ActiveCell, LookIn:=xlFormulas, _
        LookAt:=xlPart, SearchOrder:=xlByRows, SearchDirection:=xlNext, _
        MatchCase:=False).Activate
    Range("A1").Select
    Exit Sub
Ajoute:
    ' Une nouvelle marque étant ajoutée, ajout de celle-ci à la fin de la colonne A
    Range("A1").End(xlDown).Offset(1, 0).Value = Marque
    ' Tri des marques en ordre croissant
    Selection.Sort Key1:=Range("A1"), Order1:=xlAscending, Header:=xlGuess, _
        OrderCustom:=1, MatchCase:=False, Orientation:=xlTopToBottom
    ' Sélectionne la plage et recolore la plage des marques après ajout nouvelle marque
    With Worksheets(1)
        .Range(.A1, .A1.End(xlDown)).Select
    End With
    With Selection.Interior
        .ColorIndex = 20
        .PatternColorIndex = xlAutomatic
    End With
    Range("A1").Select
End Sub
```

2 / Bouton Annuler

```
Private Sub Annuler_Click()
    ListeDeroulanteModifiable.Hide
End Sub
```

[Les données utilisées pour remplir la liste déroulante](#)

Boite de dialogue avec liste déroulante modifiable.

Le Userform (ListeDeroulanteModifiable)

Les contrôles

Nom de la ComboBox = Constructeur
Nom du CommandButton "Valider Choix" = OK
Nom du CommandButton "Annuler" = Annuler

Le code

[Code lançant l'ouverture de la boîte de dialogue](#)

```
Sub AfficheListeDeroulanteModifiable()
    ListeDeroulanteModifiable.Show
End Sub
```

[Code d'initialisation du formulaire](#)

```
Private Sub UserForm_Initialize()
    Me.Constructeur.RowSource = "Feuil1!A1:A" & Sheets("Feuil1").Cells(1, 1).End(xlDown).Row
End Sub
```

Code d'activation du formulaire

```
Private Sub UserForm_Activate()
    Me.Constructeur.ListIndex = -1
End Sub
```

Code des boutons

1 / Bouton OK

```
Private Sub OK_Click()
    Me.Hide
    If Me.Constructeur.ListIndex = -1 Then
        Sheets("Feuil1").Cells(1, 1).End(xlDown).Offset(1, 0).Value = Me.Constructeur.Value
        ' Tri de la plage de données suite à l'ajout d'un élément
        Sheets("Feuil1").Range([A1], [A1].End(xlDown)).Select
        Selection.Sort Key1:=Range("A1"), Order1:=xlAscending, Header:=xlGuess, _
            OrderCustom:=1, MatchCase:=False, Orientation:=xlTopToBottom
    Else
        End If
        ' Mise en place de la valeur choisie en D2
        Sheets("Feuil1").Cells(2, 4).Value = Me.Constructeur.Value
    End Sub
```

2 / Bouton Annuler

```
Private Sub Annuler_Click()
    ListeDeroulanteModifiable.Hide
End Sub
```

Les données utilisées pour remplir la liste déroulante

Boite de dialogue avec sélection multiple dans liste déroulante.

Le Userform (DlgChoixMultiple)

Les contrôles

```

N
om
de la
ListB
ox =
Liste
Client
s
N
om
du
Com
mand
Butto
n
"OK"
= OK
N
om
du
Com
mand
Butto
n
"Ann
uler"
=
```

Annul
er

Le code

[Code lançant l'ouverture de la boîte de dialogue](#)

```
Sub AfficheListeDeroulanteMultiSelect()  
    DlgChoixMultiple.Show  
End Sub
```

[Code d'initialisation du formulaire](#)

```
Private Sub UserForm_Initialize()  
' Les données proviennent de la feuille Feuil1 et sont dans les colonnes A et B  
    With Sheets("Feuil1").Range("A1")  
        Me.ListeClients.RowSource = Range(.Cells, .End(xlDown)(1, 2)).Address(External:=True)  
    End With  
End Sub
```

[Code d'activation du formulaire](#)

```
Private Sub UserForm_Activate()  
' Effacement des données de la plage qui reçoit les items sélectionnés (colonnes D et E)  
    Range("D1:E1", Cells(ListeClients.ListCount, 4)).Clear  
' Propriété Multiselect du contrôle ListeClients mis sur fmMultiSelectExtended  
    ListeClients.MultiSelect = fmMultiSelectExtended  
End Sub
```

[Code des boutons](#)

1 / Bouton OK

```
Private Sub OK_Click()  
    DlgChoixMultiple.Hide  
    j = 1  
    With ListeClients  
        For i = 0 To .ListCount - 1  
            If .Selected(i) = True Then  
' Le résultat est placé dans les colonnes D et E de la feuille Feuil1  
                Cells(j, 4) = .List(i, 0): Cells(j, 5) = .List(i, 1)  
                .Selected(i) = False  
                j = j + 1  
            End If  
        Next  
    End With  
End Sub
```

2 / Bouton Annuler

```
Private Sub Annuler_Click()  
    DlgChoixMultiple.Hide  
End Sub
```

[Le résultat par rapport à la sélection dans le formulaire \(voir image du Userform au début de la page\)](#)

Boite de dialogue avec liste déroulante à 2 colonnes.

Le Userform (DlgChoix)

Les contrôles

```

    N
om
de la
Com
boBo
x =
Liste
Client
s
    N
om
du
Com
mand
Butto
n
"OK"
= OK
    N
om
du
Com
mand
Butto
n
"Ann
uler"
=
Annul
er

```

Quelques propriétés du contrôle ListeClients

ColumnCount = 2 (Nbre de colonne à afficher dans une zone de liste)
 ColumnWidths = 40 (Largeur de chaque colonne d'une zone de liste à plusieurs colonnes)
 Height = 18 (Hauteur en points d'un objet)
 ListWidth = 190 (Largeur de la liste d'un contrôle ComboBox)
 Width = 78 (largeur en points d'un objet)

Le code

[Code lançant l'ouverture de la boîte de dialogue](#)

```

Sub AfficheListeDeroulante()
    DlgChoix.Show
End Sub

```

[Code d'initialisation du formulaire](#)

```

Private Sub UserForm_Initialize()
' Les données proviennent de la feuille Feuil1 et sont dans les colonnes A et B
    With Sheets("Feuil1").Range("A1")
        Me.ListeClients.RowSource = Range(.Cells, .End(xlDown)(1, 2)).Address(External:=True)
    End With
End Sub

```

[Code d'activation du formulaire](#)

```

Private Sub UserForm_Activate()
' Effacement des données de la plage qui reçoit les items sélectionnés (colonnes D et E)
    Range("D1:E1", Cells(ListeClients.ListCount, 4)).Clear
End Sub

```

[Code des boutons](#)

1 / Bouton OK

```

Private Sub OK_Click()
    DlgChoix.Hide

```

```

    If ListeClients.ListIndex <> -1 Then
' Récupération du code et du nom du client
        Sheets("Feuil1").Cells(1, 4).Value = ListeClients.Column(0, ListeClients.ListIndex)
        Sheets("Feuil1").Cells(1, 5) = ListeClients.Column(1, ListeClients.ListIndex)
    End If
End Sub

```

2 / Bouton Annuler

```

Private Sub Annuler_Click()
    DlgChoix.Hide
End Sub

```

[Le résultat par rapport à la sélection dans le formulaire \(voir image du Userform au début de la page\)](#)

Boite de dialogue combinée.

Cette boîte de dialogue permet de mettre à jour les deux contrôles de droite (TextBox et ComboBox) suivant la sélection effectuée dans le contrôle de gauche (ListBox) et cela de manière dynamique.

Le Userform (ListeDeroulanteCombinee)

Le code

[Code lançant l'ouverture du Userform \(ListeDeroulanteCombinee\)](#)

```

Sub
Affich
eList
eDer
oulan
teCo
mbin
ee()
S
heets
("Acc
ueil").
Activ
ate
R
ange(
"B2").
Value
= ""
R
ange(
"C2")
.Value
= ""
R
ange(
"D2")
.Value
= ""
A
pplic
ation.
Scre
enUp
datin
g =
False
Li
steD
eroul
ante
Com
binee

```

```
.Sho  
w  
End  
Sub
```

Code d'ouverture du formulaire par l'événement Activate

```
Private Sub UserForm_Activate()  
Dim Arr() As String  
Dim I As Integer, NbSheets As Integer  
Dim Liste As Object  
' Suppression des entrées de la liste si celle-ci en contient  
If Onglets.ListCount >= 1 Then  
Dim ElementListe As Integer  
Dim NbreIt As Integer  
    NbreIt = Onglets.ListCount - 1  
    For ElementListe = NbreIt To 0 Step -1  
        Onglets.RemoveItem (ElementListe)  
    Next ElementListe  
End If  
' Ajout de toutes les entrées de la liste  
NbSheets = Worksheets.Count  
ReDim Arr(1 To NbSheets)  
For I = 2 To NbSheets  
    Arr(I) = Worksheets(I).Name  
    ListeDeroulanteCombinee.Onglets.AddItem Arr(I)  
Next I  
Onglets.ListIndex = 0  
End Sub
```

Code sur l'événement Change de la liste déroulante Onglets (Contrôle situé à gauche)

```
Private Sub Onglets_Change()  
Dim OngletSelect As Integer  
' Déterminer la ligne sélectionnée dans la liste déroulante  
OngletSelect = ListeDeroulanteCombinee.Onglets.ListIndex + 2  
' Mise à jour TextBox Catégorie (contrôle en haut et à droite)  
ListeDeroulanteCombinee.Categorie = Sheets(OngletSelect).Range("C1").Value  
' Mise à jour ComboBox Marque (contrôle en bas et à droite)  
Dim DerniereMarque As String  
    Sheets(OngletSelect).Activate  
    DerniereMarque = Range("A1").End(xlDown).Address  
    Marque.RowSource = "A1:" & DerniereMarque  
    Marque.ListIndex = 0  
End Sub
```

Code des boutons

1 / Bouton Valider

```
Private Sub Valider_Click()  
    ListeDeroulanteCombinee.Hide  
' Récupération des index en fonction de la sélection  
    IndexOnglet = Onglets.ListIndex  
    IndexModele = Marque.ListIndex  
' Récupération des valeurs en fonction des index  
    Onglet = Onglets.List(IndexOnglet)  
    Modele = Marque.List(IndexModele)  
    Rubrique = Categorie.Value  
' Mise en place des valeurs dans la feuille de calcul  
    Sheets("Accueil").Activate  
    Range("B2").Value = Onglet  
    Range("C2").Value = Rubrique  
    Range("D2").Value = Modele  
End Sub
```

2 / Bouton Annuler

```
Private Sub Annuler_Click()  
    ListeDeroulanteCombinee.Hide  
End Sub
```

Les données utilisées pour remplir la liste déroulante

Onglet Liste Voitures

Onglet Liste Motos

Onglet Liste Avions

Un exemple de sélection avec le résultat inscrit dans la feuille de calcul

Boite de dialogue combinée.

Cette boîte de dialogue permet en cochant une des options du contrôle de gauche (Catégorie), d'afficher la liste correspondante dans le contrôle de droite (Liste avec Noms et Prénoms)

Le Userform (ListeCombinee)

Le code

Code lançant l'ouverture du Userform (ListeCombinee)

```
Sub AfficheListeCombinee()  
    ListeCombinee.Show  
End Sub
```

Code d'ouverture du formulaire par l'événement Activate

Code sur l'événement Click des boutons d'option de la zone

Catégorie

(Contrôle situé à gauche)

```
Private Sub UserForm_Activate()  
    ' Perm et de coch er le bouton radio 1 à l'ouverture du formulaire  
    Private Sub OptionButton1_Click()  
        LastInputRow = Cells(1, 1).End(xlDown).Row  
        ListBox1.RowSource = "Effectif!A1: A" & LastInputRow  
        MyCategorie = "Débutants"  
        Me.ListBox1.ListIndex = 0  
        ' Mise à jour libellé liste déroulante  
        With ListeCombinee  
            .Label1.Caption = "Liste des " & MyCategorie & " (Nom et Prénom)"  
        End With  
    End Sub  
  
    Private Sub OptionButton2_Click()  
        LastInputRow = Cells(1, 2).End(xlDown).Row  
        ListBox1.RowSource = "Effectif!B1: B" & LastInputRow  
        MyCategorie = "Poussins"  
        Me.ListBox1.ListIndex = 0  
        ' Mise à jour libellé liste déroulante  
        With ListeCombinee  
            .Label1.Caption = "Liste des " & MyCategorie & " (Nom et Prénom)"  
        End With  
    End Sub  
  
    Private Sub OptionButton3_Click()  
        LastInputRow = Cells(1, 3).End(xlDown).Row  
        ListBox1.RowSource = "Effectif!C1: C" & LastInputRow  
        MyCategorie = "Benjamins"  
        Me.ListBox1.ListIndex = 0  
        ' Mise à jour libellé liste déroulante
```

```

With ListeCombinee
.Label1.Caption = "Liste des " & MyCategorie & " (Nom et Prénom)"
End With
End Sub

```

[Code des boutons](#)

1 / Bouton Valider

```

Private Sub CmdValider_Click()
Me.Hide
MySelection = Me.ListBox1.List(ListBox1.ListIndex)
MsgBox "Vous avez choisi le joueur " & MySelection & "." & vbCrLf & _
    "Il appartient à la catégorie des " & MyCategorie & "."
Unload Me
End Sub

```

2 / Bouton Annuler

```

Private Sub Annuler_Click()
Me.Hide
Unload Me
End Sub

```

Boite de dialogue avec 2 ListBox synchronisées.

Sur une idée de [Ole P. Erlandsen](#).

Cette boîte de dialogue permet en sélectionnant un élément dans la liste de gauche, d'afficher les éléments correspondants dans la liste de droite.

Le Userform (ListeCombinee)

Le code

[Code lançant l'ouverture du Userform \(DoubleListBox\)](#)

```

Sub
Affich
eList
Box()

Load
Doub
leList
Box

Doub
leList
Box.
Show

Unlo
ad
Doub
leList
Box
End
Sub

```

[Code d'ouverture du formulaire par l'événement Initialize](#)

Privat

```

e
Sub
User
Form
_Initiali
alize(
)
' La
mise
à jour
des
items
dans
ListB
ox1 ..
;
mettr
a à
jour
les
items
dans
la
ListB
ox2

With
Me

    U
pdate
ListB
ox
.ListB
ox1,
-1

End
With
End
Sub

```

[Code sur l'événement Change de la ListBox1 \(Contrôle situé à gauche\)](#)

```

Private Sub ListBox1_Change()
' Mise à jour des items dans la ListBox2
    UpdateListBox Me.ListBox2, Me.ListBox1.ListIndex
End Sub

```

```

Private Sub UpdateListBox(Parametres As MSForms.ListBox, IndexValue As Integer)
Dim LastInputRow As Integer, ColumnIndex As Integer, InputRange As Range
' Les données commencent à la ligne 3
Const FirstInputRow As Integer = 3
' Détermine depuis quelle colonne on prend la liste des items
    ColumnIndex = IndexValue + 2
' Détermine la dernière ligne de la colonne sélectionnée et la plage correspondante
    LastInputRow = Cells(FirstInputRow, ColumnIndex).End(xlDown).Row
    Set InputRange = ActiveSheet.Range(Cells(FirstInputRow, ColumnIndex), Cells(LastInputRow, ColumnIndex))
    With Parametres
        .ColumnHeads = True ' Affiche les en-têtes de colonne
        .RowSource = InputRange.Address ' Spécifie la source de données
        .ListIndex = 0 ' Sélectionne le premier item
    End With
    Set InputRange = Nothing
End Sub

```

[Code du bouton valider](#)

1 / Bouton Valider

```

Private Sub CmdValider_Click()
    Me.Hide
    MsgBox "Dans la catégorie : " & Me.ListBox1.List(Me.ListBox1.ListIndex) & Chr(13) & Chr(13) & _
        "Vous avez choisi : " & Me.ListBox2.List(Me.ListBox2.ListIndex), vbInformation, "Résultat de votre choix : "
End Sub

```

[La feuille de données](#)

Boîte de dialogue avec un contrôle TabStrip (contrôle Onglet).

[Quelques infos à propos du Contrôle TabStrip](#)

Un contrôle TabStrip est un contrôle contenant une collection d'un ou de plusieurs onglets.

Dans les contrôles TabStrip, ceux-ci sont indépendants des pages.

En réalité un clic sur un onglet n'affiche pas de nouvelle page.

Pour les contrôles TabStrip, la propriété Value permet de déterminer ou de définir quel onglet est actif.

Cet exemple permet en cliquant sur l'un des onglets, d'avoir instantanément les chiffres correspondants à l'activité indiquée sur l'onglet.

Le Userform (Avec 3 onglets)

Le code

[Code lançant l'ouverture de la boîte de dialogue](#)

```
Sub AfficheTabStrip()  
    UserForm1.Show  
End Sub
```

[Code d'ouverture du formulaire](#)

```
Private  
Sub  
User  
Form  
_Activate()  
  
Me.TabStrip1.Value = 0  
For I = 1 To 5  
    Me.Controls("TextB" & I).Value = Sheets("Résultats").Cells(I + 1, 2).Value  
Next I  
End Sub
```

Code sur l'événement Change

```
Private  
Sub  
TabStrip1_  
Change()  
End Sub
```

```
Select  
Case  
TabStrip1.V  
alue
```

```
Case 0
```

```
For  
I =  
1 To  
5
```

```
Me.  
Controls("T  
extBox1").  
Value =  
Sheets("R  
ésultats").  
Cells(  
I + 1,  
2).Val  
ue
```

```
Next I
```

```
Case 1
```

```
For  
I =  
1 To  
5
```

```
Me.  
Controls("T  
extBox1").  
Value =  
Sheets("R  
ésultats").  
Cells(  
I + 1,  
3).Val  
ue
```

```
Next I
```

```
Case 2
```

```
For  
I =
```

```
1 To  
5
```

```
Me.  
Contr  
ols("T  
extBo  
x" &  
I).Val  
ue =  
Shee  
ts("R  
ésult  
ats").  
Cells(  
I + 1,  
4).Val  
ue
```

```
Ne  
xt I
```

```
End  
Selec  
t  
End  
Sub
```

[Code du bouton Fermer](#)

Bouton Fermer

```
Privat  
e  
Sub  
Cmd  
Ferm  
er_Cl  
ick()
```

```
Me.H  
ide  
End  
Sub
```

[Les données de la feuille Résultats utilisées pour remplir le UserForm.](#)

Les contrôles courants

Exemples d'utilisation de InputBox

Page mise à jour le : 19/03/2003

InputBox peut être utilisée soit comme fonction, soit comme méthode.

La **fonction Inputbox** permet d'afficher une boîte de dialogue contenant une zone de texte légendée, invitant l'utilisateur à y saisir l'information attendue. L'information saisie est renvoyée sous forme de chaîne de caractères et stockée dans une variable afin de pouvoir être ensuite exploitée par le programme.

L'objet **Application** d'Excel possède une méthode InputBox que l'on peut substituer à la fonction InputBox de Visual Basic.

La **méthode Inputbox** se présente comme la fonction InputBox mais offre une possibilité très intéressante : elle permet de spécifier le type de données qui sera renvoyée (numérique, plage).

Saisie d'information via une InputBox (fonction).

Saisie d'information (valeur numérique) via une InputBox (méthode).

Saisie d'information (plage de cellules) via une InputBox (méthode).

Saisie d'information via une InputBox (fonction)

Cette boîte de dialogue avec une zone de texte légendée, invite l'utilisateur à y saisir l'information attendue. L'information saisie est renvoyée sous forme de chaîne de caractères et stockée dans une variable afin de pouvoir être ensuite exploitée par le programme.

La boîte de dialogue

Le code

[Code lançant l'ouverture de la boîte de dialogue InputBox](#)

```
Sub  
Utilis  
ation  
DeIn  
putbo  
xFon  
ction(  
)  
D  
im  
Inscri  
ption  
As  
Strin  
g  
In  
script  
ion =  
Input  
Box("Nom  
de  
l'adh  
érent  
:",  
"Nou  
veau  
mem  
bre")  
  
Si  
l'utilis  
ateur  
n'a  
saisie  
aucu  
ne  
donn  
ée et  
qu'il  
cliqu
```

```

e sur
OK,
,
ou si
l'utilis
ateur
cliqu
e sur
Annul
er, on
quitte
la
procé
dure.
If
Inscri
ption
= ""
Then

Msg
Box
"Auc
une
donn
ée
n'a
été
saisie
"

Exit
Sub
,
Sinon
la
donn
ée
saisie
est
affich
ée
El
se

Msg
Box
Inscri
ption
E
nd If
End
Sub

```

Saisie d'information (valeur numérique) via une InputBox (méthode).

Cette boîte de dialogue avec une zone de texte légendée, invite l'utilisateur à y saisir l'information attendue sous forme numérique (`type:=1`). L'information saisie est renvoyée sous forme de nombre et stockée dans une variable afin de pouvoir être ensuite exploitée par le programme.

La boîte de dialogue

Le code

[Code lançant l'ouverture de la boîte de dialogue InputBox](#)

```

Sub
Utilis
ation
Deln
putbo
xMet
hode(
)
Dim
Rece
tte As
Integ
er

```

```

Rece
tte =
Appli
catio
n.Inp
utBox
("Rec
ette
effect
uée :
",
"Saisi
e de
la
recett
e",
Type:
=1)
' Si
l'utilis
ateur
a
cliqu
é sur
Annul
er ...
If
Rece
tte =
False
Then
Exit
Sub
'
sinon
la
donn
ée
saisie
est
affich
ée

Msg
Box
"La
recett
e est
de : "
&
Rece
tte
End
Sub

```

Saisie d'information (plage de cellules) via une InputBox (méthode).

Cette boîte de dialogue avec une zone de texte légendée, invite l'utilisateur à y saisir l'information attendue sous forme de sélection de plage (`type:=8`). L'information saisie est renvoyée sous forme de plage et stockée dans une variable afin de pouvoir être ensuite exploitée par le programme.

La boîte de dialogue

Le code

[Code lançant l'ouverture de la boîte de dialogue InputBox](#)

```

Sub UtilisationDeInputboxMethode()
Dim MaPlage As Range
On Error Resume Next
Set MaPlage = Application.InputBox(prompt:="Sélectionnez une plage de cellules.", _
Title:="Sélection d'une plage", Left:=5, Top:=5, Type:=8)
' Si l'utilisateur a cliqué sur le bouton Annuler ...
If Err.Number = 424 Then
MsgBox "Vous avez choisi d'annuler"
End
Else
' sinon la plage sélectionnée va être affichée

```

```
MsgBox "La plage sélectionnée est : " & MaPlage.Address
End If
End Sub
```

Pour plus d'informations, consultez l'aide sur [InputBox](#).

Exemples de la fonction MsgBox

Page mise à jour le : 02/11/2000

Vous trouverez dans cette rubrique des exemples de la fonction MsgBox. La fonction MsgBox permet d'afficher une boîte de dialogue présentant un message et des boutons de commande, afin d'afficher une information à l'attention de l'utilisateur. ou d'obtenir une réponse à une question qui orientera l'exécution du programme.

Message d'information via une MsgBox.

Message d'information via une MsgBox avec choix de l'action à effectuer (2 boutons).

Message d'information via une MsgBox avec choix de l'action à effectuer (3 boutons).

Message d'information via une MsgBox.

Cette boîte de dialogue affiche un message, attend que l'utilisateur clique sur le bouton pour valider la lecture de ce message.

La boîte de dialogue

Le code

[Code lançant l'ouverture de la boîte de dialogue MsgBox](#)

```
Sub MessageInformation()
' Affiche un message d'information
Msg = "Votre fichier a été sauvegardé."
Title = "Démonstration de MsgBox - E. Renaud 10/2000"
Style = vbOKOnly + vbInformation
Reponse = MsgBox(Msg, Style, Title)
End Sub
```

Message d'information via une MsgBox avec choix de l'action à effectuer.

Cette boîte de dialogue affiche un message, attend que l'utilisateur clique sur un bouton, puis renvoie une valeur de type Integer qui indique le bouton choisi par l'utilisateur.

La boîte de dialogue

Le code

[Code lançant l'ouverture de la boîte de dialogue MsgBox](#)

```
Sub
Recu
plInfo
Mess
age()
Dim
Msg,
Style,
Title,
Resp
onse
'
Défini
t le
mess
```

```
age.  
Msg  
=  
"Sou  
haite  
Z-  
vous  
conti  
nuer?"  
"
```

```
'  
Défini  
t les  
bouto  
ns.  
Style  
=  
vbYe  
sNo  
+  
vbQu  
estio  
n  
,
```

```
Défini  
t le  
titre.  
Title  
=  
"Dém  
onstr  
ation  
de  
MsgB  
ox -  
E.  
Rena  
ud  
1/200  
0"  
,
```

```
'  
Affich  
e le  
mess  
age.  
Resp  
onse  
=  
MsgB  
ox(M  
sg,  
Style,  
Title)  
If  
Resp  
onse  
=  
vbYe  
s  
Then  
,
```

```
L'utili  
sateu  
r a  
choisi  
Oui.
```

```
MySt  
ring =  
"Oui"  
,
```

```
Effect  
ue  
une  
actio  
n.  
Else '  
L'utili  
sateu  
r a  
choisi  
Non.
```

```
MySt  
ring =  
"Non"
```

```
,  
Effect  
ue  
une  
actio  
n.  
End  
If  
End  
Sub
```

Message d'information via une MsgBox avec choix de l'action à effectuer.

Cette boîte de dialogue affiche un message, attend que l'utilisateur clique sur un bouton, puis renvoie une valeur de type Integer qui indique le bouton choisi par l'utilisateur.

La boîte de dialogue

Le code

[Code lançant l'ouverture de la boîte de dialogue MsgBox](#)

```
Sub  
Mess  
ageA  
vecR  
epon  
seOu  
Annul  
ation(  
)  
,  
Affich  
e un  
mess  
age  
d'info  
rmati  
on  
avec  
test  
sur le  
bouto  
n  
sélec  
tionn  
é  
M  
sg =  
"Dém  
onstr  
ation  
avec  
3  
bouto  
ns."  
Ti  
tle =  
"Exe  
mple  
de  
boîte  
de  
mess  
age"  
S  
tyle =  
vbYe  
sNoC  
ancel  
+  
vbQu  
estio  
n  
R  
epon  
se =
```

```
MsgBox(Msg, Style, Title)
'Si le bouton Annuler a été choisi ... on quitte la procédure
If Response = vbCancel Then Exit Sub
'sinon on teste lequel du bouton Oui ou Non a été choisi
If Response = vbsYes Then
    MsgBox "Vous avez sélectionné Oui"
Else
    MsgBox "Vous avez sélectionné Annuler"
End If
End Sub
```

[Pour plus d'informations, consultez l'aide sur MsgBox.](#)

Contrôles avancés

Le contrôle Calendar (calendrier).

Ce contrôle permet grâce à un calendrier de saisir des dates de début et fin pour les utiliser dans votre application.
Exemple trouvé dans livre Excel 2000 & Visual Basic pour Applications 6 disponible chez Amazon.fr

Ajout du contrôle Calendar

- 1 - Activer un UserForm dans le Visual Basic Editor.
- 2 - Faites un clic droit sur la boîte à outils et sélectionnez Contrôles supplémentaires.
- 3 - Dans la liste déroulante, sélectionnez Calendar Control 8.0.
- 4 - Cliquez sur OK pour fermer la boîte de dialogue et ajoutez le contrôle dans la boîte à outils.

La boîte à outils

Le Userform (fmContratDates)

Le code

Code lançant l'ouverture de la boîte de dialogue

```
Sub AfficheCalendar()  
    fmContratDates.Show  
End Sub
```

Code d'ouverture du formulaire

```
Private Sub UserForm_Initialize()  
    cboDateAValider.AddItem ("Date de début")  
    cboDateAValider.AddItem ("Date de fin")  
    cboDateAValider.ListIndex = 0  
    Calendrier.Value = Date  
    Calendrier.SetFocus  
End Sub
```

Code lié au choix des dates sur le contrôle Calendar

```
Private Sub Calendrier_Click()  
    ' Choix d'une date sur le calendrier  
    If cboDateAValider.ListIndex = 0 Then  
        DateDebut.Value = Calendrier.Value  
        cboDateAValider.ListIndex = 1  
    Else  
        DateFin.Value = Calendrier.Value  
    End If  
End Sub
```

Code des boutons

1 / Bouton Valider (cmdValider)

```
Private  
Sub  
Cmd  
Valid  
er_Cl  
ick()  
,  
Vérifi  
catio
```

```
n de
la
validit
é des
infor
matio
ns
If
Date
Debu
t.Valu
e = ""
Then
MsgB
ox
"Vou
s
deve
z
spéci
fier
une
date
de
début
.";
vbOK
Only
+
vbCri
tical,
"Cont
rat
d'aut
eur"
    E
xit
Sub
Elsel
f
Date
Fin.V
alue
= ""
Then
    M
sgBo
x
"Vou
s
deve
z
spéci
fier
une
date
de
fin.";
vbOK
Only
+
vbCri
tical,
"Cont
rat
d'aut
eur"
    E
xit
Sub
Elsel
f
Date
Value
(Date
Fin) -
Date
Value
(Date
Debu
t) <=
0
Then
    M
sgBo
x
"Vou
s
```

deve
z
spéci
fier
une
date
de
début
antéri
eure
à la
date
de
fin.",

- v
bOK
Only
+
vbCri
tical,
"Cont
rat
d'aut
eur"
E

xit
Sub
Elself
Date
Value
(Date
Fin) -
Date
Value
(Date
Debu
t) <
40
Then
Dim
Conti
nuer
As
Integ
er
Conti
nuer
=
MsgB
ox("V
ous
avez
indiq
ué
une
date
de
début
à
seule
ment
"

- &
Date
Value
(Date
Fin) -
Date
Value
(Date
Debu
t) & "
jours
de la
date
de
fin.

- V
alider
cette
date
?",
vbYe
sNo
+

```

vbQu
estio
n,
"Valid
er la
date
de
début
?")
If
Conti
nuer
=
vbNo
Then
Exit
Sub
End
If
fmCo
ntrat
Date
s.Hide
.
Affich
age
des
dates
sélec
tionn
ées
MsgB
ox ("
Les
dates
de
Débu
t et
Fin
sont :
" &
Date
Debu
t.Valu
e & "
et " &
Date
Fin.V
alue)
.
Mise
à jour
des
para
mètre
s
Date
Debu
t.Valu
e = ""
Date
Fin.V
alue
= ""
cboD
ateA
Valid
er.Lis
tInde
x = 0
Cale
ndrier
.Valu
e =
Date
Cale
ndrier
.SetF
ocus
End
Sub

```

2 / Bouton Annuler (cmdAnnuler)

```

Private
Sub
Cmd
Annul
er_Click()
Dim
Rep
As
Byte
Rep
=
MsgBox("Etes-vous sûr de vouloir fermer l'application en cours ?", _vbYesNo + vbQuestion, "Annuler l'application en cours ?")
If
Rep
=
vbNo
Then
Exit
Sub
End
If
Me.Hide
End
Sub

```

3 / Bouton Retour (cmdRetour)

```

Private
Sub
Cmd
Retour_Click()
MsgBox("Vous allez fermer l'application"
Me.Hide
End
Sub

```

Le contrôle ComboBox.

Allie les caractéristiques d'un contrôle **ListBox** et d'un contrôle **TextBox**.
L'utilisateur peut entrer une [nouvelle valeur](#), comme dans un contrôle TextBox ou bien sélectionner une valeur existante comme dans un contrôle **ListBox**.
La liste d'un contrôle ComboBox se compose de lignes de données.
La propriété par défaut d'un contrôle ComboBox est la propriété **Value**.

Le principe de fonctionnement de ce contrôle, se décompose comme suit :

- 1 - Remplir le contrôle ComboBox avec des données.
- 2 - Choisir une de ces données en la sélectionnant.
- 3 - Récupérer la donnée sélectionnée à la fermeture du formulaire.

Remplir le contrôle Combobox avec des données

Exemple : Afficher une liste de Domaines viticoles.

Je vous donne 4 méthodes pour remplir cette ComboBox (en fonction de votre besoin)

La propriété (name) du contrôle ComboBox est [cboComboBox](#)

Méthode 1 : Les données sont dans une feuille de calcul (Feuil1).

```
Privat
e
Sub
User
Form
_Initiali
alize(
)
M
e.cbo
Com
boBo
x.Ro
wSou
rce =
"Feuil
1
A1:A"
&
Shee
ts("F
euil1"
).Cell
s(1,
1),En
d(xID
own).
Row
End
Sub
```

Méthode 2 : Les données sont figées et vous les ajoutez grâce à la méthode AddItem.

```
Private Sub UserForm_Initialize()
With Me.cboComboBox
cboComboBox.AddItem "Domaine Alexis Rouge"
cboComboBox.AddItem "Domaine du Grand Crès Blanc"
cboComboBox.AddItem "Domaine du Grand Crès Muscat"
cboComboBox.AddItem "Domaine du Grand Crès Rosé"

```

```

        cboComboBox.AddItem "Domaine du Grand Crès Rouge"
    End With
End Sub

```

Méthode 3 : Les données correspondent à la liste des mois (Janvier, Février, ..., Décembre).

On va utiliser la méthode AddItem avec une variable (i).

```

Private Sub UserForm_Initialize()
    Dim Mois(1 To 12) As String
    Dim i As Integer
    ' Création d'un tableau des noms de mois
    For i = 1 To 12
        Mois(i) = Format(DateSerial(1, i, 1), "mmm")
        Me.cboComboBox.AddItem Mois(i)
    Next i
End Sub

```

Méthode 4 : Les données sont figées et vous les ajouter grâce _
à la fonction VBA.Array(arglist).

```

Private
Sub
User
Form
_Initiali
ze(
)
    D
im
vaDe
partm
ent
As
Variat
nt
    D
im
stDe
ptList
() As
Strin
g
    D
im i
As
Integ
er
    v
aDep
artem
ent =
VBA.
Array
("Ain"
,
"Aisn
e",
"Allie
r","Al
pes-
de-
Hte-
Prov
ence"
,
"Alp
es-
Haut
es",
"Alpe
s-
Mariti
mes",
"Arde
nnes"
)
    R
eDim

```

```

stDe
ptList
(0 To
UBou
nd(va
Depa
rteme
nt))
    F
or i =
0 To
UBou
nd(va
Depa
rteme
nt)

    stD
eptLi
st(i) =
vaDe
parte
ment(
i)
    N
ext i
    c
boCo
mbob
ox.Li
st =
stDe
ptList
End
Sub

```

Sélectionner une données dans la liste

```

Il
suffit
simpl
emen
t de
sélec
tionn
er
avec
la
sour
is la
donn
ée
désir
ée.

```

Récupérer l'item sélectionné

```

Privat
e
Sub
cmd
OK_
Click(
)
    M
sgBo
x
"L'ite
m
sélec
tionn
é
est : "
&
cboC
ombo
Box.
Value
,
vbInf

```

Format
Union
Download
Menu
End
Sub

Récupérer une nouvelle valeur saisie

Il est possible dans le cas où les données sont issues d'une feuille Excel, d'insérer cette nouvelle valeur dans la feuille Excel.
Lorsque l'on ajoute une nouvelle donnée, celle-ci se trouve en position **ListIndex = -1**. On utilise pour cela un test sur la propriété **ListIndex**.
En règle générale, cette propriété ListIndex à la valeur **-1** à

l'ouverture
du formulaire.
Si on sélectionne la première donnée de la liste, la valeur ListIndex est égale à 0, et ainsi de suite.

```
Private Sub OK_Click()  
    Me.Hide  
    If Me.ComboBox.BoxListIndex = -1 Then  
        Sheets("Feuil1").Cells(1, 1).End(xlDown).Offset(1, 0).Value = Me.ComboBox.Value  
    End Sub
```

Le contrôle RefEdit.

Ce contrôle placé dans un formulaire affiche l'adresse d'une plage de cellules que vous avez entrée ou sélectionnée dans une feuille de calcul. Pour sélectionner une plage, cliquez sur le bouton dans le contrôle pour réduire le formulaire utilisateur, sélectionnez la plage, puis cliquez de nouveau sur le bouton dans le contrôle pour développer le formulaire utilisateur.

Exemple d'utilisation du contrôle RefEdit

Effectuer une opération arithmétique (+, -, * ou x) sur une plage de cellules.
(Ex. Conversion Frs/Euros, majorer un prix, appliquer une remise).

Principe d'utilisation du contrôle RefEdit

Etape 1 : Cliquez sur le trait horizontal du contrôle.

Etape 2 : Affichage d'une boîte de sélection.

Etape 3 : Sélectionnez avec la souris la plage désirée.

Etape 4 : Cliquez sur la flèche rouge du contrôle pour valider votre sélection.

Mise en oeuvre du contrôle RefEdit

- 1 - Activer un UserForm dans le Visual Basic Editor.
- 2 - Faites un clic droit sur la boîte à outils et sélectionnez Contrôles supplémentaires.
- 3 - Dans la liste déroulante, sélectionnez RefEdit.Ctrl.
- 4 - Cliquez sur OK pour fermer la boîte de dialogue et ajoutez le contrôle dans la boîte à outils.

Ajout du contrôle RefEdit

La boîte à outils

Le Userform (SelectOperation)

Le code

Code lançant l'ouverture de la boîte de dialogue

```
Sub  
Affich  
eRef  
Edit()  
S  
elect  
Oper  
ation.  
Show  
  
End  
Sub
```

Code d'ouverture du formulaire

```
Private Sub UserForm_Initialize()  
' Affiche la sélection en cours  
    RefEdit1.Text = ActiveWindow.RangeSelection.Address  
' Rend le bouton Addition actif par défaut  
    Frame1.OptionAddition.Value = True  
End Sub
```

Code des boutons

1 / Bouton Valider (CmdValider)

```
Private
Sub
Cmd
Valid
er_Click()
'
Teste
si
une
valeur
a
été
saisie
dans
la
TextBox
(Opération)

On
Error
Resume
Next
If
TextBox1.Value
= ""
Then

    MsgBox
    "Vous
    devez
    saisir
    une
    valeur
    d'opération"

    RefEdit1.Set
    Focus

    On
    Error
    GoTo
    0

    Exit
    Sub

End
If

Set
Plage
Sélection =
Range(RefEdit1.
Text)

MyOpération =
TextBox1.V
```

```
alue  
,  
Teste  
la  
sélec  
ectio  
n  
effect  
uée  
est  
corre  
ct  
I  
f Err  
<> 0  
Then
```

```
    M  
    sgBo  
    x "La  
    plage  
    sélec  
    tionn  
    ée  
    est  
    invali  
    de"
```

```
    R  
    efEdit  
    1.Set  
    Focu  
    s
```

```
    O  
    n  
    Error  
    GoTo  
    0
```

```
    E  
    xit  
    Sub
```

```
End  
If  
,
```

```
Effect  
ue  
l'opér  
ation  
choisi  
e sur  
la  
plage  
de  
cellul  
es  
sélec  
tionn  
ées
```

```
For  
Each  
Cell  
In  
Plage  
Selec  
t
```

```
    If  
    Optio  
    nAddi  
    tion.V  
    alue  
    =  
    True  
    Then  
    Cell.  
    Value  
    =  
    Cell.  
    Value  
    +  
    MyO  
    peran
```

de

```
    If  
    OptionSou  
    straction.V  
    alue  
    =  
    True  
    Then  
    Cell.  
    Value  
    =  
    Cell.  
    Value  
    -  
    MyO  
    peran  
    de
```

```
    If  
    OptionMult  
    iplicat  
    ion.V  
    alue  
    =  
    True  
    Then  
    Cell.  
    Value  
    =  
    Cell.  
    Value  
    *  
    MyO  
    peran  
    de
```

```
    If  
    OptionDivi  
    sion.  
    Value  
    =  
    True  
    Then  
    Cell.  
    Value  
    =  
    Cell.  
    Value  
    /  
    MyO  
    peran  
    de
```

Next
Cell

Selec
tOper
ation.
Hide

Unlo
ad
Selec
tOper
ation
End
Sub

2 / Bouton Annuler (CmdAnnuler)

```
Privat  
e  
Sub  
Cmd  
Annul  
er_Cl  
ick()
```

Selec
tOper
ation.
Hide

Unlo
ad
Selec
tOper
ation
End
Sub

Le contrôle TextBox (Zone de texte).

Ce contrôle placé dans un Userform, permet la saisie de données.

Ces données saisies peuvent parfois être erronées (erreur de frappe, texte à la place de nombre, date non conforme, etc ...).

Pour cela il peut être intéressant de contrôler soit lors de la validation par OK ou lors de la

sortie du contrôle si les données saisies sont conformes par rapport à la fonction du contrôle.

Convention d'écriture

Le Userform est nommé **UserForm1**

Le contrôle TextBox est nommé **TextBox1**

Le bouton OK est nommé **CmdOK**

Exemples de tests sur une TextBox

S'assurer qu'il y a eu une saisie dans la TextBox

S'assurer que la saisie dans la TextBox est de type numérique (nombre)

Le contrôle ProgressBar, version 6.0.

Ce contrôle permet d'afficher par exemple, la progression d'une mise à jour de cellules (ex. multiplier une plage par un coefficient).

Exemple trouvé sur le site de [Ole P. Erlandsen](#)

- 1 - Activer un UserForm dans le Visual Basic Editor.
- 2 - Faites un clic droit sur la boîte à outils et sélectionnez Contrôles supplémentaires.
- 3 - Dans la liste déroulante, sélectionnez Microsoft ProgressBar Control, version 6.0.
- 4 - Cliquez sur OK pour fermer la boîte de dialogue et ajoutez le contrôle dans la boîte à outils.

Ajout du contrôle ProgressBar Control

La boîte à outils

Le Userform (frmProgressBar)

Le code

[Code lançant l'indicateur de progression](#)

```
Sub  
Start  
Processin  
g1()  
,  
Displ  
ays a  
progr  
ess  
bar  
while  
a  
macr  
o  
runs,  
requir  
es a  
refer  
ence  
to  
MSC  
OMC  
TRL.  
OCX  
Dim  
lngTo  
tal As  
Long,  
lngI  
As  
Long  
,  
,  
Initiat  
e  
Progr  
essB  
ar  
Load  
frmPr  
ogres  
sBar  
With  
frmPr  
ogres  
sBar  
. .  
Progr  
essB  
ar.Sc  
rollin  
g =  
ccScr  
olling  
Stan  
dard '  
or  
ccScr  
olling  
Smo  
oth  
. .  
Show  
' set  
the  
User  
Form  
s  
Show  
Moda  
l  
prope  
rty to  
false  
befor
```

```
e
runni
ng ,

or
.Sho
w
False
End
With
Upda
tePro
gress
Bar
0,
"Proc
essin
g..."
set
initial
progr
ess
statu
s
' start
the
proce
ss
IngTo
tal =
2000
For
IngI =
1 To
IngTo
tal
    If
    IngI
Mod
50 =
0
    Then
    .
    Upda
te the
Progr
essB
ar for
every
50th
loop
        U
pdate
Progr
essB
ar
    IngI /
    IngTo
tal *
    100,
    "Proc
essin
g " &
    Form
at _
    (I
ngI /
    IngTo
tal,
    "0%")
    & "... "
    End
    If
    ' Do
    some
    thing,
    place
    your
    code
    here
    Rang
e("D1
").For
mula
=
Form
```

```
at(Ti  
me,  
"hh:  
mm:ss")  
Next  
Ingl  
Rang  
e("D1  
").Cle  
arCo  
ntent  
'  
Clea  
n up  
frmPr  
ogres  
sBar.  
Hide  
Unlo  
ad  
frmPr  
ogres  
sBar  
End  
Sub
```

[Code lié à la macro UpdateProgressBar](#)

```
Privat  
e  
Sub  
Upda  
tePro  
gress  
Bar(  
New  
Value  
As  
Singl  
e,  
Optio  
nal  
New  
Capti  
on As  
Strin  
g)  
'  
Upda  
tes  
the  
progr  
essb  
ar  
dialo  
g  
With  
frmPr  
ogres  
sBar  
If  
Not  
IsMis  
sing(  
New  
Capti  
on)  
Then  
.Capt  
ion =  
New  
Capti  
on  
'  
Progr  
essB  
ar.Val  
ue =  
New  
Value  
If
```

```
New  
Value  
= 0  
Then  
.Rep  
aint  
End  
With  
End  
End  
Sub
```

Le contrôle DTPicker (calendrier).

Ce contrôle permet grâce à un calendrier de saisir des dates de début et des dates de fin de façon simple et sans erreur possible, pour les utiliser ensuite dans votre application.

Cet exemple a été réalisé sous Excel 2000.

Aspect du contrôle

Ajout du contrôle DTPicker

- 1 - Activer un UserForm dans le Visual Basic Editor.
- 2 - Faites un clic droit sur la boîte à outils et sélectionnez Contrôles supplémentaires.
- 3 - Dans la liste déroulante, sélectionnez Microsoft Date and Time Picker Control, version 6.0
- 4 - Cliquez sur OK pour fermer la boîte de dialogue et ajoutez le contrôle dans la boîte à outils.

La boîte à outils

Le Userform (frmConges)

Le code

Code lançant l'ouverture du formulaire

```
Sub  
Affich  
eCon  
ges()  
fr  
mCo  
nges.  
Show  
End  
Sub
```

Codes d'ouverture du formulaire

```
Privat  
e  
Sub  
User  
Form  
_Initi  
alize(  
End
```

```

)
,
Définir
butée
mini
pour
contrôle
Date
début
(DTPicker1
)
Me
e.DTPicker1.MinDate =
Now
- 60
,
Définir
butée
maxi
pour
contrôle
Date
fin
(DTPicker2
)
Me
e.DTPicker2.MaxDate =
Now
+ 365
End
Sub

```

```

Private Sub UserForm_Activate()
' Remplissage des 2 listes déroulantes (Nom agent et type de congés)
Me.cboAgents.RowSource = "Congés!B7:B21"
Me.cboConges.RowSource = "Congés!B24:B31"
' Activer le premier enregistrement de chaque liste déroulante
Me.cboAgents.ListIndex = 0
Me.cboConges.ListIndex = 0
' Définir la date affichée par défaut dans chaque contrôle DTPicker
Me.DTPicker1.Value = Now
Me.DTPicker2.Value = Now
End Sub

```

[Code lié à l'événement Change sur le contrôle DTPicker2](#)

```

Private
Sub
DTPicker2
_Change()
,
Récupération
des
dates
sélectionnées
Me
yDateDebut =
DTPicker1
.Value

```

```

    M
yDateFin
=
DTPicker2
.Value
e
    If
MyDateFin <=
MyDateDebut
Then

    MsgBox
"La
date
de fin
ne
peut
être
inférieure
à la
date
de
début
"

    DTPicker2.SetFocus
E
nd If
Sub

```

[Code des boutons](#)

1 / Bouton Valider (cmdValider)

```

Private
Sub
Cmd
Valid
er_Click()
    M
yDateDebut =
DTPicker1
.Value
    M
yDateFin =
DTPicker2
.Value
    .
Dans
le
cas
où le
choix
est
effectué
sur le
DTPicker1

```

If
MyDateFin
n <= MyDateDebut
Then

MsgBox
"La date de fin ne peut être inférieure à la date de début"

DTPicker1.SetFocus
Else

Unload Me

Mise en place des données dans la feuille Base Congés

Sheets("Base Congés").Cells(65536, 1).End(xlUp).Offset(1, 0).Value = Me.cboAgents.List_

(Me.cboAgents.ListIndex)

Sheets("Base Congés").Cells(65536,

```
2).End  
d(xlU  
p).Off  
set(1,  
0).Val  
ue =  
Me.c  
boCo  
nges.  
List _
```

```
    (  
    Me.c  
    boCo  
    nges.  
    ListIn  
    dex)
```

```
    She  
    ets(""  
    Base  
    Cong  
    és").  
    Cells(  
    6553  
    6,  
    3).En  
    d(xlU  
    p).Off  
    set(1,  
    0).Val  
    ue =  
    Form  
    at(My  
    Date  
    Debu  
    t,  
    "mm/  
    dd/yy  
    yy")
```

```
    She  
    ets(""  
    Base  
    Cong  
    és").  
    Cells(  
    6553  
    6,  
    4).En  
    d(xlU  
    p).Off  
    set(1,  
    0).Val  
    ue =  
    Form  
    at(My  
    Date  
    Fin,  
    "mm/  
    dd/yy  
    yy")
```

```
    E  
    nd If  
    End  
    Sub
```

2 / Bouton Annuler (cmdAnnuler)

```
Privat  
e  
Sub  
Cmd  
Annul  
er_CI  
ck()  
    M  
e.Hid  
e  
End  
Sub
```

Quelques infos sur les UserForms

Dans cette rubrique vous trouverez des infos pour la gestion des UserForms.

Retour au menu Contrôles divers

Empêcher le déplacement d'un Userform.

Mettre hors fonction la croix (X) de fermeture du UserForm.
Minimiser un formulaire (Excel 2000 uniquement).

Ajout d'une image sur un bouton de formulaire.

Empêcher le déplacement d'un Userform, et non affichage de la croix de fermeture.

[Exe
mple
propo
sé
par
[Laure
nt
Longr
e](#)]

Le
code
qui
suit
devra
être
dans
le
modu
le de
code
du
User
Form
.

```
Private  
Declare  
Function  
Find  
WindowA  
Lib  
"User  
32" _  
(  
ByVal  
lpCla  
ssName  
As  
String,  
ByVal  
lpWin  
dowName  
As  
String
```

g) As
Long

Private
Declare
Function
GetWindowLongA
Lib
"User32" _
(
ByVal
hwnd As
Long,
ByVal
nIndex As
Long)
As
Long

Private
Declare
Function
SetWindowLongA
Lib
"User32" _
(
ByVal
hwnd As
Long,
ByVal
nIndex As
Long,

ByVal
dwNewLong As
Long)
As
Long

Private
Sub
UserForm
_Activate()
Dim
hwnd As
Long,
exLong As
Long
hWnd
= FindWindow(
vbNullString

```

g,
Me.C
aption)
    e
xLon
g =
GetW
indo
wLon
gA(h
Wnd,
-16)
    If
exLo
ng
And
&H88
0000
Then

    Set
Wind
owLo
ngA
hWn
d,
-16,
exLo
ng
And
&HF
F77F
FFF

    Me.
Hide:
Me.S
how
E
nd If
End
Sub

```

Mettre hors fonction la croix (X) de fermeture du UserForm.

```

Privat
e
Sub
User
Form
_Que
ryClo
se(C
ancel
As
Integ
er,
Close
Mode
As
Integ
er)
    If
Close
Mode
=
vbFor
mCo
ntrol
Menu
Then

    Msg
Box
"Vous
ne
pouv
ez
pas
utilise
r ce
bouto

```

n de
ferme
ture."

```
Cancel =  
True  
End If  
End  
Sub
```

Minimiser un UserForm.

[Exe
mple
propo
sé
par
[Laure
nt
Longr
e](#)]

Mettr
e la
propri
été
**Show
Modal**
du
User
Form
sur
**Fals
e**
(cette
propri
été
n'exis
te
pas
sous
Excel
97)

Défin
tions
:

1
-
Lorsq
u'un
objet
**User
Form**
est
moda
l
(**Show
Modal=Tr
ue**),
l'utilis
ateur
doit
fourni
r des
infor
matio
ns

ou
ferme
r
l'obje
t
**User
Form**
pour
pouv

oir
utilise
r
toute
autre
partie
de
l'appli
catio
n.

2

-
Lorsq
u'un
objet
**User
Form**
n'est
pas
moda
l
(Sho
wMo
dal=F
alse),
l'utilis
ateur
peut
affich
er
d'autr
es
feuill
es

ou
fenêtr
es
sans
ferme
r cet
objet.

Le
code
qui
suit
devra
être
dans
le
modu
le de
code
du
User
Form

.

```
Private  
Declare  
Function  
Find  
WindowA  
Lib  
"User  
32" _  
(  
ByVal  
lpCla  
ssName  
As  
String,  
ByVal  
lpWin  
downName
```

As
String) As
Long

Private
Declare
Function
GetWidthLong
As
Lib
"User32" _
(
ByVal
hwnd As
Long,
ByVal
nIndex As
Long)
As
Long

Private
Declare
Function
SetWidthLong
As
Lib
"User32" _
(
ByVal
hwnd As
Long,
ByVal
nIndex
As
Long,

ByVal
dwNewLong
As
Long)
As
Long

Private
Sub
UserForm
_Activate()
Dim
hwnd As
Long,
exLong
As
Long
hwnd =
FindWindowA(

```

vbNul
lStrin
g,
Me.C
aptio
n)
    e
xLon
g =
GetW
indo
wLon
gA(h
Wnd,
-16)
    If
(exLo
ng
And
&H20
000)
= 0
Then

    Set
Wind
owLo
ngA
hWn
d,
-16,
exLo
ng Or
&H20
000

    Me.
Hide:
Me.S
how
    E
nd If
End
Sub

```

Ajouter une image à un bouton de formulaire.

Pour ajouter une image à un bouton de formulaire, il faut posséder une image de type Gif par exemple avec une taille d'environ 26x20 comme celles ci-dessous.

Vous créez votre bouton normalement et ensuite vous allez dans les propriétés de ce bouton.

- 1 - Sélectionner la propriété **Picture** et cliquez sur le bouton avec 3 points. Vous sélectionnez votre image.
- 2 - Sélectionner la propriété **PicturePosition** et choisissez l'option **2 - fmPicturepositionLeftBottom**. (C'est celle qui me paraît être la mieux adaptée).

[Remplir une combobox ou listbox](#)

Depuis une sheet

```

' *****
' Remplir une ComboBox ou Listbox depuis
' une feuille de calcul
' *****
Private Sub UserForm_Initialize()
    With Sheets( "Feuil1" ).Range( "A1" )
        Me.ComboBox1.RowSource = "Feuil1!A1:A" & Sheets( "Feuil1" ).Cells( 1 , 1 ).End(xlDown).Row
    End With
End Sub

```

Depuis une sheet (2 colonnes)

```

Private Sub UserForm_Initialize()
    With Sheets( "Feuil1" ).Range( "A1" )

```

```

        Me.ComboBox1.RowSource = "Feuil1!A1:B" & Sheets( "Feuil1" ).Cells( 1 , 1
    ).End(xlDown).Row
    End With
End Sub
' *****
' Avec une autre syntaxe
' *****
Private Sub UserForm1_Initialize()
    With Sheets( "Feuil1" ).Range( "A1" )
        Me.ComboBox1.RowSource = Range(.Cells, .End(xlDown)( 1 , 2 )).Address(External:= True )
    End With
End Sub

```

Avec la méthode Additem

```

' *****
' Ajout d'item avec la méthode AddItem pour
' remplir une ComboBox ou ListBox.
' *****
Private Sub UserForm_Initialize()
    With Me.ComboBox1
        ComboBox1.AddItem "Domaine Alexis Rouge"
        ComboBox1.AddItem "Domaine du Grand Crès Blanc"
        ComboBox1.AddItem "Domaine du Grand Crès Muscat"
        ComboBox1.AddItem "Domaine du Grand Crès Rosé"
        ComboBox1.AddItem "Domaine du Grand Crès Rouge"
    End With
End Sub

```

Avec la liste des onglets

```

' *****
' Ajout de la liste des onglets du classeur pour
' remplir une ComboBox ou ListBox.
' *****
Public Sub UserForm_Initialize()
    For Each c In ActiveWorkbook.Sheets
        ComboBox1.AddItem c.Name
    Next
End Sub

```

Avec les noms des mois

```

' *****
' Ajout d'items correspondant à des mois pour
' remplir une ComboBox ou ListBox.
' *****
Private Sub UserForm_Initialize()
    Dim Mois( 1 To 12 ) As String
    Dim i As Integer
    ' Creation d'un tableau des noms de mois
    For i = 1 To 12
        Mois(i) = Format ( DateSerial ( 1 , i , 1 ) , "mmm" )
        Me.ComboBox1.AddItem Mois(i)
    Next i
End Sub

```

Avec une plage nommée

```

' *****
' Utiliser une plage nommée pour
' remplir une ComboBox ou ListBox.
' *****
' La liste est définie dans le Userform par son nom entré à l'aide
' de la commande Insertion/Nom/Définir avec la formule suivante :
' NomClient=DECALER(Codes!$A$1;;;NBVAL(Codes!$A:$A));
' ou Codes correspond au nom de l'onglet du classeur et
' Nomclient étant le nom de la plage de données.
Private Sub UserForm_Initialize()
    ' Activation du classeur Workbook.xls
    Workbooks( "WorkBook.xls" ).Activate
    ComboBox1.RowSource = "Codes!NomClient"

```

End Sub

Avec une boucle et additem

```
' *****  
' Ajout d'items correspondant à des années pour  
' remplir une ComboBox ou Listbox.  
' *****
```

```
Private Sub UserForm_Initialize()  
    ReDim Annee( 1 To 12 )  
    For i = 1 To 12  
        Annee(i) = 1998 + (i)  
        Me.ComboBox1.AddItem Annee(i)  
    Next  
End Sub
```

Avec un fichier texte

```
' *****  
' Ajout d'item depuis un fichier texte pour  
' remplir une ComboBox ou Listbox.  
' *****  
Private Sub UserForm_Initialize()  
    ' Ouvre le fichier en lecture.  
    Open "C:\Excel\Donnees.txt" For Input As # 1  
    Do While Not EOF ( 1 )  
        ' Lit les données  
        Input # 1 , NomClient  
        ' Ajoute les données dans la ComboBox ou Listbox  
        Me.ComboBox1.AddItem NomClient  
    Loop  
    Close # 1  
End Sub
```

Avec la fonction array (tableau)

```
Private Sub UserForm_Initialize()  
    Dim vaDepartment As Variant  
    Dim vaDeptCode As Variant  
    Dim stDeptList() As String  
    Dim i As Integer  
    ' Liste des départements  
    vaDepartment = VBA.Array( "Ain" , "Aisne" , "Allier" , "Alpes (Hte Prov.)" , "Alpes (Hautes)" ,  
"Alpes (Maritimes)" , _  
    "Ardèche" , "Ardennes" )  
    ' Code des départements  
    vaDeptCode = VBA.Array( "01" , "02" , "03" , "04" , "05" , "06" , "07" , "08" )  
    ReDim stDeptList( 0 To UBound( vaDepartment) , 0 To 1 )  
    ' Boucle pour mise en place des valeurs  
    For i = 0 To UBound( vaDepartment)  
        stDeptList(i, 0) = vaDeptCode(i)  
        stDeptList(i, 1) = vaDepartment(i)  
    Next i  
    ComboBox1.List = stDeptList  
End Sub
```

```
' *****  
' Autre variante possible  
' *****
```

```
Private Sub UserForm1_Initialize()  
    Dim TabMonnaies( 4 , 2 )  
    ComboBox1.ColumnCount = 2  
    'Nom des monnaies proposées  
    TabMonnaies( 0 , 0 ) = "Franc"  
    TabMonnaies( 1 , 0 ) = "Euro"  
    TabMonnaies( 2 , 0 ) = "Dollar"  
    TabMonnaies( 3 , 0 ) = "Livre"  
    'Symboles des monnaies  
    TabMonnaies( 0 , 1 ) = " F"  
    TabMonnaies( 1 , 1 ) = " €"  
    TabMonnaies( 2 , 1 ) = " $"  
    TabMonnaies( 3 , 1 ) = " £"  
    ComboBox1.List() = TabMonnaies  
End Sub
```

Avec les fichiers XLS d'un répertoire

```
' *****  
' Lister les fichiers d'un répertoire donné  
' et les afficher dans une ComboBox ou Listbox.  
' Seul le nom du fichier sera affiché (Function ShortFilename)  
' *****  
Private Sub UserForm_Initialize()  
    Dim F  
    Dim LongFilename As String  
    With Application.FileSearch  
        .NewSearch  
        .LookIn = "C:\Excel\Traitement des déchets"  
        .Execute  
        On Error Resume Next  
        For Each F In .FoundFiles  
            LongFilename = F  
            Namefile = ShortFilename(LongFilename)  
            Me.ComboBox1.AddItem Namefile  
        Next F  
    End With  
End Sub
```

Boucle sur des contrôles

Efface le contenu des textbox

```
' *****  
' Met à blanc les différentes TextBox  
' contenues dans le formulaire  
' *****  
Private Sub cmdOK_Click()  
    Dim i As Integer  
    For Each Ctrl In Me.Controls  
        If TypeOf Ctrl Is MSForms.TextBox Then  
            i = i + 1  
            Ctrl.Value = ""  
        End If  
    Next Ctrl  
End Sub
```

Exporter les données des textbox

```
' *****  
' Exporte les valeurs des différentes TextBox  
' dans les cellules A1, A2, A3, etc..., lors  
' de la validation par le bouton OK.  
' *****  
Private Sub cmdOK_Click()  
    Dim i As Integer  
    For Each Ctrl In Me.Controls  
        If TypeOf Ctrl Is MSForms.TextBox Then  
            i = i + 1  
            Cells(i, 1) = Ctrl  
        End If  
    Next Ctrl  
End Sub
```

Remplir les textbox à l'ouverture

```
' *****  
' Remplit les contrôles TextBox du formulaire  
' avec les valeurs contenues dans les cellules A1, A2, etc...,  
' lors de l'ouverture du formulaire.  
' *****
```

```

Private Sub UserForm_Initialize()
    Dim i As Integer
    For Each Ctrl In Me.Controls
        If TypeOf Ctrl Is MSForms.TextBox Then
            i = i + 1
            Controls( "TextBox" & i).Value = Cells(i, 1 ).Value
        End If
    Next Ctrl
End Sub

```

Initialisation de checkbox

```

' *****
' Permet via un bouton placé sur le formulaire de
' réinitialiser les contrôles Checkbox (Mise à False)
' *****
Private Sub CmdInitCheckBox_Click()
    Dim Ctrl As MSForms.Control
    ' Remise à False des cases à cocher
    For Each Ctrl In UserForm1.Controls
        If TypeOf Ctrl Is MSForms.CheckBox Then Ctrl.Value = False
    Next Ctrl
End Sub

```

Les feuilles Excel (Worksheets)

Page mise à jour le : 28/03/2001

Vous trouverez dans cette rubrique des exemples concernant des programmes s'appliquant aux feuilles de calcul.

Comparer deux colonnes.

Effectuer un calcul sur une plage (Ex. Multiplier par 2 les cellules de la plage).

Effectuer une recopie incrémentée.

Insérer une ligne vide toutes les 2 lignes (ou plus).

Introduire une variable dans une formule SOMME (Ex. Un N° de ligne).

Introduire une variable dans une formule SOMME (Ex. l'adresse d'une cellule).

Mettre en évidence les cellules répondant à un critère.

Recherche du nombre d'occurrence.

Rechercher la dernière cellule d'une plage.

Recopier une plage variable vers une autre feuille du classeur.

Récupérer la colonne active sous forme littérale (A, AB, AZ, etc...).

Récupérer la somme d'une colonne dans une variable.

Récupérer le nombre de colonnes.

Récupérer le nombre de lignes.

Récupérer la somme du produit de 2 colonnes (Somme des N (A*B)).

Répertoire et nom de fichier dans pied de page.
Supprimer les doublons dans une colonne.
Supprimer les lignes vides d'une plage.
Supprimer un nom dans une feuille de calcul.
Supprimer tous les noms dans une feuille de calcul.
Supprimer les noms avec un joker dans une feuille de calcul.
Supprimer les lignes contenant du texte.
Utiliser des macro-commandes dans une feuille protégée.

Comparer deux colonnes.

```
Perm  
et  
d'ajo  
uter  
dans  
la  
colonne 1  
les  
valeu  
rs qui  
serai  
ent  
dans  
la  
colonne 2  
et  
pas  
dans  
la  
colonne 1.  
Sub  
Com  
pareT  
woCo  
lonne  
s()  
Dim  
Cell  
As  
Rang  
e,  
Plage  
As  
Rang  
e, l  
As  
Long  
Set  
Plage  
=  
Rang  
e("A1  
",  
[A1].  
End(  
xlDo  
wn))  
l =  
Plage  
.Cou  
nt  
Appli  
catio  
n.Scr  
eenU  
pdati
```

```

ng =
False
For
Each
Cell
In
Rang
e("B1
",
[B1].
End(
xlDo
wn))
If
Plage
.Find(
Cell,
Plage
(1).
xlVal
ues,
xlWh
ole)
Is
Nothi
ng
Then

I = I
+ 1

Cell
s(I, 1)
=
Cell
E
nd If
Next
Cell
End
Sub

```

Effectuer un calcul sur une plage.

```

Perm
et de
multi
plier
l'ens
embl
e
d'une
plage
par
une
valeu
r (2
dans
l'exe
mple)
Sub
Multi
plierP
arDe
ux()
'
Utilis
ation
d'un
table
au
inter
médi
aire
Dim
Tabl
Dim I
As
Integ
er, J
As
Integ
er
Tabl
=
Rang
e("A1

```

```

:C10
00").
Value
F
or I =
1 To
1000

For
J = 1
To 3

Tabl(I
, J) =
Tabl(I
, J) *
2

Nex
t J
N
ext I
Appli
catio
n.Scr
eenU
pdati
ng =
False
Rang
e("A1
:C10
00").
Value
=
Tabl
End
Sub

```

Effectuer une recopie incrémentée.

```

Perm
et
d'effe
ctuer
une
recop
ie
incré
ment
ée à
partir
de A1
et A2
Sub
Reco
pieIn
crem
entee
()
Set
Plage
Sour
ce =
Work
sheet
s("Fe
uil1").
Rang
e("A1
:A2")
.
Effect
ue
une
recop
ie
incré
ment
ée de

```

```
A1 à  
A20  
Set  
Plage  
ARe  
mplir  
=  
Work  
sheet  
s("Fe  
uil1").  
Rang  
e("A1  
:A20"  
)  
Plage  
Sour  
ce.Au  
toFill  
Desti  
natio  
n:=Pl  
ageA  
Rem  
plir  
End  
Sub
```

Insérer une ligne vide toutes les 2 lignes (ou plus...).

```
Perm  
et  
d'ins  
érer  
dans  
un  
table  
au  
une  
ligne  
vide  
toute  
s les  
deux  
ligne  
s (ou  
plus)  
Sub  
Macr  
olnse  
rtUne  
Ligne  
SurD  
eux()  
Dim  
Line  
As  
Integ  
er  
Rang  
e("A2  
").Sel  
ect  
Line  
= 1  
Reco  
mme  
nce:  
Line  
=  
Line  
+ 2  
Rows  
(Line)  
.Sele  
ct  
Selec  
tion.I  
nsert  
Shift:  
=xID  
own  
If  
Line  
<
```

Activ
eShe
et.Us
edRa
nge.
Rows
.Cou
nt
Then
G
oTo
Reco
mme
nce
End
If
End
Sub

**Introduire une variable dans une formule SOMME.
(Ex. Un N° de ligne)**

Sub
Varia
bleSo
mme(
)
' La
colon
ne A
est la
colon
ne où
l'on
doit
effect
uer
une
som
me
' Se
place
r sur
la
premi
ère
ligne
vide
de la
colon
ne
pour
poser
la
formu
le
Som
me
Rang
e("A1
").En
d(xID
own).
Offse
t(1,
0).Se
lect
'
Récu
pérati
on du
nomb
re de
ligne
s à
comp
tabilis
er
dans
la
formu
le
Mon
NoDe
Ligne

```

= -
(Activ
eCell.
Row)
+ 1
Activ
eCell.
Form
ulaR
1C1
=
"=SU
M(R[
&
Mon
NoDe
Ligne
&
]C:R
[
1]C)"
End
Sub

```

Introduire une variable dans une formule SOMME. (Ex. L'adresse d'une cellule)

```

Sub
Varia
bleAd
ress
Dans
Som
me()
Rang
e("A1
").En
d(xID
own).
Offse
t(1,
0).Val
ue =
"=SU
M(A1
:" &
Rang
e("A1
").En
d_
(xIDo
wn).A
ddres
s &
)"
End
Sub

```

Mettre en évidence les cellules répondant à un critère.

Ce programme va sélectionner toutes les cellules correspondant à la valeur contenue dans D2.

```

Sub SelectCellulesValeurDeterminee()
LaValeur = Range("D2").Value
Range("A1").Select
For Each cll In ActiveCell.CurrentRegion
If cll.Value = LaValeur Then Plg = Plg & cll.Address() & ","
Next cll
If Len(Plg) > 0 Then Range(Left(Plg, Len(Plg) - 1)).Select
End Sub

```

Pour sélectionner les lignes à la place des cellules :

```

Remplacer la ligne de code
If cll.Value = LaValeur Then Plg = Plg & cll.Address() & ","
par
If cll.Value = LaValeur Then Plg = Plg & cll.Row() & ":" & cll.Row() & ","

```

Recherche du nombre d'occurrence.

Ce
progr

```

amm
e va
comp
ter le
nomb
re de
fois
qu'il
renco
ntre
la
valeu
r
Valeu
rACh
erche
r
Sub
Rech
erche
NbOc
curen
ce()
Résul
tat =
Appli
catio
n.Co
untif(
Rang
e("A:
A"),
"Vale
urAC
herch
er")
MsgB
ox
("Le
texte
Valeu
rACh
erche
r est
prése
nt : "
&
Résul
tat &
"
fois.")
End
Sub

```

Rechercher la dernière cellule d'une plage.

```

Ce
progr
amm
e
perm
et de
positi
onner
le
curse
ur sur
la
demi
ère
cellul
e
d'une
plage
corre
spon
dant
à la
cellul
e en
cours
.
Sub
Aller
ADer

```

```

niere
cellul
e()
'Séle
ction
ne la
plage
en
cours
Activ
eShe
et.Us
edRa
nge.
Selec
t
'Séle
ction
ne la
dermi
ère
cellul
e de
la
plage
en
cours
Activ
eShe
et.Us
edRa
nge.
Speci
alCell
s(xlC
ellTy
peLa
stCell
).Offs
et(0,
0).Se
lect
End
Sub

```

Recopier une plage variable vers une autre feuille du classeur.

Ce programme permet de recopier une plage variable A8 à Gx vers une autre feuille (Feuil2) en A1. correspondant à la cellule en cours.

```

Sub SelectRecopie()
' Redéfinir la plage en fonction du nombre de lignes
' Sélectionner cette plage
' Copier les valeurs contenues dans cette plage sur la feuille 2 à la cellule "A1"
With Worksheets(1)
.Range(. [A1], . [G1].End(xlDown)).Copy Worksheets(2).[A1]
End With
End Sub

```

Récupérer la colonne active sous forme littérale (A, AB, AZ, etc...).

Ce programme permet de récupérer la lettre de la colonne de la cellule active.

```

Sub LetCol()
Dim Let_Col As String
Let_Col = Left(Mid(ActiveCell.Address, 2), _
Len(Mid(ActiveCell.Address, 2)) - (Len(Mid(Mid(ActiveCell.Address, 2), _
Application.Search("$", Mid(ActiveCell.Address, 2), 1) + 1)) + 1))
MsgBox Let_Col
End Sub

```

Autre variante avec utilisation des valeurs VRAI (-1) et FAUX (0)

```

Sub LettreColonne()
' Utilisation des valeurs VRAI (-1) et FAUX (0)
Lettre_Col = Left(ActiveCell.Address(0, 0), (ActiveCell.Column < 27) + 2)
MsgBox Lettre_Col
End Sub

```

Récupérer la somme d'une colonne dans une variable.

Ce programme effectue la somme de la colonne A et la met dans une variable LaSomme.

```

Sub FaireSommeColonne()
LaSomme = Application.Sum(Range("A1").EntireColumn)
MsgBox LaSomme
End Sub

```

Récupérer le nombre de colonnes.

```

1 -
Nom
bre
de

```

colon
nes
comp
rises
entre
la
premi
ère
cellul
e et
la
demi
ère
cellul
e
de la
plage
sélec
tionn
ée (y
comp
ris
évent
uelle
ment
les
colon
nes
vides
)

Sub
Nom
breD
eCol
onne
s()
NbCo
lonne
s=Act
iveSh
eet.U
sedR
ange.
Colu
mns.
Coun
t
MsgB
ox
NbCo
lonne
s
End
Sub

2 -
Nom
bre
de
colon
nes
comp
rises
entre
la
colon
ne A
(vide
ou
pas)
et
la
demi
ère
cellul
e
utilisé
e
(cette
sélec
tion
peut
conte
nir
des
colon

```

nes
vides
).
Sub
Nom
breD
eCol
onne
s()
NbCo
lonne
s =
Cells.
Speci
alCell
s(xlC
ellTy
peLa
stCell
).Col
umn
MsgB
ox
NbCo
lonne
s
End
Sub

```

Récupérer le nombre de lignes.

1 - Nombre de lignes comprises entre la première cellule et la dernière cellule de la plage sélectionnée (y compris éventuellement les lignes vides)

```

Sub NombreDeColonnes()
NbLignes=ActiveSheet.UsedRange.Rows.Count
MsgBox NbLignes
End Sub

```

2 - Nombre de lignes comprises entre la colonne A (vide ou pas) et la dernière cellule utilisée (cette sélection peut contenir des lignes vides).

```

Sub NombreDeLignes()
NbLignes = Cells.SpecialCells(xlCellTypeLastCell).Column
MsgBox NbLignes
End Sub

```

Récupérer la somme du produit de 2 colonnes.

Somme des N valeurs (A1*B1 + A2*B2 ...)

Ce programme effectue le produit des valeurs de la colonne A par celles de la colonne B et additionne ces résultats.

```

Sub SommeProduit()
Cells(1, "A").Select
NbLignes = Cells(Range("A:A").Count, ActiveCell.Column).End(xlUp).Row
NbLignes = Cells(16, 1).End(xlUp).Row
TotalIntermediaire = 0
For Lgn = 2 To NbLignes 'Etiquettes en ligne 1
    If Not Rows(NbLignes).Hidden Then
        TotalIntermediaire = TotalIntermediaire + Cells(Lgn, "A") * Cells(Lgn, "B")
    End If
Next Lgn
MsgBox TotalIntermediaire
End Sub

```

Répertoire et nom de fichier dans pied de page.

Proc
édur
e à
mettr
e
dans
This
Work
Book
La
mise
à jour
ce
fait
dès
que
l'on
activ
e un
ongle
t.

```
Private
Sub Workbook_SheetActivate(ByVal Sh As Object)
    Dim ZtPath As String
    ZtPath = Application.ActiveWorkbook.Path & "\ " & "&F"
End Sub
```

Supprimer les doublons dans une colonne.

```
Sub EffacerDoublonsColonnes()
    Const CellDépart As String = "A1"
    Dim Fin As Rang, I As
```

```
Long,
J As
Long,
Col
As
Integ
er
Dim
Mode
Calcu
l As
Long
With
Appli
catio
n
    M
odeC
alcul
=
.Calc
ulation
.
Calcu
lation
=
xlCal
culati
onMa
nual
.
Scre
enUp
datin
g =
False
End
With
Col =
Rang
e(Cel
l_Dé
part).
Colu
mn
Set
Fin =
Rang
e(Cel
l_Dé
part).
End(
xlDo
wn)
(2)
On
Error
Resu
me
Next
Do
    I
= J +
1
    J
=
Rang
e(Cel
ls(I,
1),
Fin).
Colu
mnDi
fferen
ces(
Cells(
I, 1)
(0).R
ow
    If
J > I
Then
Rang
e(Cel
ls(I +
```

```
1, 1),  
Cells(  
J,  
1)).ClearContents  
Loop  
Until  
Err  
Application.Calculation =  
ModeCalculation  
End  
Sub
```

Supprimer les lignes vides d'une plage.

```
Sub  
programme  
supprime  
les  
lignes  
vides  
dans  
une  
plage  
.  
Sub  
DetruireLigne()  
DerniereLigne =  
ActiveSheet.Range.  
Rows.Count  
Application.ScreenUpdating =  
False  
For R =  
=  
DerniereLigne  
To 1  
Step  
-1  
If  
Application.CountA(Rows(R)) = 0  
Then  
Rows(R).Delete  
Next  
R  
End  
Sub
```

Supprimer un nom dans une feuille de calcul.

Ce
progr
amm
e
suppr
ime
le
nom
"Bas
e"
dans
toute
s
les
feuill
es du
class
eur.
Sub
rime
UnNo
m()
For
Each
Nm
In
Activ
eWor
kboo
k.Na
mes
If
Nm.N
ame
Like
"Bas
e"
Then
N
m.De
lete
End
If
Next
Nm
End
Sub

Supprimer tous les noms dans une feuille de calcul.

Ce
progr
amm
e
suppr
ime
tous
les
noms
dans
le
class
eur.
Sub
Supp
rimeT
ousle
sNo
ms()
For
Each
Nm
In
Activ
eWor
kboo
k.Na
mes
N
m.De
lete
Next
Nm
End
Sub

Supprimer tous les noms (avec un joker) dans une feuille de calcul.

```
Ce
progr
amm
e
suppr
ime
tous
les
noms
dans
le
class
eur.
Sub
Supp
rime
Nom
sAve
cJok
er()
For
Each
Nm
In
Activ
eWor
kboo
k.Na
mes
If
Nm.N
ame
Like
"*Bas
e"
Then
N
m.De
lete
End
If
End
Sub
```

Supprimer les lignes contenant du texte

Ce programme recherche dans la colonne A les cellules contenant du texte et supprime les lignes associées à ces cellules.

```
Sub SupprimeLignesAvecTexte()
NbRw = Application.CountA(Columns("A:A"))
For Rw = NbRw To 1 Step -1
    If Application.IsText(Cells(Rw, 1)) Then Rows(Rw).Delete
Next Rw
End Sub
```

Utiliser des macro-commandes dans une feuille protégée.

Ce programme permet d'utiliser des macro-commandes dans une feuille protégée par Outils, Protection.

[Worksheets\("Feuil1"\).Protect UserInterfaceOnly:=True](#) permet de manipuler par macro une feuille protégée, tout en laissant la protection active par rapport aux manipulations de l'utilisateur.

```
Private Sub Workbook_Open()
    Worksheets("Feuil1").Protect UserInterfaceOnly:=True
End Sub
```

Instruction à mettre dans [Workbook_Open](#), cette propriété n'étant pas enregistrée avec le classeur.

Les fichiers (Excel ou autres)

Page mise à jour le : 10/02/2002

Vous trouverez dans cette rubrique des exemples de programmes se rapportant au traitement sur des fichiers Excel ou autres (txt, ini, log).

Ecrire dans un fichier de type texte (Date, heure) à l'ouverture ou la fermeture d'Excel.

Lire et incrémenter un fichier INI.

Lire les enregistrements dans un fichier texte.

Lister des fichiers texte et les ouvrir avec GetOpenFileName.

Lister tous les fichiers XLS dans une feuille de calcul.

Réaliser une copie du classeur actif sous un autre nom.

Récupérer le nom d'un fichier sélectionné par GetOpenFileName.

Ecrire dans un fichier de type texte (date et heure).

Ce
programm
e
utilise
les
évén
ements
Workbook
Open et
Workbook
Close
pour
inscri
re
des
infor
matio
ns
dans
un
fichier
de
type
texte
lors
de
l'ouv
erture
et
de la
ferme
ture
d'Exc
el.
Dans
l'ex
emple,
le
fichier
texte
se
nom
me
activité.log

Private
Sub
Workbook
Open()
End Sub

```
D
im
LogFile As String
L
ogFile =
"C:\Excel\
activite.log"
C
hDir
"C:\Excel"
D
onnees =
Now(
)
O
pen
LogFile For
Append
Shared As
#1
P
rint
#1,
"Ouv
erture
d'Excel a "
&
Donnees
C
lose
#1
End
Sub
```

```
Private
Sub
Workbook
_BeforeClose(
Cancel As Boolean)
D
im
LogFile As String
L
ogFile =
"C:\Excel\
activite.log"
C
hDir
"C:\Excel"
D
onnees =
Now(
)
O
pen
```

```

LogFi
le For
Appe
nd
Shar
ed As
#1
P
rint
#1,
"Fer
metur
e
d'Exc
el a "
&
Donn
ees
P
rint
#1,
"-----
-----
-----
-----
"
C
lose
#1
End
Sub

```

E
xem
ple
de
fichi
er
géné
ré

Lire et incrémenter un fichier INI

Ce programme permet d'incrémenter un fichier `incred.ini` et de récupérer la valeur contenue dans ce fichier. Arrivé à 1000, le compteur est réinitialisé à 1.

Structure du fichier `incred.ini`

```

[Numero]
NUMERO=4

```

Ne pas oublier de copier les deux lignes qui suivent en tête de votre module.

```

Declare Function GetPrivateProfileStringA Lib "Kernel32" (ByVal lpAppName As _
String, ByVal lpKeyName As String, ByVal lpDefault As String, ByVal lpReturnedString _
As String, ByVal nSize As Long, ByVal lpFileName As String) As Long

```

```

Declare Function WritePrivateProfileStringA Lib "Kernel32" (ByVal lpAppName _
As String, ByVal lpKeyName As String, ByVal lpString As String, _
ByVal lpFileName As String) As Long

```

```

Sub IncrémentIni()
Dim Compteur As String * 10
GetPrivateProfileStringA "Numero", "NUMERO", 1, Compteur, 10, "C:\Windows\Incred.ini"
WritePrivateProfileStringA "Numero", "NUMERO", CStr(CLng(Compteur) + 1), _
"C:\Windows\Incred.ini"
MsgBox "Le compteur est incrémenté à : " & Compteur & "."
If Compteur = 1000 Then
MsgBox (" La valeur de 1000 est atteinte. Remise à 1 du compteur.")
Compteur = 1
WritePrivateProfileStringA "Numero", "NUMERO", CStr(CLng(Compteur)), _
"C:\Windows\Incred.ini"
End If
End Sub

```

Lire les enregistrements dans un fichier texte

Ce programme lit les différents enregistrements dans un fichier texte et les inscrit dans une feuille de calcul.

Structure du fichier Listing.txt

Jordan,Durand,15
Eric,Bataille,52
Marcel,Dupond,35

```
Sub LireFichierTexte()  
Dim Prenom, Nom, Age  
' Ouvre le fichier en lecture  
Open "C:\Excel\Listing.txt" For Input As #1  
' Effectue la boucle jusqu'à la fin du fichier  
Do While Not EOF(1)  
' Lit les données dans trois variables  
Input #1, Prenom, Nom, Age  
' Ecrit les données dans la feuille de calcul à partir de la ligne 2  
Range("A65536").End(xlUp)(2).Value = Prenom  
Range("B65536").End(xlUp)(2).Value = Nom  
Range("C65536").End(xlUp)(2).Value = Age  
Loop  
' Ferme le fichier  
Close #1  
End Sub
```

Le résultat

Lister des fichiers texte et les ouvrir avec GetOpenFileName

[GetOpenFileName](#)

Affiche la boîte de dialogue standard Ouvrir et lit le nom de fichier tapé ou sélectionné par l'utilisateur

sans réellement ouvrir les fichiers.
Sub ChoixFichierTexteAOuvrir()
C
hDir "C:\"
C
hDir "c:\E

```
xcel"
    C
eFich
ier
=App
licatio
n.Get
Open
Filen
ame(
"Text
Files
(*.txt)
,
*.txt")
If
VarTy
pe(C
eFich
ier) =
vbBo
olean
Then

    Exit
Sub
Else

    Wor
kboo
ks.O
penT
ext
Filen
ame:
=CeF
ichier
,
Origi
n:=xl
Wind
ows,
-

Start
Row:
=1,
Data
Type:
=xlD
elimit
ed,
Text
Quali
fier:=
xlDou
bleQ
uote,
-

Cons
ecuti
veDel
imiter
:=Fal
se,
Tab:=
True,
Semi
colon
:=Fal
se,
Com
ma:=
False
, -

Spac
e:=F
alse,
Other
:=Fal
```

```

se,
FieldI
nfo:=
Array
(1, 1)
E
nd If
End
Sub

```

Lister tous les fichiers XLS dans une feuille de calcul

Ce programme permet de rechercher tous les fichiers Excel du répertoire Excel, de les placer dans un tableau et de copier ce tableau dans une feuille de calcul.

```

Sub RechercheClasseursSurDisque()
Dim Classeurs() As String, I As Long
With Application.FileSearch
.NewSearch
.FileType = msoFileTypeExcelWorkbooks
.LookIn = "C:\Excel\"
.SearchSubFolders = True
.Execute
With .FoundFiles
ReDim Classeurs(1 To .Count, 1 To 1)
For I = 1 To .Count
Classeurs(I, 1) = .Item(I)
Next I
Application.ScreenUpdating = False
With Range("A1").Resize(.Count)
.Value = Classeurs
.Sort [A1]
End With
End With
End With
End Sub

```

Réaliser une copie du classeur actif sous un autre nom

```

Ce
progr
amm
e
enreg
istre
le
class
eur
actif
sous
un
autre
nom
(une
copie
)
sans
pour
autan
t
modif
ier le
nom
du
class
eur
actif.
Sub
Save
Copy
As()
A
ctive
Work
book.
Save
Copy
As
"C:\e
xcel\
Doub
le.xls
"
End
Sub

```

Récupérer le nom d'un fichier sélectionné par GetOpenFileName

[Cas d'utilisation on](#)
Par exemple si vous faites une lecture/écriture (via OpenFileDialog #1) d'un fichier sélectionné par GetOpenFileName pour inscrire les enregistrements dans un nouveau classeur (via Workbook.Worksheet), vous n'avez à aucun moment ouvert ce fichier mais vous voulez en connaître le nom pour le donner à votre classeur actif.

```
Public
NameSan
sExtension
As
String
Sub
SelectionFichier(
)
Dim
Long
Filename
As
String
L
ongFilename =
Application.Get
Open
Filename(
"Text
Files
(*.txt)
,
*.txt")
ShortFilename
(Long
Filename)
MsgBox "Le
nom
sans
extension
du
fichier est :
" &
NameSan
sExtension
End
Sub
```

```
Function
Short
Filename(
Long
Filename
As
String) As
String
```

```
For i
=
Len(Long
Filename)
To 1
Step
```

-1

```
If  
Mid(L  
ongFi  
lena  
me, i,  
1) =  
"1"  
Then  
Exit  
For  
N  
ext  
S  
hortFi  
lena  
me =  
Mid(L  
ongFi  
lena  
me, i  
+ 1,  
Len(L  
ongFi  
lena  
me))  
N  
ame  
Sans  
Exten  
sion  
=  
Mid(  
Short  
Filen  
ame,  
1,  
Len(  
Short  
Filen  
ame)  
- 4)  
End  
Funct  
ion
```

Les fonctions personnalisées

Page mise à jour le : 28/03/2001

Insérer le chemin du classeur dans une feuille de calcul.
Somme des cellules avec un motif rouge par exemple.

Insérer le chemin d'un classeur dans une feuille de calcul.

Cette fonction insère le chemin du classeur dans la feuille de calcul.

Le code

[Code de la fonction InsereCheminClasseur](#)

```
Function  
InsereCheminClasseur()  
As String  
  
Application.Volatile  
|  
InsereCheminClasseur =  
Application.Caller.Parent.Parent.FullName  
End Function
```

Somme des cellules avec un motif rouge.

Cette fonction calcule la somme d'une plage en fonction de la couleur du motif des cellules.

Le code

[Code de la fonction SumByColor](#)

```
Function SumByColor(PlageEntree As Range, CouleurPlage As Range) As Double  
Dim Cell As Range  
TempSum As Double  
ColorIndex As Integer  
ColorIn
```

```

dex =
Coul
eurPl
age.
Cells(
1,
1).Int
erior.
Color
Index
T
emp
Sum
= 0
O
n
Error
Resu
me
Next
F
or
Each
Cell
In
Plage
Entre
e.Cell
s

If
Cell.
Form
ula <
> ""
Then

I
f
Cell.I
nterio
r.Col
orInd
ex =
Color
Index
Then
Temp
Sum
=
Temp
Sum
+ -

Cell.
Value

End
If
N
ext
Cell
O
n
Error
GoTo
0
S
et
Cell
=
Nothi
ng
S
umBy
Color
=
Temp
Sum
End
Funct
ion

```

Les paramètres de la fonction

PlagePermet de sélectionner la plage à calculer.

CouleurEntree : Sélectionne la cellule servant de référence pour la couleur du motif.

Les formules Excel

Page mise à jour le : 10/02/2002

Vous trouverez dans cette rubrique un recueil de formules pouvant vous rendre de précieux services.

Calcul sur 12 mois glissants.

Calcul sur les heures.

Calcul de l'ancienneté en Mois et Années.

Calculer une somme entre deux bornes.

Exemples de formats personnalisés.

Moyenne ne tenant pas compte des valeurs zéro.

Réaliser un filtre élaboré (Champ calculé).

Réaliser un filtre élaboré (Champ avec condition ET).

Récupérer la valeur d'une cellule résultant d'une intersection Ligne/Colonne.

Somme des cellules (A1 par ex.) de plusieurs feuilles.

Somme d'une colonne avec un nombre de lignes variable.

Trouver le quantième du jour.

Trouver les dates mobiles (jours fériés).

Calcul sur 12 mois glissants

Calcul sur les heures

Calcul de l'ancienneté en Mois et Années

Calculer une somme entre deux bornes

Exemples de formats personnalisés

Moyenne ne tenant pas compte des valeurs zéro

Réaliser un filtre élaboré (Champ calculé)

Le champ calculé permet d'extraire les noms dont le salaire actuel est \geq au salaire de début x 2 .

Réaliser un filtre élaboré (Champ calculé avec condition ET)

Extraire les noms dont la date de début est \geq au 31/12/1997 **ET** la date de fin $<$ 01/01/2001.

Récupérer la valeur d'une cellule résultant d'une intersection Ligne/Colonne

Somme des cellules (A1 par ex.) de plusieurs feuilles

Somme d'une colonne avec un nombre de lignes variable

Cas 1 : Les valeurs commencent à la ligne 1

=SOMME(DECALER(\$A\$1;NB(A:A);;-NB(A:A)))

Cas 2 : Les valeurs commencent à la ligne 2

=SOMME(DECALER(\$A\$1;NB(A:A);;-NB(A:A)-1))

Trouver le quantième du jour

Trouver les dates mobiles (jours fériés)

Les jours fériés sont de deux sortes :

Jours à date fixe (comme le 14 juillet, le 11 novembre et le 25 décembre)

Jours mobiles, dont la date varie d'une année sur l'autre. Ces jours fériés mobiles sont calculés d'après la date du dimanche de Pâques.

Dès lors que l'on connaît la date du dimanche de Pâques, on peut en dériver les dates de tous les jours fériés mobiles par simple addition.

Le dimanche de Pâques est le premier dimanche qui suit la première pleine lune après l'équinoxe de printemps (21 mars).

La formule suivante, signée Norbert Hetterich, renvoie la date du dimanche de Pâques pour l'année A

(entre 1900 et 2078) :

=FRANC((JOUR(MINUTE(A/38)/2+55)&"/4/"&A)/7;)*7-6

Les jours fériés mobiles en France et en Belgique sont le Lundi de Pâques (Dimanche de Pâques ("DP") + 1),

le Jeudi de l'Ascension (DP + 39) et le Lundi de Pentecôte (DP + 50).

En partant de la formule de N. Hetterich, on peut donc obtenir facilement les dates de ces trois jours

fériés pour une année donnée (entre 1900 et 2078) :

Les graphiques Excel

Page mise à jour le : 28/03/2001

Vous trouverez dans cette rubrique un recueil de graphiques pouvant vous rendre de précieux services.

Réaliser un graphique dynamique (Données en colonnes).

Réaliser un graphique dynamique sur les 12 dernières valeurs (Données en colonnes).

Réaliser un graphique dynamique (Données en lignes).
Réaliser un graphique dynamique sur les 12 dernières valeurs (Données en lignes).

Réaliser un graphique dynamique (données en colonnes)
L'onglet s'intitule Données et le fichier GraphiqueDynamique.
Les données sont en colonnes.

Réaliser un graphique dynamique sur les 12 dernières valeurs (en colonnes)
L'onglet s'intitule Etat Commandes et le fichier GraphiqueDynamique.
Les données sont en colonnes.

Réaliser un graphique dynamique sur données en lignes
L'onglet s'intitule Personnel et le fichier GraphiqueDynamique.
Les données sont en lignes.

Réaliser un graphique dynamique sur les 12 dernières valeurs en lignes
L'onglet s'intitule Effectif et le fichier GraphiqueDynamique.
Les données sont en lignes.

Les menus et barres d'outils

Vous trouverez dans cette rubrique des exemples pour créer, modifier ou personnaliser les menus d'Excel.

Ajout d'un item au menu Outils.

1 - Ajouter un item intitulé Exemple en dernière position du menu Outils.

```
Le code pour installer l'item dans le menu Outils
Sub AddItemInMenuOutils()
    With Application.CommandBars("Tools").Controls.Add(msoControlButton)
        .Caption = "Exemple"
        .OnAction = "Affiche"
        .FaceId = 343
    End With
End Sub
Le code pour effacer l'item du menu Outils
Sub EffaceItem()
    CommandBars("Tools").Controls("Exemple").Delete
End Sub
Le code de l'action associée à l'item Exemple
Sub Affiche()
    Dim Msg, Style, Title
    Msg = "Vous avez ajouté un item intitulé {Exemple} dans le menu Outils"
    Style = vbYes
    Title = "Ajout d'items dans le menu Outils - © Eric RENAUD - 11/1999"
    Response = MsgBox(Msg, Style, Title)
End Sub
```

2 - Ajouter un item intitulé Exemple après l'item Protection du menu Outils.

```
Le
code
pour
installer
l'item
dans
le
menu
Outil
```

```
s
Sub
AddIt
emIn
Menu
Outils
()
  Wit
  h
  Appli
  catio
  n.Co
  mma
  ndBa
  rs("T
  ools")
  .Cont
  rols.A
  dd(m
  soCo
  ntrol
  Butto
  n,
  befor
  e:=7)
  .
  Capti
  on =
  "Exe
  mple"
  .
  OnAc
  tion =
  "Affic
  he"
  .
  FaceI
  d =
  343
  En
  d
  With
  End
  Sub
  Le
  code
  pour
  effac
  er
  l'item
  du
  men
  u
  Outil
  s
  Sub
  Effac
  eltem
  ()
  Co
  mma
  ndBa
  rs("T
  ools")
  .Cont
  rols("
  Exem
  ple").
  Delet
  e
  End
  Sub
  Le
  code
  de
  l'acti
  on
  asso
  ciée
  à
  l'item
  Exe
  mple
  Sub
  Affich
  e()
```

```

Dim
Msg,
Style,
Title
Msg
=
"Vous
s
avez
ajout
é un
item
intitul
é
{Exe
mple}
dans
le
menu
Outils
"
Style
=
vbYe
s
Title
=
"Ajou
t
d'ite
ms
dans
le
menu
Outils
- ©
Eric
REN
AUD
-
11/19
99"
Resp
onse
=
MsgB
ox(M
sg,
Style,
Title)
End
Sub

```

Ajout d'un item au menu Outils, Protection.

Ajouter un item intitulé **Exemple** dans le sous-menu **Protection** du menu **Outils**.

Le code pour installer l'item dans le menu Outils, Protection

```

Sub AddItemInMenuOutilsProtection()
    With Application.CommandBars("Protection").Controls.Add(msoControlButton)
        .Caption = "Exemple"
        .OnAction = "Affiche"
        .FaceId = 343
    End With
End Sub

```

Le code pour effacer l'item du menu Outils, Protection

```

Sub EffaceItem()
    CommandBars("Tools").Controls("Protection").Controls(4).Delete
End Sub

```

Le code de l'action associée à l'item Exemple

```

Sub Affiche()
    Dim Msg, Style, Title
    Msg = "Vous avez ajouté un item intitulé {Exemple} dans le menu Outils."
    Style = vbYes
    Title = "Ajout d'items dans le menu Outils, Protection - © Eric RENAUD - 11/1999"
    Response = MsgBox(Msg, Style, Title)
End Sub

```

Ajout d'un nouveau menu.

1 - Ajouter un menu **Exemple** dans la barre de menus après le menu **?**

Le code pour ajouter le menu
u
Exemple

```
.
Sub
Inser
enou
veau
Menu
()
With
Com
mand
Bars(
1).Co
ntrols
.Add(
mso
Contr
olPop
up)

.Capt
ion =
"Exe
mple"

With .
Contr
ols.A
dd(m
soCo
ntrol
Butto
n)

.Capt
ion =
"Tri
crois
sant"
.F
acelD
= 210

Begin
Grou
p =
True

.OnAc
tion =
"Affic
he"
En
d
With
Wit
h
.Cont
rols.A
dd(m
soCo
ntrol
Butto
n)

.Capt
ion =
"Tri
décro
issant"
.F
acelD
= 211

Begin
Grou
```

```
p =  
True  
. .  
OnAc  
tion =  
"Affic  
he"  
End  
d  
With  
End  
With  
End  
Sub  
Le  
code  
pour  
supp  
rimer  
le  
men  
u  
Exe  
mple  
. .  
Sub  
Supp  
rime  
Nouv  
eauM  
enu()  
Com  
mand  
Bars(  
1).Co  
ntrols  
("Exe  
mple"  
).Del  
ete  
End  
Sub  
Le  
code  
de  
l'acti  
on  
asso  
ciée  
aux  
item  
s  
Exe  
mple  
Sub  
Affich  
e()  
Dim  
Msg,  
Style,  
Title  
Msg  
=  
"Vou  
s  
avez  
ajout  
é un  
nouv  
eau  
menu  
{Exe  
mple}  
dans  
la  
barre  
de  
menu  
s"  
Style  
=  
vbYe  
s  
Title  
=  
"Ajou
```

t d'un
nouv
eau
menu
- ©
Eric
REN
AUD
-
11/19
99"
Resp
onse
=
MsgB
ox(M
sg,
Style,
Title)
End
Sub

2 - Ajouter un menu **Exemple** dans la barre de menus avant le menu **Format**.

Le
code
pour
ajout
er le
men
u
Exe
mple
avan
t le
men
u
Form
at.
Ce
code
est
identi
que à
l'exe
mple
précé
dent
sauf
qu'il
faut
ajout
er
before
=5,
5
étant
le le
numé
ro du
menu
avant
leque
l on
veut
insér
er le
menu
Exem
ple.
La
ligne
de
code
:
With
Com
mand
Bars(
1),Co
ntrols
.Add(
mso
Contr
olPop

```
up)
devie
nt :
  With
  Com
  mand
  Bars(
  1),Co
  ntrols
  .Add(
  mso
  Contr
  olPop
  up,be
  fore:
  =5)
```

Ajout d'items au menu contextuel (clic droit).

1 - Ajouter un item [Exemple](#) dans menu contextuel (clic droit souris).

```
Le
code
pour
ajouter
le
menu
contextuel.
Sub
Inser
eMen
uCon
textu
el()
With
Appli
catio
n.Co
mma
ndBa
rs("C
ell").
Contr
ols.A
dd(m
soCo
ntrol
Butto
n)
.
Capti
on =
"Exe
mple"
.
Begin
Grou
p =
True
.F
aceld
= 343
.
OnAc
tion =
"Affic
he"
End
With
End
Sub
Le
code
pour
supp
rimer
le
menu
cont
```

```
extuel.  
Sub  
Supprime  
Menu  
Cont  
extuel()  
Application.  
CommandBars("Cell").  
Reset  
End  
Sub  
Le  
code  
de  
l'action  
associée  
à  
l'item  
Exemple  
Sub  
Affiche()  
Dim  
Msg, Style,  
Title  
Msg  
= "Vous  
avez  
ajouté un  
item  
intitulé  
{Exemple}  
dans  
le  
menu  
Outils  
"  
Style  
= vbYes  
Title  
= "Ajout  
d'item  
dans  
menu  
contextuel  
- ©  
Eric  
REN  
AUD  
-  
11/19  
99"  
Response  
= MsgBox(Msg,  
Style,  
Title)  
End  
Sub
```

2 - Ajouter un menu popup depuis l'item **Exemple** dans menu contextuel (clic droit souris).

```
Le
code
pour
ajouter
le
menu
u
cont
extu
el.
Sub
Inser
eMen
uCon
textu
ePop
Up()
With
Appli
catio
n.Co
mma
ndBa
rs("C
ell")
    Wit
h
    .Cont
rols.A
dd(m
soCo
ntrol
Popu
p)
    .
    Capti
on =
    "Exe
mple"
    .
    Begin
    Grou
p =
    False
    .
    Sous
    -
    menu
    1
    (Exe
mple
1.1)
    .
    Contr
ols.A
dd
(mso
Contr
olButt
on)
    .
    Contr
ols(1)
    .Capt
ion =
    "Exe
mple
1.1"
    With .
    Contr
ols(1)
    .OnA
ction
    =
    "Affic
he"
    .Face
    Id =
```

351

E

nd

With

.

Sous

-

menu

2

(Exe

mple

1.2)

.

Contr

ols.A

dd

(mso

Contr

olButt

on)

.

Contr

ols(2)

.Capt

ion =

"Exe

mple

1.2"

.

With .

Contr

ols(2)

.

.OnA

ction

=

"Affic

he"

.

.Face

Id =

352

E

nd

With

En

d

With

End

With

End

Sub

Le

code

pour

supp

rimer

le

men

u

cont

extu

el.

Sub

Supp

rime

Menu

Cont

extue

l()

Appli

catio

n.Co

mma

ndBa

rs("C

ell").

Rese

t

End

Sub

Le

code

de

l'acti

on

```

asso
ciée
à
l'item
Exe
mple
Sub
Affich
e()
Dim
Msg,
Style,
Title
Msg
=
"Vous
avez
ajout
é un
item
intitul
é
{Exe
mple}
dans
le
menu
Outils
"
Style
=
vbYe
s
Title
=
"Ajou
t
d'ite
m
dans
menu
conten
tuel
- ©
Eric
REN
AUD
-
11/19
99"
Resp
onse
=
MsgB
ox(M
sg,
Style,
Title)
End
Sub

```

Masquer des items dans un menu (ou réafficher).

1 - Rendre inaccessible le menu [Outils](#).

```

Le
code
pour
rendr
e
inacc
essib
le
l'item
Outil
s.
Sub
Inhib
eMen
uOuti
ls()
Com

```

```
mand
Bars(
1).Co
ntrols
(6).E
nable
d =
False
End
Sub
Le
code
pour
rendr
e
acce
ssibl
e
l'item
Outil
s.
Sub
Affich
eMen
uOuti
ls()
Com
mand
Bars(
1).Co
ntrols
(6).E
nable
d =
True
End
Sub
```

2 - Rendre inaccessible le menu **Outils, Protection**.

```
Le
code
pour
rendr
e
inacc
essib
le
l'item
Outil
s,
Prote
ction
.
Sub
Inhib
eOuti
lsProt
ectio
n()
Com
mand
Bars(
1).Co
ntrols
(6).C
ontrol
s(6).
Enabl
ed =
False
End
Sub
Le
code
pour
rendr
e
acce
ssibl
e
l'item
Outil
s,
```

```
Protection
.
Sub
AfficheOutilsProtection()
CommandBars(1).Controls(6).Enabled = True
End
Sub
```

Supprimer l'affichage d'un menu (ou réafficher).

Supprimer l'intitulé **Outils** de la barre de menus.

```
Le code pour supprimer l'intitulé Outils de la barre de menus.
Sub
SupprimeIntituléOutils()
CommandBars(1).Visible = False
End
Sub
Le code pour réafficher l'intitulé Outils dans la barre de menus.
Sub
ReinstalleIntituléOutils()
Command
```

Bars(
1),Co
ntrols
(6).Vi
sible
=
True
End
Sub

Sélection dans une feuille de calcul (à l'aide du code VBA)

Page mise à jour le : 28/03/2001

Vous trouverez dans cette rubrique des exemples de programmes VBA pour effectuer différents types de sélection de cellules.

Sélection d'une plage avec la souris.

Sélection d'une plage.

Sélection de 2 plages (non contigües).

Sélection de deux colonnes non contigües.

Sélection de cellules caractéristiques (par ex. cellules égales à 20).

Sélection de cellules vers le bas depuis la cellule active (Ctrl+Shift+Bas).

Sélection de cellules vers le haut depuis la cellule active (Ctrl+Shift+Haut).

Sélection de cellules vers la droite depuis la cellule active (Ctrl+Shift+Droit).

Sélection de cellules vers la gauche depuis la cellule active (Ctrl+Shift+Gauche).

Sélection de cellules dans la plage courante depuis la cellule active.

Sélection de cellules contigües dans la colonne de la cellule active.

Sélection de cellules contigües dans la ligne de la cellule active.

Sélection d'une colonne entière par rapport à la cellule active.

Sélection d'une colonne entière par rapport à la cellule active.

Sélection d'une plage avec la souris.

Perm
et de
sélec
tionn
er
une
plage
avec
la
souri
s
(Méth
ode
Input
Box)
Sub
Selec
tionPl
ageA
vecS
ouris(
)
D

```

im
Plage
As
Rang
e
S
et
Plage
=
Appli
catio
n.Inp
utBox
("Sél
ectio
nnez
une
plage
!",
"Séle
ction
de
cellul
es",
Type:
=8)
M
sgBo
x
("La
plage
que
vous
avez
sélec
tionn
ez
est : "
&
Plage
.Address)
End
Sub

```

Sélection d'une plage.

```

Perm
et de
sélec
tionn
er la
zone
en
cours
par
rappo
rt à la
cellul
e
activ
e.
Sub
Selec
tionPl
age()
A
ctive
Cell.
Curre
ntRe
gion.
Selec
t
M
sgBo
x
("La
plage
sélec
tionn
ée
est : "
&
Selec
tion.A
ddres

```

```
s)
End
Sub
```

Sélection de 2 plages (non contigües).

Permet de sélectionner deux plages discontinues (Plage A1:Ax et C1:Cx)

```
Sub SelectionDiscontinue()
    Dim Z1, Z2, MaPlageMultiZone As Range
    Worksheets("Feuil1").Select
    Range("A1").Select
    ActiveCell.End(xlDown).Select
    Zone1 = ActiveCell.Address
    Selection.Offset(0, 2).Select
    Zone2 = ActiveCell.Address
    Set Z1 = Range("A1", Zone1)
    Set Z2 = Range("C1", Zone2)
    Set MaPlageMultiZone = Union(Z1, Z2)
    ZoneSelection = MaPlageMultiZone.Select
End Sub
```

Sélection de deux colonnes non contigües.

```
Perm
et de
sélec
tionn
er
deux
colon
nes A
et D
par
exem
ple.
Sub
Selec
tionD
euxC
olonn
esNo
nCon
tigüe
s()
' Soit
les
colon
nes A
(1) et
D (4)
à
sélec
tionn
er
    N
    Col1
    = 1
    N
    Col2
    = 4
    U
    nion(
    Cells(
    1,
    NCol
    1),
    Cells(
    1,
    NCol
    2)).E
    ntire
    Colu
    mn.S
    elect
    End
    Sub
```

Sélection de cellules caractéristiques (par ex. cellules égales à 20).

```
Perm
et de
sélec
tionn
er
toute
s les
```

```

cellul
es
dont
la
valeu
r est
20
Sub
Cellul
esVal
eurD
eterm
inee()
' La
valeu
r 20
est
saisie
en
E2
    L
aVale
ur =
Rang
e("E2
").Val
ue
    R
ange(
"A1").
Selec
t
    F
or
Each
Cll In
Activ
eCell.
Curre
ntRe
gion

    If
Cll.V
alue
=
LaVal
eur
Then
plg =
plg &
Cll.A
ddres
s() &
","
    N
ext
Cll
    If
Len(p
lg) >
0
Then
Rang
e(Left
(plg,
Len(p
lg) -
1)).S
elect
End
Sub

```

Sélection de cellules vers le bas depuis la cellule active.

```

Sub SelectDown()
    Range(ActiveCell, ActiveCell.End(xlDown)).Select
End Sub

```

Sélection de cellules vers le haut depuis la cellule active.

```

Sub
Selec
tDow

```

```

n()
Rang
e(Acti
veCel
l,
Activ
eCell.
End(
xlUp)
).Sel
ect
End
Sub

```

Sélection de cellules vers la droite depuis la cellule active.

```

Sub SelectDown()
    Range(ActiveCell, ActiveCell.End(xlToLeft)).Select
End Sub

```

Sélection de cellules vers la gauche depuis la cellule active.

```

Sub
Select
tDow
n()
    R
ange(
Activ
eCell,
Activ
eCell.
End(
xlToR
ight)).
Select
t
End
Sub

```

Sélection de cellules dans la plage courante depuis la cellule active.

```

Sub SelectCurrentRegion()
    ActiveCell.CurrentRegion.Select
End Sub

```

Sélection de cellules contigues dans la colonne de la cellule active.

```

Sub SelectActiveColumn()
    If IsEmpty(ActiveCell) Then Exit Sub
    ' Ignore error si Activecell est dans la ligne 1
    On Error Resume Next
    If IsEmpty(ActiveCell.Offset(-1, 0)) Then Set TopCell = ActiveCell Else Set TopCell = ActiveCell.End(xlUp)
    If IsEmpty(ActiveCell.Offset(1, 0)) Then Set BottomCell = ActiveCell Else Set BottomCell = ActiveCell.End(xlDown)
    Range(TopCell, BottomCell).Select
End Sub

```

Sélection de cellules contigues dans la ligne de la cellule active.

```

Sub
Select
tActiv
eRow
()
    If
IsEmpty
pty(A
ctive
Cell)
Then
Exit
Sub
'
Ignore
error
si
Activ
ecell
est
dans
la
colon
ne A

```

```

    O
    n
    Error
    Resu
    me
    Next
    If
    IsEm
    pty(A
    ctive
    Cell.
    Offse
    t(-1,
    0))
    Then
    Set
    TopC
    ell =
    Activ
    eCell
    Else
    Set
    TopC
    ell =
    Activ
    eCell.
    End(
    xlToL
    eft)
    If
    IsEm
    pty(A
    ctive
    Cell.
    Offse
    t(1,
    0))
    Then
    Set
    Botto
    mCell
    =
    Activ
    eCell
    Else
    Set
    Botto
    mCell
    =
    Activ
    eCell.
    End(
    xlToR
    ight)
    R
    ange(
    TopC
    ell,
    Botto
    mCell
    ).Sel
    ect
    End
    Sub

```

Sélection d'une colonne entière par rapport à la cellule active.

```

Sub SelectEntireColumn()
    Selection.EntireColumn.Select
End Sub End Sub

```

Sélection d'une ligne entière par rapport à la cellule active.

```

Sub SelectEntireRow()
    Selection.EntireRow.Select
End Sub

```

Commandes, codes, fonctions, etc ... à utiliser dans vos applications

Page mise à jour le : 28/03/2001

Adapter la largeur d'une colonne en fonction de son contenu.

```
Columns("A").AutoFit
```

Afficher un message ordinaire.

```
MsgBox("Vous êtes le bienvenu")
```

Afficher un message dans la barre de status.

```
Application.StatusBar="Programme en cours d'exécution ..."
```

Copier les données issues d'un filtre automatique vers une autre feuille.

```
Sheets("Feuil1").AutoFilter.Range.Copy Sheets("Feuil2").Range("A1")
```

ou

```
Sheets("Feuil1").Range("_FilterDatabase").Copy Sheets("Feuil2").Range("A1")
```

Définir l'imprimante par défaut.

```
ActivePrinter="Canon LBP-465 sur LPT1"
```

Définir la ligne de titre.

```
ActiveSheet.PageSetUp.PrintTitleRows="$1$1"
```

Délimiter du texte dans une cellule.

```
ActiveCell.WrapText=True
```

Documenter une fonction personnalisée.

```
Application.MacroOptions "InsereCheminClasseur", "Récupère le chemin du  
classeur."
```

InsereCheminClasseur : Nom de la fonction

Récupère le chemin du classeur : Définition de la fonction

Figurer le défilement de l'écran.

```
Application.ScreenUpdating=False
```

Insérer un commentaire dans une cellule (non VBA).

```
=SOMME(A10:A11)+N("Ceci est un commentaire")
```

Insérer la formule SOMME à une position variable.

```
Range("A1").End(xlDown).Offset(1, 0).Value = "=SUM(A1:" & Range_  
("A1").End(xlDown).Address & ")"
```

Redéfinir la plage pour utiliser la combinaison de touches Ctrl+Fin.

Lorsque l'on supprime des lignes ou colonnes, Excel garde toujours la même référence de la dernière cellule utilisée bien qu'elle soit vide.

Pour remédier à cela, une ligne de code :

```
ActiveSheet.UsedRange
```

Sélectionner la plage utilisée dans une feuille de calcul.

```
ActiveSheet.UsedRange.Select
```

Sélectionner la plage en cours sans prendre en compte la ligne de titres.

```
With Selection.CurrentRegion
```

```
Intersect(.Cells, .Offset(1)).Select
```

```
End With
```

Sélectionner la zone en cours.

Nota : La zone en cours est une plage limitée par toute combinaison de lignes et de colonnes vides.

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

Somme d'une plage variable (Non VBA).

Nota : La plage se situe dans la colonne A (A1 à Ax) et celle-ci possède un titre.

```
=SOMME(DECALER(A2;NB(A:A)-1;;-NB(A:A)-1))
```

Trouver la première ligne vide.

Cette commande permet de trouver dans une plage, la première ligne vide.

`NoLgn = ActiveSheet.UsedRange.Row + ActiveSheet.UsedRange.Rows.Count`

Cette commande permet de trouver dans une colonne (A), la première ligne vide.

`premLiVide = Columns(1).Find("", [A65536], , , xlByRows, xlNext).Row`

Des programmes de toutes sortes

Page mise à jour le : 3/05/2002

Récupérer le contenu d'une variable d'environnement (SET).

Récupérer le nom de login (réseau).

Récupérer une valeur stockée dans la base de registre.

Récupérer le contenu d'une variable d'environnement (SET).

Ce programme permet de récupérer une variable d'environnement (voir DOS pour détail sur commande SET).

Dans cet exemple, la variable AU_LOCAL_DIR me retourne un chemin (différent suivant les utilisateurs) où je trouve un fichier à traiter avec Excel.

Ce chemin est du type : AU_LOCAL_DIR="f:\renaud\audros\tmp"

Le code

Code de la procédure RechercheVariableEnvironnement

```
Sub  
Rec  
her  
che  
Vari  
abl  
eEn  
viro  
nne  
me  
nt()  
,
```

```
    Déc  
    lara  
    tion  
    des  
    vari  
    able  
    s.
```

```
    Dim  
    Env  
    Stri  
    ng,  
    Indx  
    ,  
    Msg  
    ,  
    Path
```

Len

Indx
= 1
,

Initi
alis
e
l'ind
ex à
1

Do

,
Extr
ait
la
vari
able
d'e
nvir
onn
em
ent
(AU
_LO
_CAL
_DI
_R)

Env
Stri
ng
=

Envi
ron
(Ind
x)

,
Véri
fie
l'en
trée
AU_
LOC
AL_
DIR

If
Left
(Env
Stri
ng,
13)
=

"AU
_LO
_CAL
_DI
_R="

The
n

,
Extr
ait
la
lon
gue
ur

*et
met
l'en
trée
dan
s
une
vari
able
.*

```
Path  
Len  
=  
Len  
( En  
viro  
n(  
"AU  
_LO  
_CAL  
_DI  
_R" ))
```

```
Msg  
=  
Mid  
(Env  
Stri  
ng,  
14,  
Path  
Len)
```

```
Exit  
Do
```

```
Else
```

*,
Pas
d'e
ntrée
e
AU_
LOC
AL_
DIR,
don
c on
incr
ém
ent
e.*

```
Indx  
=  
Indx  
+ 1
```

```
End  
If
```

```
Loop  
Until  
|
```

```

Env
Stri
ng
= ""
If
Path
Len
> 0
The
n

Msg
Box
Msg
.
Affi
che
le
mes
sag
e
"f:\r
ena
ud\
aud
ros\
tmp
"

Else

Msg
Box
" Il
n'ex
iste
pas
de
vari
able
" &
"d'e
nvir
onn
eme
nt
AU_
LOC
AL_
DIR.
"

End
If
End
Sub

```

Récupérer le nom de login (réseau).

Cette fonction permet de récupérer le nom de login (et non pas le nom d'utilisateur déclaré dans Excel).

Le code

[Code de la fonction GetUserName](#)

```

Decl
are

```

```
Function
WN
etGetUserLib
"mpr.dll"
Alias
"WNetGetUserA" (
    ByVal lpName,
    As String,
    -
    ByVal lpUser,
    As String,
    lpLength,
    As Long) As Long
Const NoError = 0
Function GetUserName()
Const lpLength As Integer = 255
Dim status As
```

Integer

Dimension
Name

,
User
Name

AsString

User
Name

=
Space\$(
Length
+ 1
)

status
=
WaitGetU
ser(I
pName,
User
Name
,
Length
)

If
status
=
NoError
Then

User
Name

=
Left\$
(User
Name,
InStr
(User
Name,
Chr
(0)
) - 1
)

Else

```
Msg  
Box  
"Im  
pos  
sibl  
e  
d'ob  
teni  
r le  
logi  
n."
```

```
End
```

```
End  
If
```

```
Get  
Use  
rNa  
me  
=  
IpUs  
erN  
ame  
End  
Fun  
ctio  
n
```

Code de la procédure AfficheLogin

```
Sub AfficheLogin()  
MsgBox GetUserName  
End Sub
```

Récupérer une valeur stockée dans la base de registre.

Cette procédure permet de récupérer une valeur stockée dans la base de registre.
Cette valeur est stockée dans la clé :

HKEY_CURRENT_USER\Software\VB and VBA Program Settings\

J'utilise personnellement cette procédure pour incrémenter un numéro de facture.

Lecture/Ecriture dans la base de registre

```
Priv  
ate  
Sub  
Lec  
tur  
eRe  
gist  
ry()  
,
```

```
Lect  
ure  
des  
info  
s  
dan  
s la  
bas  
e  
de  
regi  
stre
```

Call
Info
Regi
stry

Réc
upé
rati
on
du
N°
de
fact
ure

Nu
mFa
c =
Get
Sett
ing
(ap
pna
me:
=

"Fac
ture
Gra
nd
Crès
",
sect
ion:
=

"Nu
mFa
ctur
e",
key:
=
"Nu
mér
o")

Incr
ém
ent
atio
n
du
N°
de
fact
ure

Nu
mFa
c =
Nu
mFa
c +
1

Ecri
ture
de
la
nou
vell
e
vale
ur
Nu
mF

*ac
dan
s la
bas
e
de
regi
stre*

**Sav
eSe
ttin
g**
app
nam
e:=
"Fac
ture
Gra
nd
Crès
",
sect
ion:
=
"Nu
mFa
ctur
e",
key:
=
"Nu
mér
o",
setti
ng:
=Nu
mFa
c
**End
Sub**

**Priv
ate
Sub
Info
Reg
istr
y()**

*Lect
ure
de
la
vale
ur
dan
s la
bas
e
de
regi
stre*

Nu
mFa
c =
**Get
Sett
ing**
(ap
pna
me:

```
=  
"Fac  
ture  
Gra  
nd  
Crès  
",  
sect  
ion:  
=  
"Nu  
mFa  
ctur  
e",  
key:  
=  
"Nu  
mér  
o" )  
,
```

```
Si  
auc  
une  
vale  
ur  
n'es  
t  
prés  
ente  
dans  
la  
base  
de  
regi  
stre  
...
```

```
If  
Nu  
mFa  
c =  
""  
The  
n
```

```
Nu  
mde  
part  
=  
Inpu  
tBox  
( "In  
diqu  
er le  
N°  
de  
votr  
e  
der  
nièr  
e  
fact  
ure  
man  
uell  
e.",  
"Init  
ialis  
atio  
n du  
num  
éro
```

```
de  
fact  
ure"  
)
```

```
' Si  
on  
cliq  
ue  
sur  
le  
bou  
ton  
Ann  
uler  
, fin  
de  
pro  
céd  
ure
```

```
If  
Nu  
mde  
part  
= ""  
The  
n  
End
```

```
'Sin  
on,  
on  
insc  
rit  
le  
nu  
mér  
o  
sais  
i  
dan  
s la  
boît  
e  
de  
dial  
ogu  
e
```

```
Sav  
eSe  
ttin  
g  
app  
nam  
e:=  
"Fac  
ture  
Gra  
nd  
Crès  
",  
sect  
ion:  
=  
"Nu  
mFa  
ctur  
e",  
key:  
=
```

"Nu
mér
o",
setti
ng:
=Nu
mde
part

End
If
End
Sub

Effacement de la clé dans la base de registre

Sub
Effa
ceR
egi
stry
()

*Effa
cem
ent
de
la
clé
dan
s la
bas
e
de
regi
stre*

On
Erro
r
Res
ume
Nex
t

Del
ete
Sett
ing
"Fac
ture
Gra
nd
Crès
"

End
Sub

Echanges de données entre Excel et Word

Page mise à jour le : 28/03/2001

Vous trouverez dans cette rubrique des exemples de programmes permettant d'envoyer des données Excel vers Word ou récupérer des données Word dans Excel.

Ces exemples fonctionnent avec Word et Excel 2000. Ils n'ont pas été testés sous la version 97.

Pour que ces exemples fonctionnent, vous devrez créer une référence de bibliothèques d'objets de Word.

Dans Excel, choisissez la commande **Références** du menu **Outils** de Visual Basic Editor et cochez **Microsoft Word 9.0 Object Library**.

Récupérer les données d'un document Word et les copier dans Excel.

Copier des données Excel dans un document Word (format tableau).

Copier des données Excel dans un nouveau document Word (format tabulation).

Copier des données Excel dans un tableau Word afin de le compléter.

Récupérer les données d'un document Word et les copier dans Excel

Ce programme permet d'aller chercher des données dans un document Word, de les copier, de les coller dans Excel et de refermer Word.

Sub DonnéesWordVersExcel()

```
Dim DocWord As Word.Document
```

```
Dim AppWord As Word.Application
```

```
Set AppWord = New Word.Application
```

```
AppWord.ShowMe
```

```
AppWord.Visible = True
```

```
'Ouvre le document Word (Fichier.doc) et effectue une copie des données
```

```
Set DocWord = AppWord.Documents.Open( "c:\Excel\Fichier.doc" , ReadOnly:= True )
```

```
With AppWord
```

```
    .Selection.WholeStory
```

```
    .Selection.Copy
```

```
End With
```

```
' Copie des données dans Excel
```

```
ThisWorkbook.Worksheets( "Feuil1" ).Paste
```

```
' Fermeture de Word
```

```
AppWord.Application.Quit
```

```
Application.CutCopyMode = False
```

```
End Sub
```

Les données Word La copie dans Excel

Copier des données Excel dans un document Word (format tableau)

Ce
programm
e perm
et de
copier
des
donn
ées
d'Exc
el,
d'ouv
rir un
docu
ment
Word
et d'y
coller
les
donn

ées
sous
forme
de
table
au.

Sub
Env
oye
rDo
nne
esE
xce
lVer
sW
ord
()

Dim
Doc
Wor
d **As**
Wor
d.D
ocu
men
t

Dim
App
Wor
d **As**
Wor
d.A
ppli
cati
on

Set
App
Wor
d =
New
Wor
d.A
ppli
cati
on

Appl
icati
on.
Disp
layA
lerts
=
True

App
Wor
d.Sh
ow
Me

App
Wor
d.Vi
sibl
e =
True

*'Ou
vre
le
doc*

*um
ent
Wor
d*

Set
Doc
Wor
d =
App
Wor
d.D
ocu
men
ts.O
pen
("c:
\exc
el\Fi
chie
r.do
c" ,
Rea
dOn
ly:=
**Fals
e**)
,

*Cop
ie
les
don
née
s
Exc
el*

This
Wor
kbo
ok.
Wor
ksh
eets
("Fe
uil1
").R
ang
e("
A1:
B6"
).Co
py
,

*Coll
e
les
don
née
s
dan
s
Wor
d*

Doc
Wor
d.Ra
nge.
Past
eSp
ecia
l

```
Appl  
icati  
on.  
Cut  
Cop  
yMo  
de  
=  
Fals  
e
```

```
Doc  
Wor  
d.A  
ppli  
cati  
on.A  
ctiv  
eDo  
cum  
ent.  
Sav  
e
```

```
App  
Wor  
d.A  
ppli  
cati  
on.  
Quit  
End  
Sub
```

Les La
do cop
nn ie
ées da
Exc ns
el Wo
rd

Copier des données Excel dans un document Word (format tabulations)

Ce programme permet de copier des données d'Excel, d'ouvrir un document Word et d'y coller les données. Les données étant séparées par des tabulations.

Sub CopierDonneesExcelVersWord()

```
Range( "A1:B6" ).Copy  
Set WW = CreateObject( "word.application" )  
WW.Visible = True  
WW.Documents.Add  
WW.Selection.PasteSpecial Link:= False , DataType:=wdPasteText, Placement:= _  
wdInLine, DisplayAsIcon:= False
```

End Sub

Les La
do cop
nn ie
ées da
Exc ns
el Wo
rd

Copier des données Excel dans un tableau Word afin de le compléter

Ce programme permet de copier des données d'Excel, d'ouvrir un document Word existant, d'ajouter une ligne au tableau existant dans Word et de compléter chaque cellule de la ligne créer.

```

Sub RemplirTableauWordDepuisDonnéesExcel()
  Dim DocWord As Word.Document
  Dim AppWord As Word.Application
  Set AppWord = New Word.Application
  Application.DisplayAlerts = True
  AppWord.ShowMe
  AppWord.Visible = True
  'Ouvre le document Word
  Set DocWord = AppWord.Documents.Open( "c:\excel\Fichier.doc" , _
  ReadOnly:= False )
  ' Copie les données Excel
  Contrats_ISBN = Range( "A2" ).Value
  Contrats_Titre = Range( "B2" ).Value
  Contrats_Nom = Range( "C2" ).Value
  ' Colle les données dans Word
  DocWord.Tables( 1 ).Rows.Add
  Derligne = DocWord.Tables( 1 ).Rows.Count
  With DocWord.Tables( 1 )
    .Cell(Derligne, 1 ).Range.InsertAfter Contrats_ISBN
    .Cell(Derligne, 2 ).Range.InsertAfter Contrats_Titre
    .Cell(Derligne, 3 ).Range.InsertAfter Contrats_Nom
  End With
  DocWord.Application.ActiveDocument.Save
  AppWord.Application.Quit
End Sub

```

Le tableau Word avant mise à jour

Les données Excel à copier dans le tableau Word

Le tableau Word après mise à jour

Echanges de données entre Excel et Access

Page mise à jour le : 1/08/2002

Vous trouverez dans cette rubrique des exemples de programmes permettant d'envoyer des données Excel vers Access, de récupérer des données d'Access dans Excel ou de mettre à jour des enregistrements dans Access depuis Excel.

Ces exemples fonctionnent avec Access et Excel 2000. Ils n'ont pas été testés sous la version 97.

Pour que ces exemples fonctionnent, vous devrez créer une référence de bibliothèques d'objets d'accès aux données de Microsoft (DAO). Dans Excel, choisissez la commande **Références** du menu **Outils** de Visual Basic Editor et cochez **Microsoft DAO 3.6 Object Library**.

Un lien intéressant (mes sources) traitant de [External Data Access](#) with DAO (Data Access Objects) and ADO (ActiveX Data Objects)

Envoyer les données d'une feuille Excel vers une table Access.

Récupérer les données d'une table Access dans une feuille de calcul Excel.

Récupérer les données d'une table Access dans une feuille de calcul Excel via une

requête .
Mettre à jour un enregistrement dans Access depuis Excel.

Envoyer les données d'une feuille Excel vers une table Access

Ce programme permet de copier les données d'une feuille de calcul Excel vers une table Access.

Définition des éléments utilisés :

Nom de la base de données --> **Commandes.mdb**
Table recevant les données Excel --> **Factures**
Champs de la table Factures --> **NoFacture, Client, Date et Solde**
Feuille de calcul Excel d'où l'on exporte les données --> **DAOSheet**

Sub WritingWorksheetData_DAO()

```
Dim Plage As Range
Dim Array1 As Variant
Dim x As Variant
Dim Db1 As Database
Dim Rs1 As Recordset
' Ouverture de la base de données Commandes.mdb
Set Db1 = DBEngine.OpenDatabase(ThisWorkbook.Path & "\Commandes.mdb")
' Ouverture de la table Factures
' Un objet Recordset représente les enregistrements d'une table
Set Rs1 = Db1.OpenRecordset("Factures", dbOpenDynaset)
' Détermination de la taille de la plage à envoyer vers Access
Set Plage = Worksheets("DAOSheet").Range("A1").CurrentRegion.Offset(1, 0)
Set Plage = Plage.Resize(Plage.Rows.Count - 1, Plage.Columns.Count)
Plage.Select
' Lecture de la plage pour renvoyer une valeur contenant un tableau
Array1 = Plage.Value
' Ecriture des données depuis Excel vers les enregistrement de la table Factures
For x = 1 To UBound(Array1, 1)
    With Rs1
        .AddNew
        .Fields("NoFacture") = Array1(x, 1)
        .Fields("Client") = Array1(x, 2)
        .Fields("Date") = Array1(x, 3)
        .Fields("Solde") = Array1(x, 4)
        .Update
    End With
Next
' Fermeture de la base Commandes.mdb
Db1.Close
' Effacement des données copiées vers la base (sauf les titres)
With Selection.CurrentRegion
    Intersect(.Cells, .Offset(1)).Select
End With
Selection.ClearContents
End Sub
```

Les
do
nn
ées
Exc
el

Les
données
copiées
dans la

table Factures

Récupérer les données d'une table Access dans une feuille de calcul Excel

Ce programme permet de récupérer les données d'une table Access
et de les copier dans une feuille Excel.

Définition des éléments utilisés :

Nom de la base de données --> **Commandes.mdb**

Table possédant les données à récupérer dans Excel --> **Factures**

Champs de la table Factures --> **NoFacture, Client, Date et Solde**

Feuille de calcul Excel recevant les données de la table; --> **DonnéesDataBase**

Sub CopyFromRecordset_DAO()

Dim Db1 **As** Database

Dim Rs1 **As** Recordset

' Ouverture de la base de données

Set Db1 = DBEngine.OpenDatabase(ThisWorkbook.Path & "\Commandes.mdb")

' Ouverture de la table Factures

' Un objet Recordset représente les enregistrements d'une table

Set Rs1 = Db1.OpenRecordset(**Name** := "Factures" , **Type** :=dbOpenDynaset)

' Effacement des données existantes dans la WorkSheet (sauf les titres)

' et copie des enregistrements

With Worksheets("DonnéesDataBase").Range("A2")

With Selection.CurrentRegion

Intersect(.Cells, .Offset(1)).Select

End With

Selection.ClearContents

.CopyFromRecordset Rs1

End With

'Fermeture de la Base de données

Db1.Close

End Sub

Les données de la table Factures

Les données récupérées dans la feuille Excel

Récupérer les données d'une table Access dans une feuille de calcul Excel via une requête

Ce
progr
amm
e
perm
et de
récup
érer
les
donn
ées
d'une
table
Acce
ss
par
l'inter
médi
are
d'une
requê
te et
de
les
copie
r
dans
une

feuille
Excel
.

Définition
des
éléments
utilisés
s :

Nom
de la
base
de
données
-->

Commandes.mdb
Table
possédant
les
données à
récupérer
dans
Excel
-->

Factures
Champs
de la
table
Factures
-->

NoFacture
,
Cli
t,
Date
et
Solde
e
Nom
de la
requête -
->

Factures pour un client
Code
SQL
-->

SELECT
Factures.
NoFacture
,

Factures.
Cli
t,
Factures.
Date,
Factures.
Solde
FROM
M

Fact
ures

WHE
RE
(((Fa
cture
s.Cli
ent)
Like
"ALL
EE
DES
VINS
");
Feuill
e de
calcul
Excel
recev
ant
les
donn
ées
de la
table;
-->
Don
nées
Data
Base

**Sub
Cop
yFr
om
Rec
ord
set
_DA
O()**

Dim
Db1
As
Dat
aba
se

Dim
Rs1
As
Rec
ords
et

*Ouv
ertu
re
de
la
bas
e
de
don
née
s*

Set
Db1
=
DBE
ngin
e.O

```
pen
Dat
aba
se(T
his
Wor
kbo
ok.P
ath
&
"\Co
mm
and
es.
mdb
") ,
Ouv
ertu
re
de
la
tabl
e
Fact
ure
s ,
Un
obje
t
Rec
ord
set
repr
ése
nte
les
enr
egis
tre
me
nts
d'u
ne
tabl
e
Set
Rs1
=
Db1
.Op
enR
ecor
dset
( Na
me :
=
"Fac
ture
s
pou
r un
clie
nt" ,
Typ
e :
:=d
bOp
enS
nap
shot
)
```

Effacement
des données
existantes
dans la
Worksheet
(sauf les
titres)

et
copie
des
enregistrements

With
Worksheets
("Database")
.Range("A2")
)

With
Selection
CurrentRegion

Intersect(.Cells,
.Offset(1)).
Select

End
With

Sele
ctio
n.Cl
ear
Con
tent
s

.Co
pyFr
omR
ecor
dset
Rs1

End
Wit
h

'Fer
met
ure
de
la
Bas
e
de
don
née
s

Db1
.Clo
se
End
Sub

Les
do
nn
ées
de
la
tab
le
Fac
tur
es

Les données récupérées dans la feuille Excel

La requête Factures pour un client en mode Création

Mettre à jour un enregistrement dans Access depuis Excel

Ce
progr

amm
e
perm
et de
mettr
e à
jour
dans
la
table
Factu
res,
le
cham
p
(Sold
e)
corre
spon
dant
à
l'enre
gistre
ment
dont
le N°
de
factur
e est
égal
à la
varia
ble
N°Fa
cture.

Défini
tion
des
éléme
nts
utilisé

s :
Nom
de la
base
de
donn
ées
-->

**Com
man
des.
mdb**
Table
poss
édant
les
donn
ées à
mettr
e à
jour;
-->

**Fact
ures**
Cha
mp
de la
table
Factu
res à
mettr
e à
jour
-->

**Sold
e**
Rech
erche
à
partir
du
-->

**NoF
actur**

e
Note
: Afin
de
propo
ser
les 2
possi
bilités
de
synta
xe,
l'exe
mple
se
propo
se de
tester
quel
est le
type
de la
varia
ble
Nofa
cture
car la
synta
xe
est
différ
ente
suiva
nt
que
la
varia
ble
est
un
nomb
re
(vbInt
eger)
ou
une
chaîn
e
(vbSt
ring).

**Sub
Upd
ate
Dat
aAc
ces
s_D
AO(
)**

Dim
Db1
As
Dat
aba
se

Typ
eVar
iabl
e =
VarT
ype
(Nu
mFa
ctur
e)

Ouv
ertu
re
de
la
bas
e
de
don
née
s

Set
Db1
=
DBE
ngin
e.O
pen
Dat
aba
se(T
his
Wor
kbo
ok.P
ath
&
"Co
mm
and
es.
mdb
")
,

Mis
e à
jour
de
l'en
regi
stre
me
nt

If
Typ
eVar
iabl
e =
2
Then

' La
vari
able
Nu
mfa
ctur
e
est
de
typ
e
vbln
teg
er
(2)

Db1
.Exe
cute
"UP

```
DAT
E
Fact
ures
SET
Sold
e=
'Oui'
WH
ERE
NoF
actu
re =
" &
Nu
mFa
ctur
e
```

Else

```
' La
vari
able
Nu
mfa
ctur
e
est
de
typ
e
vbS
trin
g
(8)
```

```
Db1
.Exe
cute
"UP
DAT
E
Fact
ures
SET
Sold
e =
'Oui'
WH
ERE
NoF
actu
re =
' " &
Nu
mFa
ctur
e &
" ' "
```

End
If

```
Db1
.Clo
se
End
Sub
```

Téléchargement

Page mise à jour le : 23/04/2001

Vous trouverez sur cette page, des exemples présentés dans les différentes rubriques et disponibles en téléchargement.

Exemples présents sur le site Fichier Archive

Boite de dialogue avec case à cocher

Boite
de
dialo
gue
avec
bouto
ns
d'opti
ons

Boite
de
dialo
gue
avec
liste
dérou
lante

Boite
de
dialo
gue
avec
liste
dérou
lante
modif
iable

Boite
de
dialo
gue
avec
liste
dérou
lante
modif
iable
simpli
fiée

Boite
de
dialo
gue
avec
liste
dérou
lante
à
sélec
tion
multi
ple

Boite
de
dialo
gue
avec
liste
dérou

lante
à
deux
colon
nes

Boite
de
dialo
gue
avec
liste
dérou
lante
comb
inée

Le
contr
ôle
Cale
ndar
(Cale
ndrier
)

Le contrôle ProgressBar, version 6.0 

Supprimer les
doublons dans une
colonne