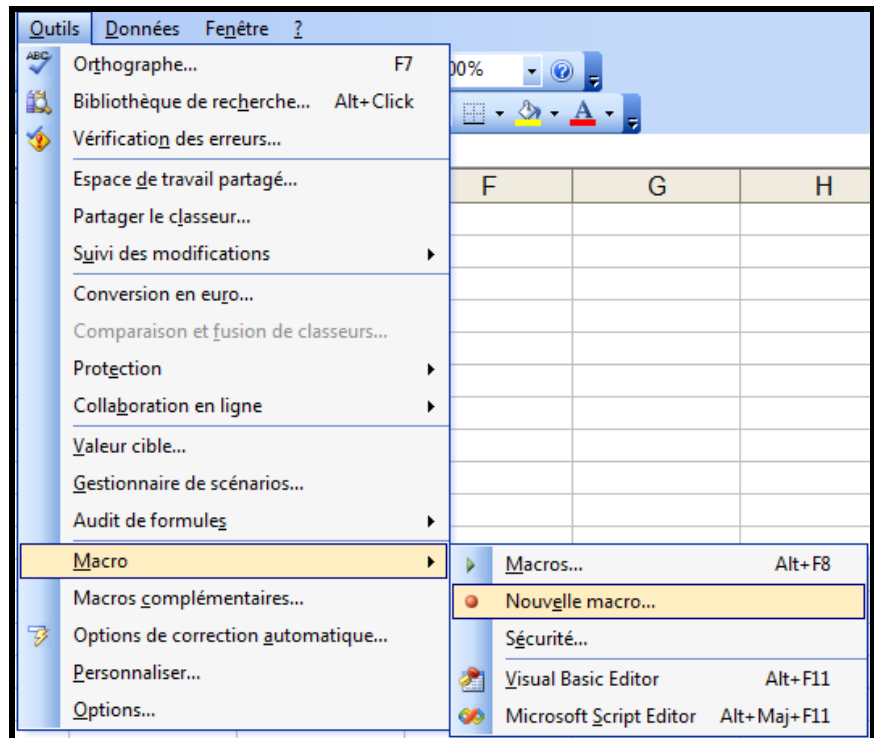


Sommaire

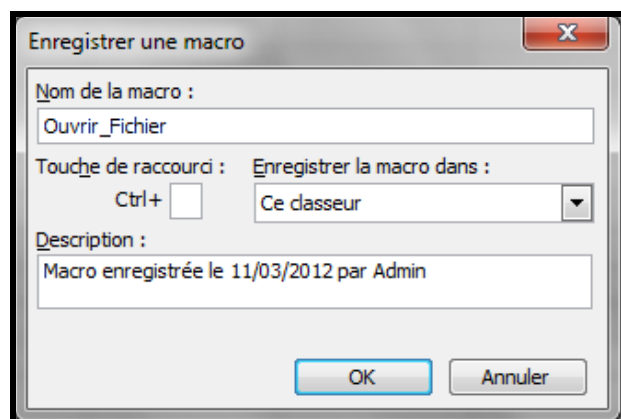
A) - CRÉER une MACRO avec la BARRE des MENUS	2
1) - CRÉATION AUTOMATIQUE d'une MACRO	2
2) - OUVRIR un FICHIER DÉFINI	4
3) - OUVRIR un NOUVEAU CLASSEUR VIDE	5
4) - FERMER un FICHIER SANS SAUVEGARDE	5
5) - FERMER un FICHIER AVEC SAUVEGARDE	5
6) - Fermer Fichier Excel en cours et Sauvegarde en changeant de Nom	6
7) - Fermer Fichier Excel en cours et Sauvegarde comme Modèle	6
8) - Sélection d'une cellule dans Fichier Excel	8
9) - Sélection d'un Groupe de cellules dans Fichier Excel	8
10) - Sélection de cellules non contiguës dans Fichier Excel	9
11) - Copier/Coller un groupe de cellules	9
12) - Couper/Coller un groupe de cellules	10
13) - Couper/Insérer un groupe de cellules	10
14) - Copier/Collage Spécial d'un groupe de cellules	12
15) - Largeur d'une Colonne	13
16) - Hauteur d'une Ligne	14
17) - Effacer une Cellule ou Ligne ou Colonne	14
18) - Sélectionner une Feuille dans un Fichier	15
19) - Insérer une Feuille dans un Classeur	15
20) - Renommer une Feuille dans un Classeur	16
21) - Déplacer une Feuille dans un Classeur	16
22) - Copier une Feuille dans un Classeur	16
23) - Supprimer une Feuille dans un Fichier	17
24) - Sélectionner un tableau entier dans un Fichier (Ctrl+*)	17
25) - Insérer une valeur précise dans une cellule	17
26) - Donner un NOM à une ou plusieurs cellules	18
27) - Protéger et déprotéger un classeur (sans mot de passe)	18
28) - Protéger et déprotéger un classeur (AVEC mot de passe)	19

A) - CRÉER une MACRO avec la BARRE des MENUS**1) - CRÉATION AUTOMATIQUE d'une MACRO**

Pour accéder à la création de macro, cliquez dans la barre des menus sur "**Outils**", puis sur "**Macro**", et enfin sur "**Nouvelle macro...**".



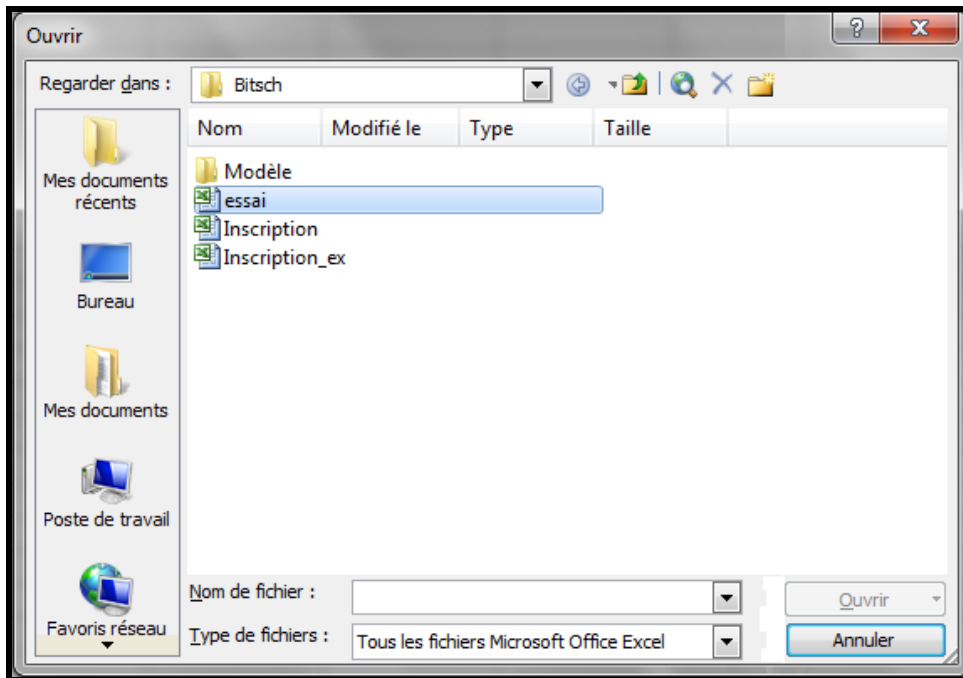
Dans la boîte de dialogue qui apparaît, entrez dans la zone "**Nom de la macro**", le titre que vous allez donner à votre macro (par exemple **Ouvrir_fichier**), puis dans la zone "**Enregistrer la macro dans**", l'endroit d'où sera accessible cette macro. Soit dans le *Classeur personnel*, soit dans un *nouveau classeur*, ou bien dans *ce classeur*. Nous choisissons cette dernière option et nous cliquons sur le bouton "**OK**" pour confirmer.



A partir de cet instant, toutes les commandes que vous effectuerez dans ce classeur d'Excel, seront interprétées en tant que commande de macros.



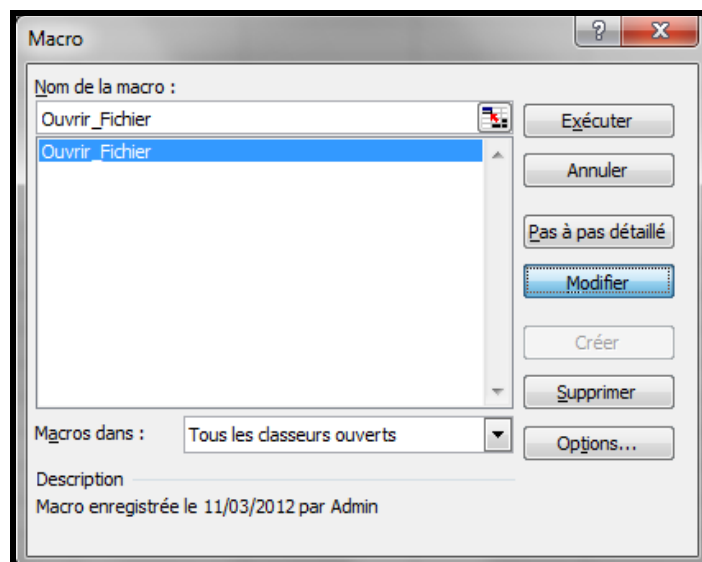
Une nouvelle barre d'outils apparaît sur votre feuille Excel avec le bouton "**Arrêter l'enregistrement**". Si vous cliquez dessus, vous mettez fin à l'enregistrement de vos commandes dans la macro

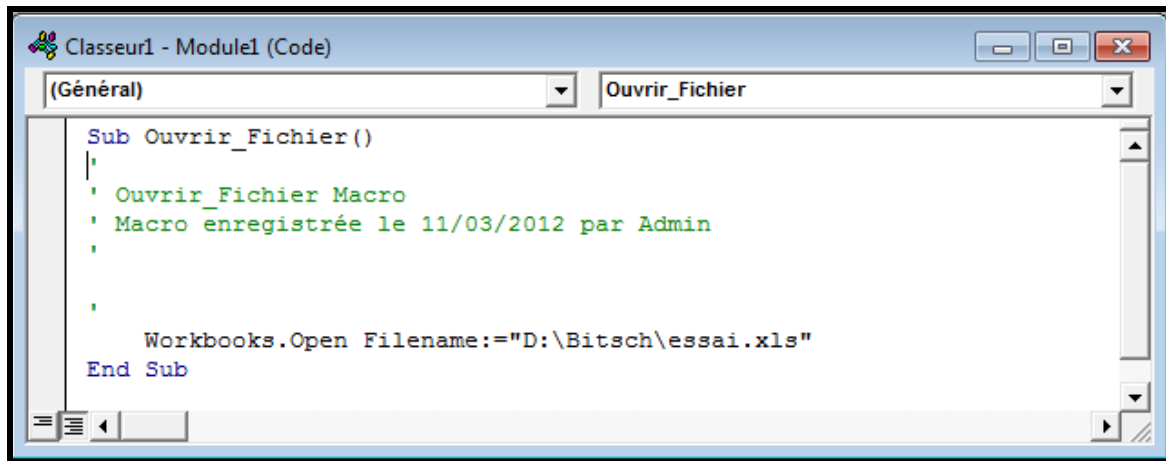


Dans notre exemple, nous avons cliqué sur "**Fichier**", puis "**Ouvrir**", puis nous avons choisi, dans la boîte de dialogue, le répertoire où se trouve notre fichier et enfin nous avons sélectionné le fichier que nous voulons ouvrir. Pour terminer, nous avons cliqué sur le bouton "**Ouvrir**".

Dès que le fichier apparaît à l'écran, cliquez sur le bouton "**Arrêter l'enregistrement**" dans la barre d'outils qui se trouve sur votre feuille Excel. Afin de voir le résultat des commandes qui ont été créées automatiquement par Excel pour effectuer cette tâche, cliquons dans la barre des menus sur "**Outils**", puis sur "**Macro**" et enfin sur "**Macros...**".

Dans la boîte de dialogue qui apparaît, choisir le nom de la macro dont nous voulons voir la syntaxe, puis cliquons sur le bouton "**Modifier**".



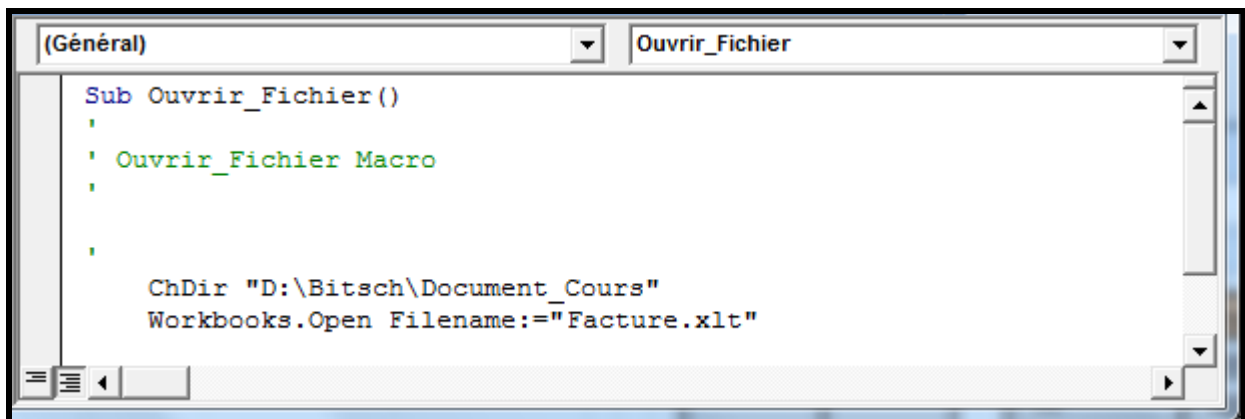


Nous pouvons voir que la commande permettant l'ouverture d'un fichier donné, dans un répertoire spécifique est définie par la commande **Worksheets.Open Filename := "Chemin\nom_fichier"**. Cette commande est contenue dans une structure "**Sub Nom_macro**" et se terminant par "**End Sub**".

Pour créer des macros simples, vous pouvez opérer de la même manière, c'est à dire en effectuant les commandes à l'aide des menus et de la souris. Par contre, si vous souhaitez introduire des variables dans ces macros, il vous faudra modifier ces dernières en y incluant les commandes permettant un choix au niveau de l'utilisateur et éventuellement la gestion des erreurs.

2) - OUVRIR un FICHIER DÉFINI

La commande **ChDir** permet de se placer dans un répertoire spécifique dont nous définissons le chemin. La commande **Worksheets.Open Filename:=** permet d'ouvrir un fichier spécifique dont nous indiquons le nom.



3) - *OUVRIR un NOUVEAU CLASSEUR VIDE*

Pour cela nous devons utiliser la commande **Workbooks.Add**.

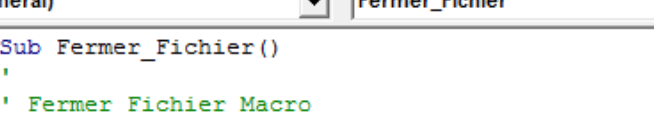


The screenshot shows the VBA editor with the 'Ouvrir_Classeur' macro selected in the 'Project Explorer' on the left. The 'Code' window on the right displays the following VBA code:

```
Sub Ouvrir_Classeur()  
'  
' Ouvrir_Classeur Macro  
'  
'  
'  
    Workbooks.Add  
End Sub
```

4) - **FERMER un FICHER SANS SAUVEGARDE**

Pour cela nous devons utiliser la commande **ActiveWorkbook.Close**



The screenshot shows the VBA editor window with the 'Fermer_Fichier' macro selected in the 'Project Explorer' on the left. The 'Code' window on the right displays the following VBA code:

```
Sub Fermer_Fichier()  
'  
' Fermer_Fichier Macro  
'  
'  
  
    Windows ("Classeur2").Activate  
    ActiveWorkbook.Close  
End Sub
```

5) - *FERMER un FICHER AVEC SAUVEGARDE*

Pour cela nous devons utiliser la commande **ActiveWorkbook.SaveAs**.

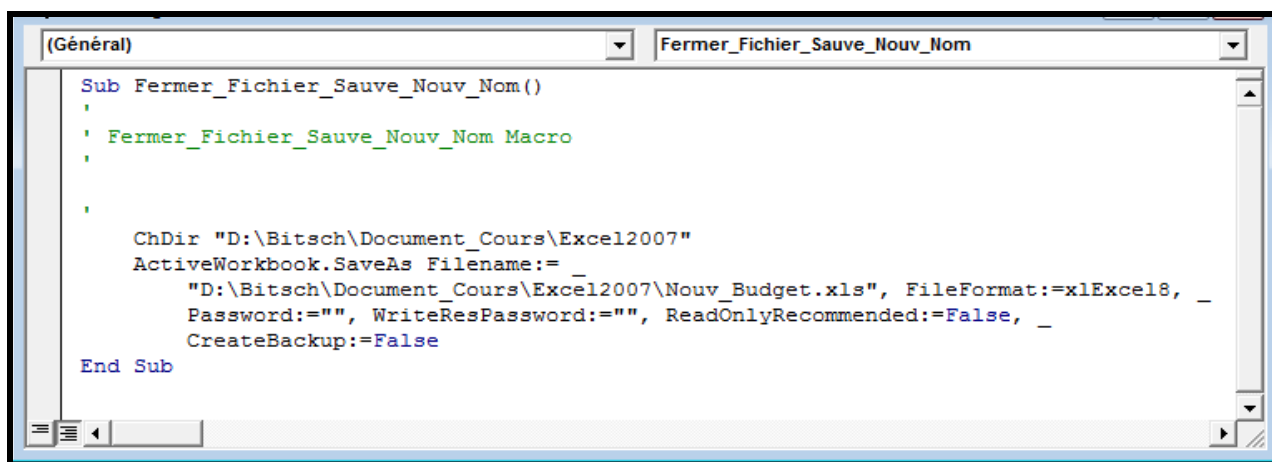


The screenshot shows the VBA editor with the 'Fermer_Fichier_Sauve' macro selected in the 'Project Explorer' on the left. The 'Properties Window' on the right shows the 'Macro' tab with the 'Name' set to 'Fermer_Fichier_Sauve' and the 'Source' set to 'ThisWorkbook'. The 'Code Window' on the right displays the following VBA code:

```
Sub Fermer_Fichier_Sauve()  
    ' Fermer_Fichier_Sauve Macro  
    '   
    '   
    Windows("budgets.xls").Activate  
    ActiveWorkbook.SaveAs Filename:="D:\Bitsch\Document_Cours\budgets.xls", _  
        FileFormat:=xlExcel8, Password:="", WriteResPassword:="", _  
        ReadOnlyRecommended:=False, CreateBackup:=False  
End Sub
```

6) - Fermer Fichier Excel en cours et Sauvegarde en changeant de Nom

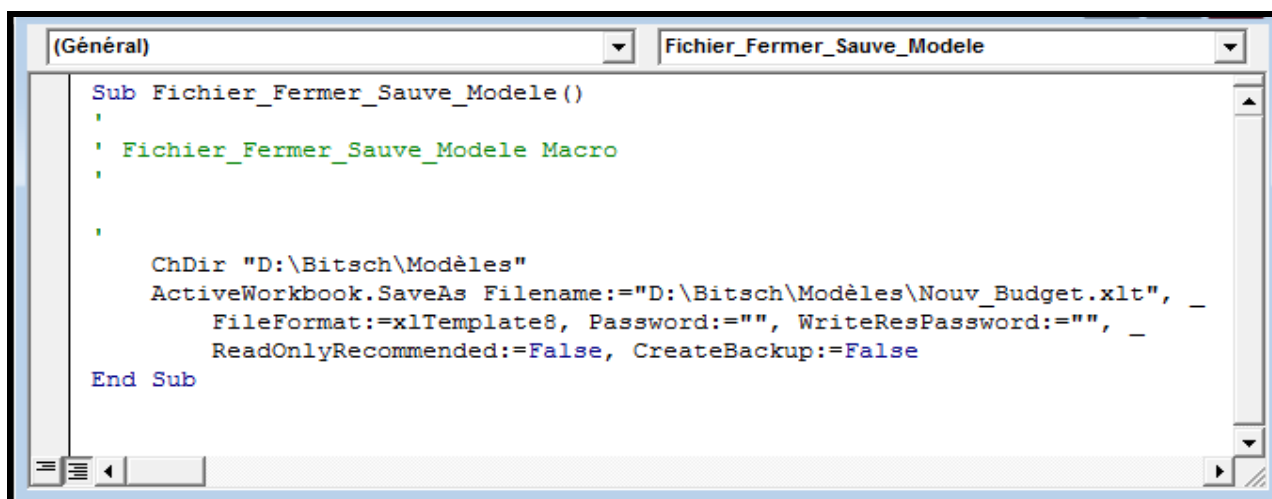
Dans notre exemple, nous commençons avec la commande **ChDir** à changer le répertoire où se fera la sauvegarde, puis avec la commande **ActiveWorkbook.SaveAs Filename:= "Nouveau_Nom _Fichier", FileFormat:=.....**



```
(Général) Fermer_Fichier_Sauve_Nouv_Nom

Sub Fermer_Fichier_Sauve_Nouv_Nom()
' Fermer_Fichier_Sauve_Nouv_Nom Macro
'

ChDir "D:\Bitsch\Document_Cours\Excel2007"
ActiveWorkbook.SaveAs Filename:= _
"D:\Bitsch\Document_Cours\Excel2007\Nouv_Budget.xls", FileFormat:=xlExcel8, _
Password:="", WriteResPassword:="", ReadOnlyRecommended:=False, _
CreateBackup:=False
End Sub
```

7) - Fermer Fichier Excel en cours et Sauvegarde comme Modèle

```
(Général) Fichier_Fermer_Sauve_Modele

Sub Fichier_Fermer_Sauve_Modele()
' Fichier_Fermer_Sauve_Modele Macro
'

ChDir "D:\Bitsch\Modèles"
ActiveWorkbook.SaveAs Filename:="D:\Bitsch\Modèles\Nouv_Budget.xlt", _
FileFormat:=xlTemplate8, Password:="", WriteResPassword:="", _
ReadOnlyRecommended:=False, CreateBackup:=False
End Sub
```

Ci-dessous sont indiquées les différentes valeurs possibles de **FileFormat**.

FileFormat, propriété	
Voir aussi	Exemple S'applique à
Renvoie le format du fichier et/ou le type du classeur. Argument de type Long en lecture seule. Il peut s'agir de l'une des constantes XlFileFormat suivantes :	
xlAddIn	xlTemplate
xlCSV	xlCurrentPlatformText
xlCSVMac	xlTextMac
xlCSVMSDOS	xlTextMSDOS
xlCSVWindows	xlTextPrinter
xlDBF2	xlTextWindows
xlDBF3	xlWJ2WD1
xlDBF4	xlWK1
xlDIF	xlWK1ALL
xlExcel2	xlWK1FMT
xlExcel2FarEast	xlWK3
xlExcel3	xlWK4
xlExcel4	xlWK3FMT
xlExcel5	xlWKS
xlExcel4Workbook	xlWorks2FarEast
xlIntlAddIn	xlWQ1
xlIntlMacro	xlWJ3
xlWorkbookNormal	xlWJ3FJ3
xlSYLK	

FileFormat, propriété - Exemple

Cet exemple montre comment enregistrer le classeur actif dans le format de fichier Normal si son format de fichier en cours est WK3.

```
If ActiveWorkbook.FileFormat = xlWK3 Then  
    ActiveWorkbook.SaveAs fileFormat:=xlNormal  
End If
```

Nous vous donnons également sur la page ci-dessous les explications concernant la syntaxe de la fonction **SaveAs**.

SaveAs, méthodeVoir aussi [Exemple](#) S'applique à

Enregistre les modifications de la feuille (Syntaxe 1) ou du classeur (Syntaxe 2) dans un fichier différent.

Syntaxe 1

expression.**SaveAs**(*Filename*, *FileFormat*, *Password*, *WriteResPassword*, *ReadOnlyRecommended*, *CreateBackup*, *AddToMru*, *TextCodePage*, *TextVisualLayout*)

Syntaxe 2

expression.**SaveAs**(*Filename*, *FileFormat*, *Password*, *WriteResPassword*, *ReadOnlyRecommended*, *CreateBackup*, *AccessMode*, *ConflictResolution*, *AddToMru*, *TextCodePage*, *TextVisualLayout*)

expression Obligatoire. Expression qui renvoie un objet **Chart** ou **Worksheet** (Syntaxe 1) ou un objet **Workbook** (Syntaxe 2).

Filename Argument de type **Variante** facultatif. Chaîne qui indique le nom du fichier à enregistrer. Vous pouvez inclure un chemin complet ; si vous ne le faites pas, Microsoft Excel enregistre le fichier dans le dossier en cours.

FileFormat Argument de type **Variante** facultatif. Format de fichier à utiliser quand vous enregistrez le fichier. Pour obtenir une liste des choix possibles, consultez la propriété **FileFormat**. Dans le cas d'un fichier existant, le format par défaut est le dernier format de fichier spécifié. Dans le cas d'un nouveau fichier, la valeur par défaut est le format de la version de Microsoft Excel en cours d'utilisation.

Password Argument de type **Variante** facultatif. Chaîne qui respecte la casse (maximum 15 caractères) et indique le mot de passe de protection à donner au fichier.

WriteResPassword Argument de type **Variante** facultatif. Chaîne qui indique le mot de passe de protection en écriture du fichier. Si un fichier est enregistré avec le mot de passe et que ce mot de passe n'est pas tapé à l'ouverture du fichier, le fichier s'ouvre en lecture seule.

ReadOnlyRecommended Argument de type **Variante** facultatif. Affectez-lui la valeur **True** pour afficher, à l'ouverture du fichier, un message qui conseille d'ouvrir celui-ci en lecture seule.

CreateBackup Argument de type **Variante** facultatif. Affectez-lui la valeur **True** pour créer un fichier de sauvegarde.

AccessMode Argument de type **Variante** facultatif. Mode d'accès au classeur. Il peut s'agir de l'une des constantes **xlSaveAsAccessMode** suivantes : **xlShared** (liste partagée), **xlExclusive** (mode exclusif) ou **xlNoChange** (mode d'accès non modifié). Si cet argument n'est pas spécifié, le mode d'accès n'est pas modifié. Cet argument est ignoré si vous enregistrez une liste partagée sans modifier le nom de fichier. Pour modifier le mode d'accès, utilisez la méthode **ExclusiveAccess**.

ConflictResolution Argument de type **Variante** facultatif. Spécifie la façon dont les conflits de modification sont résolus si le classeur est une liste partagée. Il peut s'agir de l'une des constantes **xlSaveConflictResolution** suivantes : **xlUserResolution** (affiche la boîte de dialogue résolution de conflit), **xlLocalSessionChanges** (accepte automatiquement les modifications de l'utilisateur local), ou **xlOtherSessionChanges** (accepte d'autres modifications au lieu de celles de l'utilisateur local). Si cet argument n'est pas spécifié, la boîte de dialogue de résolution de conflit s'affiche.

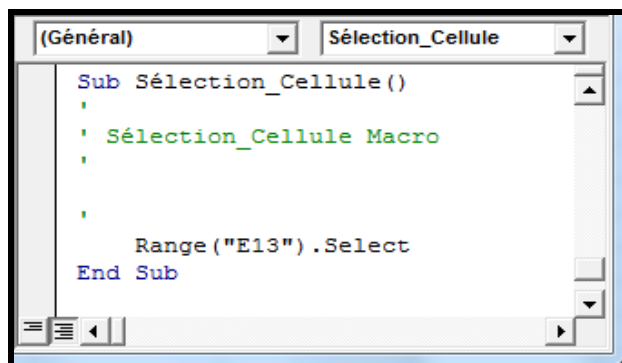
AddToMru Argument de type **Variante** facultatif. Affectez-lui la valeur **True** pour ajouter ce classeur à la liste des fichiers utilisés récemment. La valeur par défaut est **False**.

TextCodePage Argument de type **Variante** facultatif. Non utilisé dans la version française de Microsoft Excel.

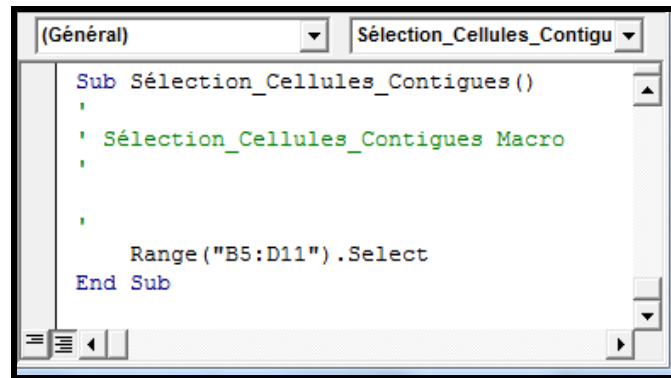
TextVisualLayout Argument de type **Variante** facultatif. Non utilisé dans la version française de Microsoft Excel.

8) - Sélection d'une cellule dans Fichier Excel

Pour cela nous devons utiliser la commande **Range()** associée avec **Select**. Dans les parenthèses, nous devons indiquer la référence de la cellule.

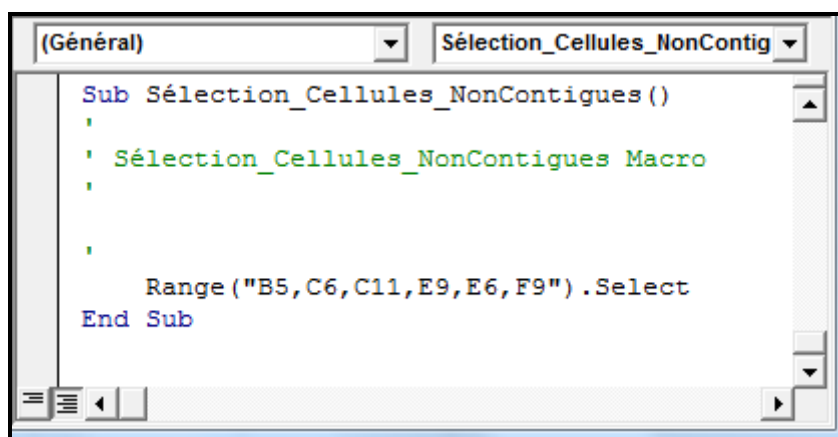
**9) - Sélection d'un Groupe de cellules dans Fichier Excel**

Pour cela nous devons utiliser la commande **Range()** associée avec **Select**. Dans les parenthèses, nous devons indiquer les références aux cellules contiguës.



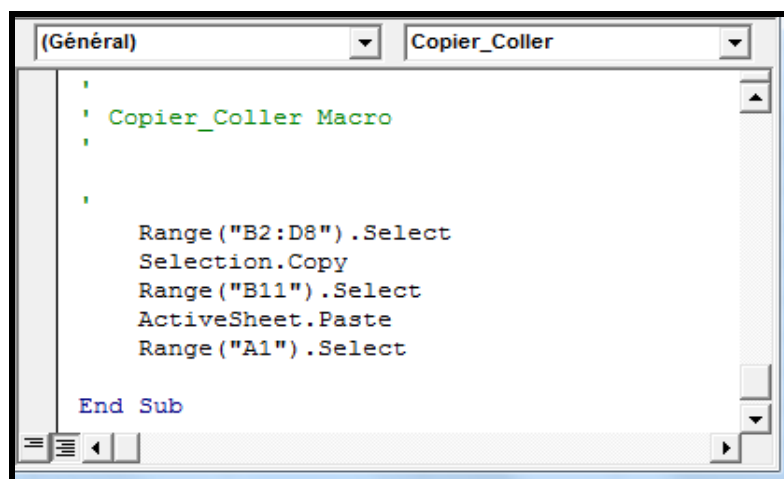
10) - Sélection de cellules non contiguës dans Fichier Excel

Pour cela nous devons utiliser la commande **Range()** associée avec **Select**. Dans les parenthèses, nous devons indiquer les références aux cellules non contiguës et éventuellement contiguës.



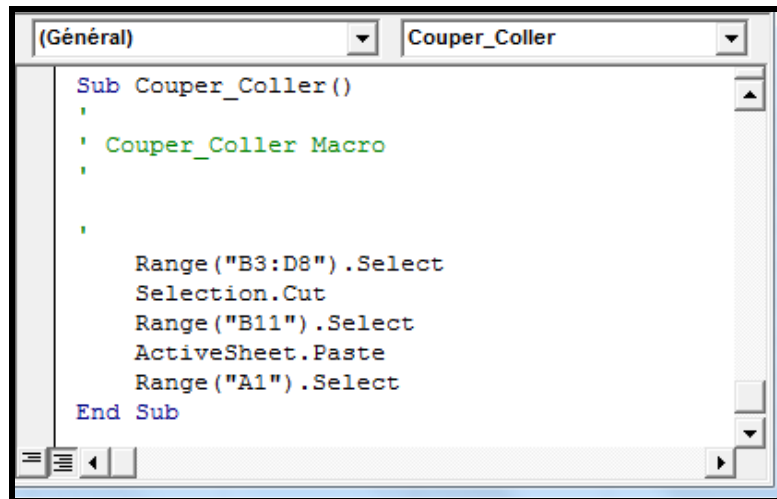
11) - Copier/Coller un groupe de cellules

Pour cela nous devons sélectionner les cellules à copier **Range()** associée avec **Select**. Puis nous utilisons la commande **Selection.Copy** pour faire la copie. Utiliser ensuite la commande **Range()** associée avec **Select**, afin d'indiquer entre les parenthèses la cellule de début de la commande Coller. Pour finir nous utilisons la commande **ActiveSheet.paste** pour coller la cellule ou la plage de cellules. Pour ôter la sélection des cellules coller, vous pouvez utiliser de nouveau la commande **Range()** associée avec **Select** afin d'indiquer entre les parenthèses la cellule qui sera active. Exemple **Range("A1").Select**.



12) - Couper/Coller un groupe de cellules

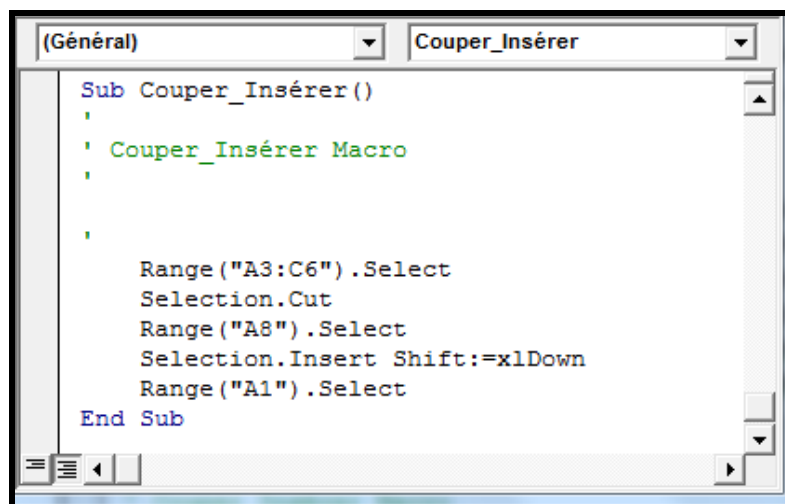
Pour cela nous devons sélectionner les cellules à copier **Range()** associée avec **Select**. Puis nous utilisons la commande **Selection.Cut** pour couper. Utiliser ensuite la commande **Range()** associée avec **Select**, afin d'indiquer entre les parenthèses la cellule de début de la commande Coller. Pour finir nous utilisons la commande **ActiveSheet.paste** pour coller la cellule ou la plage de cellules. Pour ôter la sélection des cellules coller, vous pouvez utiliser de nouveau la commande **Range()** associée avec **Select** afin d'indiquer entre les parenthèses la cellule qui sera active. Exemple **Range("A1").Select**.



```
(Général) Couper_Coller
Sub Couper_Coller()
'
' Couper_Coller Macro
'
Range("B3:D8").Select
Selection.Cut
Range("B11").Select
ActiveSheet.Paste
Range("A1").Select
End Sub
```

13) - Couper/Insérer un groupe de cellules

Pour cela nous devons sélectionner les cellules à copier **Range()** associée avec **Select**. Puis nous utilisons la commande **Selection.Cut** pour couper. Utiliser ensuite la commande **Range()** associée avec **Select**, afin d'indiquer entre les parenthèses la cellule de début de la commande Coller. Pour finir nous utilisons la commande **Selection.Insert Shift:=xlDown** pour coller la cellule ou la plage de cellules. Pour ôter la sélection des cellules coller, vous pouvez utiliser de nouveau la commande **Range()** associée avec **Select** afin d'indiquer entre les parenthèses la cellule qui sera active. Exemple **Range("A1").Select**.



```
(Général) Couper_Insérer
Sub Couper_Insérer()
'
' Couper_Insérer Macro
'
Range("A3:C6").Select
Selection.Cut
Range("A8").Select
Selection.Insert Shift:=xlDown
Range("A1").Select
End Sub
```

Insert Shift Peut avoir la valeur **xlToRight** pour insérer avec décalage vers la Droite ou **xlDown** pour insérer avec décalage vers le bas.

Voir Tableau de la page suivante.

Insert, méthodeVoir aussi [Exemple](#) [S'applique à](#)

Syntaxe 1 (objet **Range**) : Insère une cellule ou une plage de cellules dans la feuille de calcul ou la feuille macro et décale les autres cellules pour libérer la place nécessaire.

Syntaxe 2 (objet **Characters**) : Insère une chaîne précédant les caractères sélectionnés.

Syntaxe 1

expression.**Insert**(*Shift*)

Syntaxe 2

expression.**Insert**(*String*)

expression Obligatoire. Expression qui renvoie un objet **Characters** ou **Range**.

Shift Argument de type **Variant** facultatif. Spécifie la manière dont les cellules sont décalées. Il peut s'agir de l'une des constantes **XlInsertShiftDirection** suivantes : **xlShiftToRight** ou **xlShiftDown**. Si cet argument est omis, Microsoft Excel détermine lui-même le sens du décalage en fonction de la forme de la plage.

String Argument de type **String** obligatoire. Chaîne à insérer.

Insert, méthode - Exemples

Cet exemple montre comment insérer une nouvelle ligne avant la ligne 4 de Feuil1.

```
Worksheets("Feuil1").Rows(4).Insert
```

Cet exemple montre comment insérer de nouvelles cellules dans la plage A1 à C5 de Feuil1 et décale les autres cellules vers le bas.

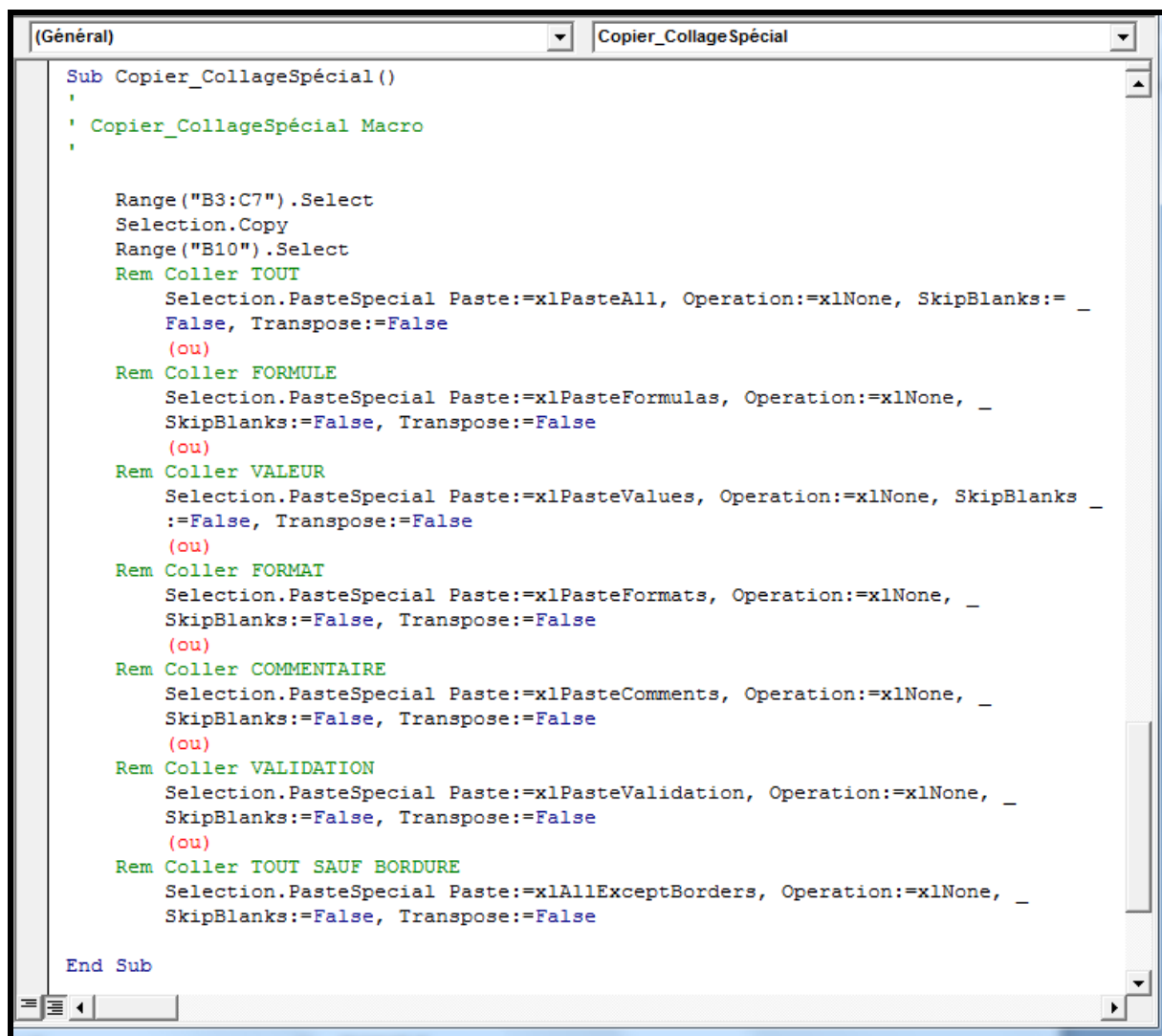
```
Worksheets("Feuil1").Range("A1:C5").Insert shift:=xlShiftDown
```

Cet exemple montre comment insérer une nouvelle ligne à l'emplacement de la cellule active. L'exemple doit être exécuté dans une feuille de calcul.

```
ActiveCell.EntireRow.Insert
```

14) - Copier/Collage Spécial d'un groupe de cellules

Pour cela nous devons sélectionner les cellules à copier **Range()** associées avec **Select**. Puis nous utilisons la commande **Selection.Copy** pour faire la copie. Utilisez ensuite la commande **Range()** associée avec **Select**, afin d'indiquer entre les parenthèses la cellule de début de la commande Coller. Pour finir nous utilisons la commande **Selection.PasteSpecial Paste:=.....** pour effectuer le collage spécial de la ou des cellules.



```
(Général) Copier_CollageSpécial
Sub Copier_CollageSpécial()
' Copier_CollageSpécial Macro
'
Range("B3:C7").Select
Selection.Copy
Range("B10").Select
Rem Coller TOUT
Selection.PasteSpecial Paste:=xlPasteAll, Operation:=xlNone, SkipBlanks:= _
False, Transpose:=False
(ou)
Rem Coller FORMULE
Selection.PasteSpecial Paste:=xlPasteFormulas, Operation:=xlNone, _
SkipBlanks:=False, Transpose:=False
(ou)
Rem Coller VALEUR
Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks _
:=False, Transpose:=False
(ou)
Rem Coller FORMAT
Selection.PasteSpecial Paste:=xlPasteFormats, Operation:=xlNone, _
SkipBlanks:=False, Transpose:=False
(ou)
Rem Coller COMMENTAIRE
Selection.PasteSpecial Paste:=xlPasteComments, Operation:=xlNone, _
SkipBlanks:=False, Transpose:=False
(ou)
Rem Coller VALIDATION
Selection.PasteSpecial Paste:=xlPasteValidation, Operation:=xlNone, _
SkipBlanks:=False, Transpose:=False
(ou)
Rem Coller TOUT SAUF BORDURE
Selection.PasteSpecial Paste:=xlAllExceptBorders, Operation:=xlNone, _
SkipBlanks:=False, Transpose:=False
End Sub
```

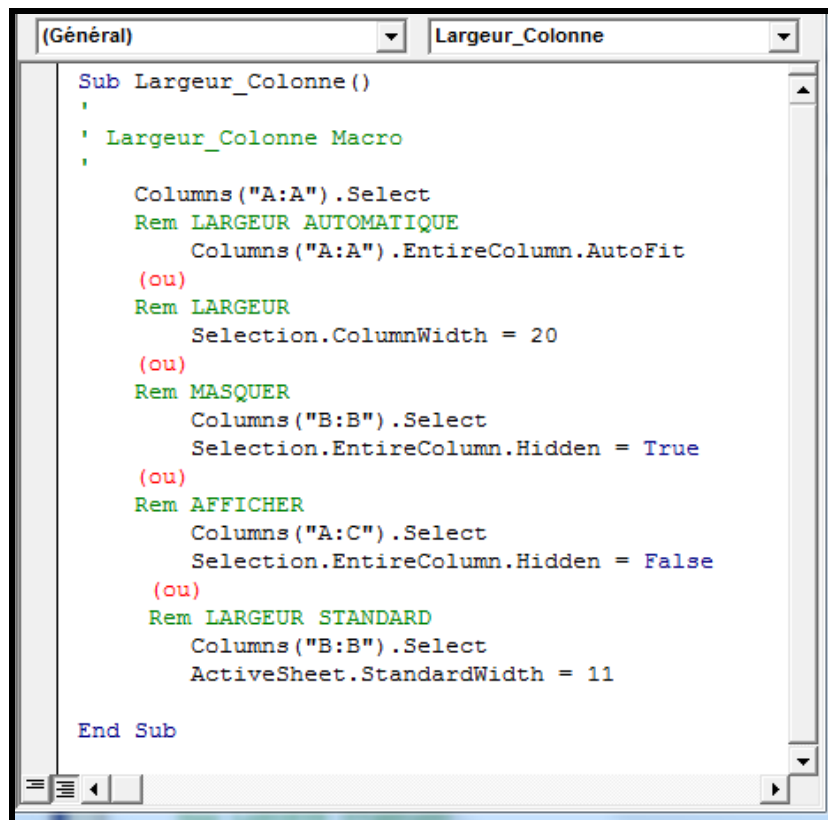
La fonction **PasteSpecial Paste**, peut avoir les valeurs suivantes:

xlAll	pour coller TOUT .
xlFormulas	pour coller seulement les FORMULES .
xlValues	pour coller seulement les VALEURS .
xlFormats	pour coller seulement le FORMAT .
xlPasteComments	pour coller seulement les COMMENTAIRES .
xlAllExceptBorders	pour coller TOUT sauf les Bordures .
xlDataValidation	pour coller seulement la VALIDATION .

Pour ôter la sélection des cellules, vous pouvez utiliser de nouveau la commande **Range()** associée avec **Select** afin d'indiquer entre les parenthèses la cellule qui sera active. Exemple **Range("A1").Select**.

15) - Largeur d'une Colonne

Pour cela nous devons sélectionner la ou les colonnes à modifier **Columns()** associée avec **EntireColumn.AutoFit**, pour une mise à largeur **Automatique**. Pour mettre une colonne à une valeur précise, nous associons **Columns()** avec **ColumnWidth=xxx**. La valeur de largeur est exprimée en **cm**.



```
Sub Largeur_Colonne ()  
,  
,  
,  
    Columns("A:A").Select  
    Rem LARGEUR AUTOMATIQUE  
    Columns("A:A").EntireColumn.AutoFit  
    (ou)  
    Rem LARGEUR  
    Selection.ColumnWidth = 20  
    (ou)  
    Rem MASQUER  
    Columns("B:B").Select  
    Selection.EntireColumn.Hidden = True  
    (ou)  
    Rem AFFICHER  
    Columns("A:C").Select  
    Selection.EntireColumn.Hidden = False  
    (ou)  
    Rem LARGEUR STANDARD  
    Columns("B:B").Select  
    ActiveSheet.StandardWidth = 11  
  
End Sub
```

La commande pour modifier les paramètres d'une colonne, peut avoir les syntaxes suivantes

Selection.Columns.autofit

Selection.ColumnWidth = xx

Selection.EntireColumn.Hidden = True

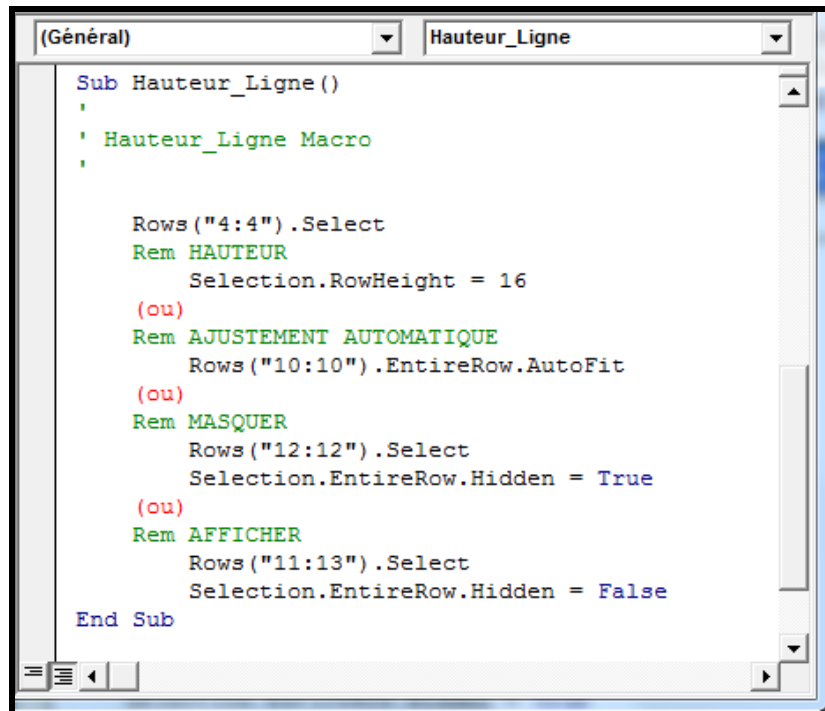
Selection.EntireColumn.Hidden = False

ActiveSheet.StandardWidth = xx

pour un **AJUSTEMENT AUTOMATIQUE**.
pour définir la **LARGEUR de la COLONNE**.
pour **MASQUER** une colonne.
pour **AFFICHER** une colonne.
pour une mise à **LARGEUR STANDARD**.

16) - Hauteur d'une Ligne

Pour cela nous devons sélectionner la ou les colonnes à modifier **Rows()** associée avec **EntireRow.AutoFit**, pour une mise à largeur **Automatique**. Pour mettre une colonne à une valeur précise, nous associons **Rows()** avec **RowHeight=xxx**. La valeur de largeur est exprimée en **cm**.



SelectionEntireRow.autofit

Selection.RowHeight = xx

Selection.EntireColumn.Hidden = True

Selection.EntireColumn.Hidden = False

pour un **AJUSTEMENT AUTOMATIQUE**.
pour définir la **LARGEUR de la COLONNE**.
pour **MASQUER** une colonne.
pour **AFFICHER** une colonne.

17) - Effacer une Cellule ou Ligne ou Colonne

Pour cela nous devons sélectionner la ou les cellules à supprimer et taper la commande

Selection.Delete Shift := et l'associée avec

XlToLeft, pour une suppression avec décalage vers la **Gauche**.

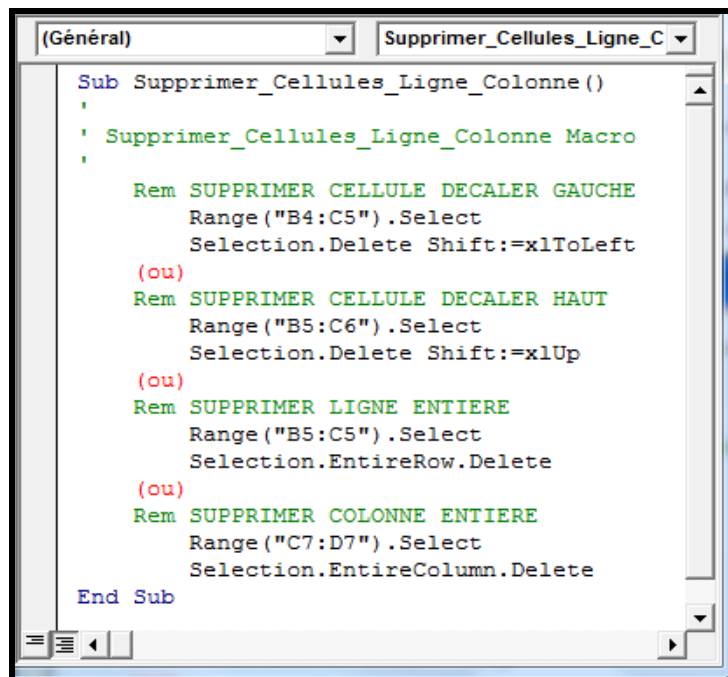
XlUp, pour une suppression avec décalage vers le **Haut**.

Pour effacer une ou plusieurs lignes, sélectionnez-les, puis taper la commande

Selection.EntireRow.Delete

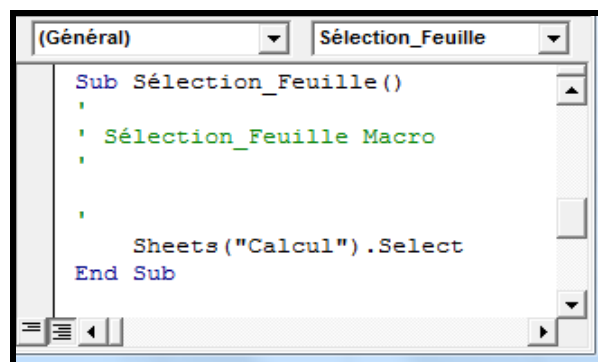
Pour effacer une ou plusieurs colonnes, sélectionnez-les, puis taper la commande

Selection.EntireColumn.Delete



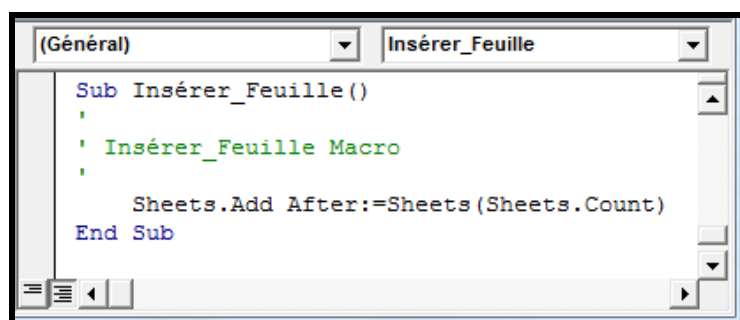
18) - Sélectionner une Feuille dans un Fichier

Pour effectuer cette sélection, vous devez utiliser la commande ***Sheets("nom_feuille").Select.***



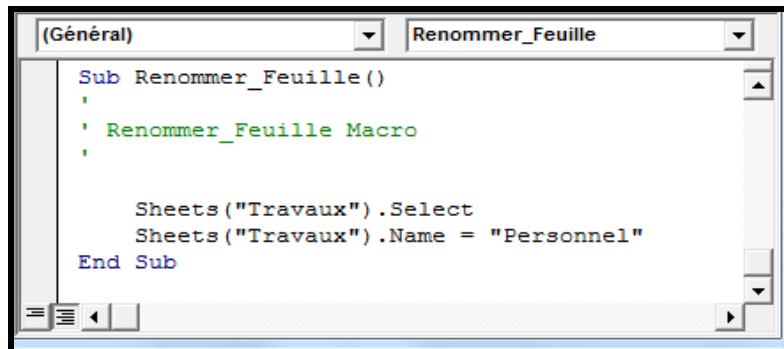
19) - Insérer une Feuille dans un Classeur

Pour insérer une nouvelle feuille dans un classeur, il faut utiliser la commande ***Sheets.Add.***



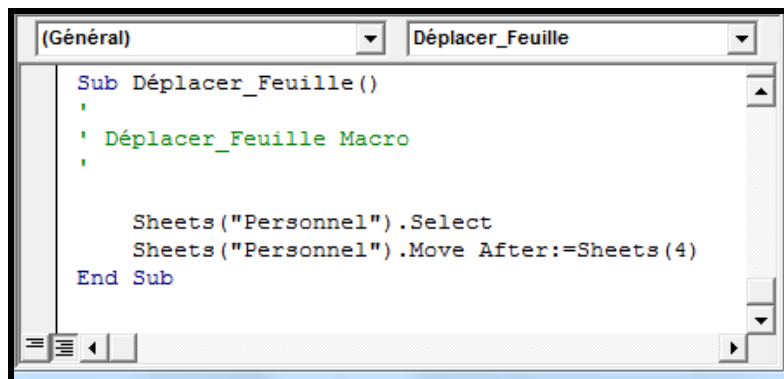
20) - Renommer une Feuille dans un Classeur

Pour renommer une feuille dans un classeur, il faut commencer par sélectionner cette feuille, puis utiliser la commande ***Sheets("Nom_feuille").Name = "Nouveau nom"***.



21) - Déplacer une Feuille dans un Classeur

Pour déplacer une feuille dans un classeur, il faut commencer par sélectionner cette feuille, puis utiliser la commande ***Sheets("Nom_feuille").Move After := Sheets("Numéro_Place")*** pour placer

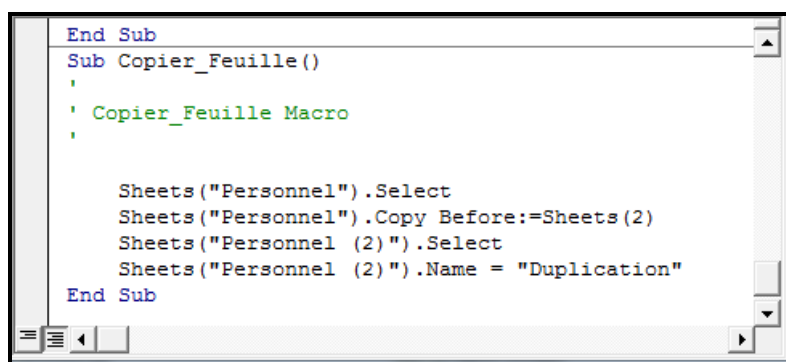


cette feuille **APRÈS** la feuille numéro (x) ou bien utiliser la commande ***Sheets("Nom_feuille").Move Before := Sheets("Numéro_Place")*** pour placer cette feuille **AVANT** la feuille numéro (x)

Dans notre exemple, Sheets(4) signifie que nous déplaçons la feuille **APRÈS** la 4^{ème} place.

22) - Copier une Feuille dans un Classeur

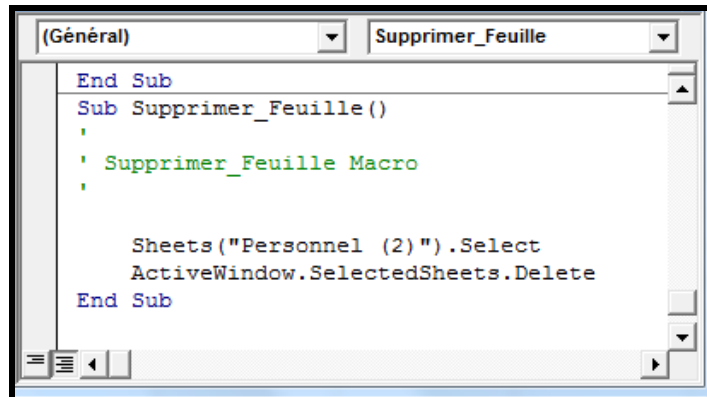
Pour effectuer la copie d'une feuille à un autre endroit du classeur, il faut commencer par sélectionner cette feuille, puis utiliser la commande ***Sheets("Nom_feuille").Copy Before:= "Nom_feuille_existante"***.



Dans l'exemple de la page précédente, nous sélectionnons la feuille "**Personnel (2)**", dont nous créons la copie avant la Feuille2. Puis nous la renommons par exemple "**Duplication**".

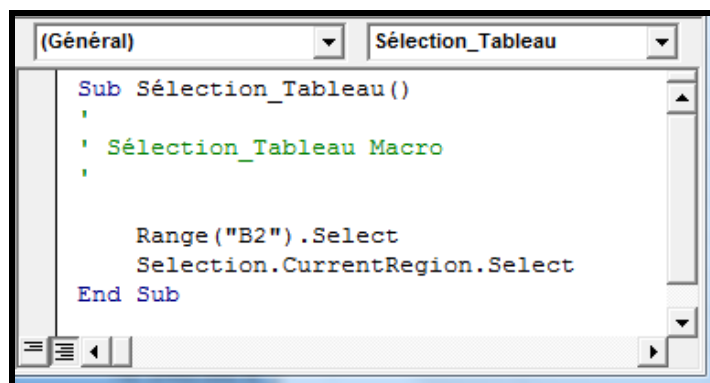
23) - Supprimer une Feuille dans un Fichier

Pour effectuer la suppression d'une feuille du classeur, il faut commencer par sélectionner cette feuille, puis utiliser la commande **ActiveWindows.SelectedSheets.Delete**.



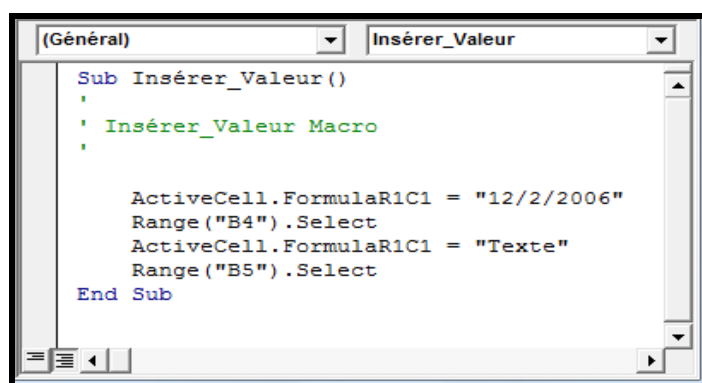
24) - Sélectionner un tableau entier dans un Fichier (Ctrl+*)

Pour effectuer la sélection d'un tableau complet dans une feuille du classeur, il faut commencer par sélectionner la 1^{ère} cellule en Haut à Gauche du tableau, puis utiliser la commande **Selection.CurrentRegion.Select**.



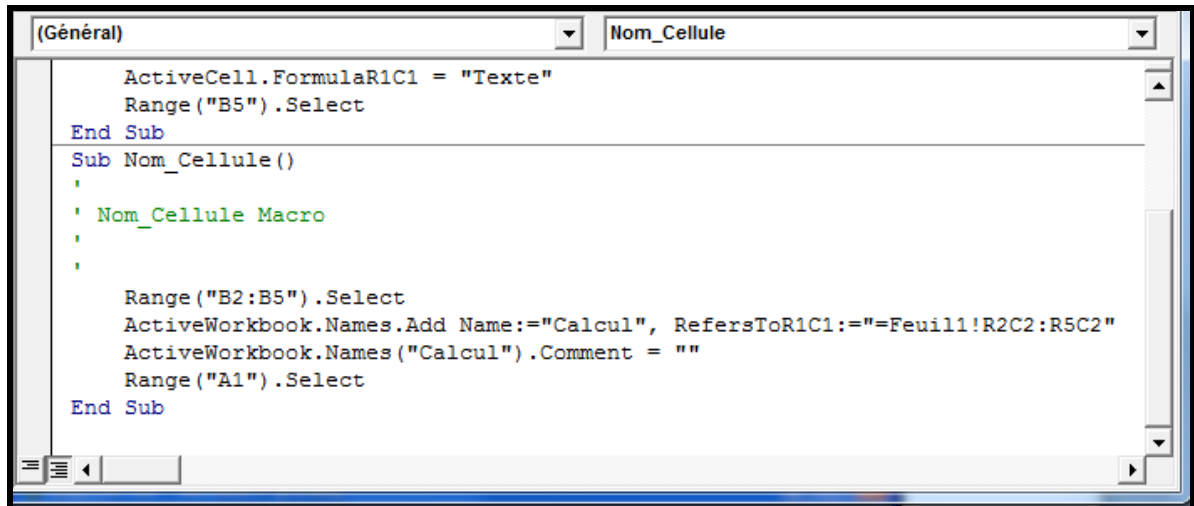
25) - Insérer une valeur précise dans une cellule

Pour effectuer l'insertion d'une valeur dans une cellule, il faut commencer par sélectionner la cellule puis utiliser la commande **ActiveCell.FormulaR1C1 = "valeur"**.



26) - Donner un NOM à une ou plusieurs cellules

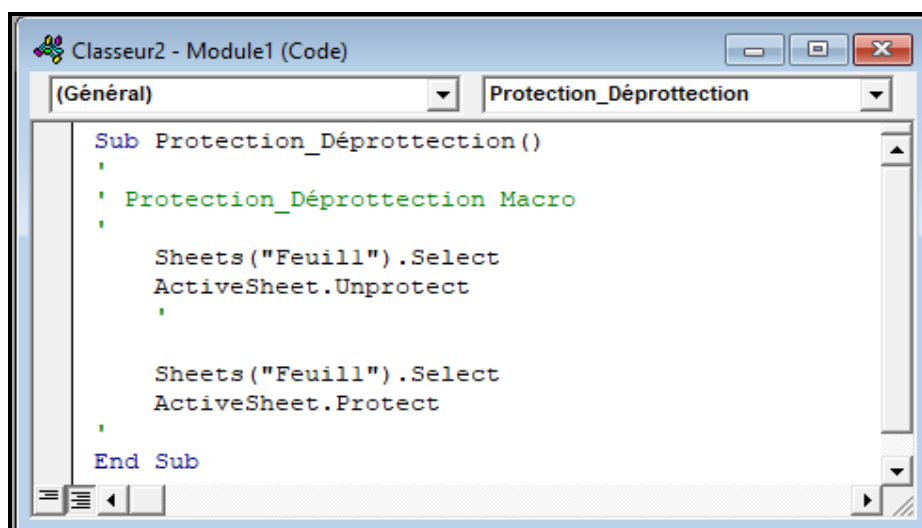
Pour donner un NOM à une ou plusieurs cellules, il faut commencer par sélectionner la cellule ou la plage des cellules concernées, puis utiliser la commande **ActiveWorkbook.Names.Add Name:= "Nom", RefersToR1C1:= "Nom_Feuille!Plage_Cellule"**. Il faut noter que la définition de la "Plage_Cellule" est donnée sous la forme RxxCxx:RxxCxx.. Exemple : si nous avons sélectionné comme plage de cellules B11:C17, la Plage_Cellule sera R11C2:R17C3.



```
(Général) Nom_Cellule
ActiveCell.FormulaR1C1 = "Texte"
Range("B5").Select
End Sub
Sub Nom_Cellule()
'
' Nom_Cellule Macro
'
Range("B2:B5").Select
ActiveWorkbook.Names.Add Name:="Calcul", RefersToR1C1:="=Feuill1!R2C2:R5C2"
ActiveWorkbook.Names("Calcul").Comment = ""
Range("A1").Select
End Sub
```

27) - Protéger et déprotéger un classeur (sans mot de passe)

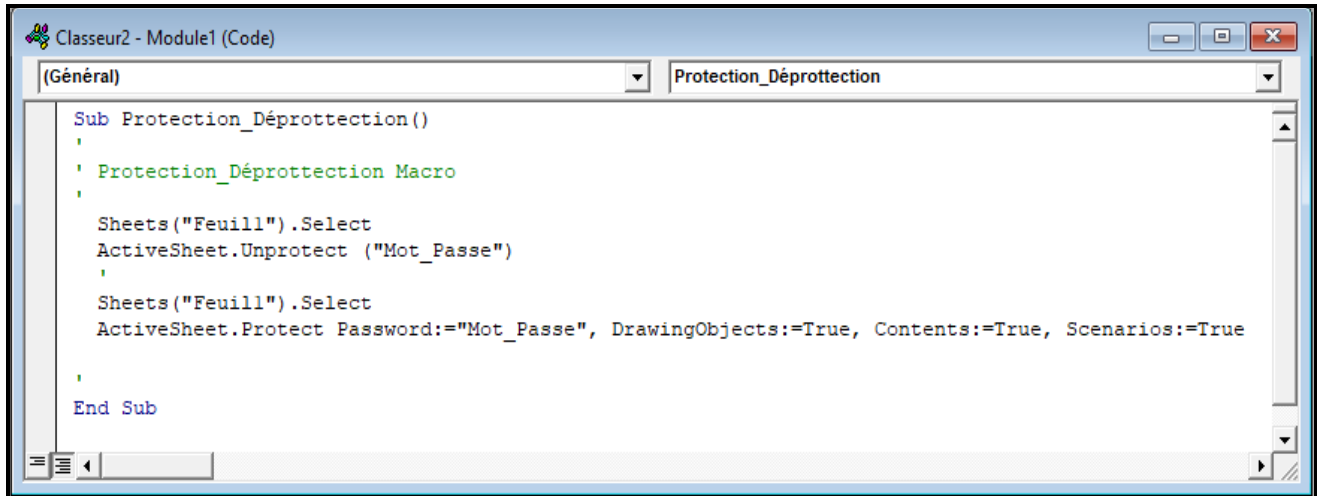
Pour protéger ou déprotéger un classeur Excel, il faut d'abord sélectionner la feuille à protéger puis utiliser la commande **ActiveSheet.Unprotect** pour la déprotéger, puis la commande **ActiveSheet.Protect** pour la protéger.



```
Classeur2 - Module1 (Code)
(Général) Protection_Déprotection
Sub Protection_Déprotection()
'
' Protection_Déprotection Macro
'
Sheets("Feuill1").Select
ActiveSheet.Unprotect
'
Sheets("Feuill1").Select
ActiveSheet.Protect
'
End Sub
```

28) - Protéger et déprotéger un classeur (AVEC mot de passe)

Pour protéger ou déprotéger un classeur Excel, il faut d'abord sélectionner la feuille à protéger puis utiliser la commande **ActiveSheet.Unprotect** pour la déprotéger, puis la commande **ActiveSheet.Protect** pour la protéger.



The screenshot shows the VBA editor window titled "Classeur2 - Module1 (Code)". The "Protection_Déprotection" macro is selected in the Project Explorer. The code in the editor is as follows:

```
Sub Protection_Déprotection()  
    ' Protection_Déprotection Macro  
    '   
    Sheets("Feuill1").Select  
    ActiveSheet.Unprotect ("Mot_Passe")  
    '   
    Sheets("Feuill1").Select  
    ActiveSheet.Protect Password:="Mot_Passe", DrawingObjects:=True, Contents:=True, Scenarios:=True  
    '   
End Sub
```