

CREER SON FORMULAIRE

 cellulexcel.blogspot.com/p/qui-na-pas-un-jour-utilise-loption.html

Page en cours de refonte pour exploitation sous Microsoft Office 2016. La programmation est exploitable pour les versions antérieures seuls les visuels et l'ergonomie est légèrement différentes.

Qui n'a pas un jour utilisé l'option formulaire dans Excel ? La plupart d'entre vous je pense. Ce formulaire a le mérite d'exister et fonctionne avec toutes les versions.

Source Microsoft : (https://support.office.com/fr-fr/article/Vue-d-ensemble-des-formulaires-contr%3%b4les-de-formulaire-et-contr%3%b4les-ActiveX-sur-une-feuille-de-calcul-dans-Excel-2016-pour-Windows-4969d331-9cea-44d3-9263-92376cad4dfc?ui=fr-FR&rs=fr-FR&ad=FR#bmdata_forms)

Excel peut générer automatiquement un formulaire de données intégré pour votre plage ou votre table.

Le formulaire de données affiche tous les en-têtes de colonnes sous forme d'étiquettes dans une boîte de dialogue unique.

Chaque étiquette comporte une zone de texte vide adjacente dans laquelle vous pouvez entrer des données pour chaque colonne, jusqu'à 32 colonnes maximum.

Dans un formulaire de données, vous pouvez entrer de nouvelles lignes, rechercher des lignes en naviguant ou modifier et supprimer des lignes en fonction du contenu des cellules.

Si une cellule contient une formule, le résultat de la formule s'affiche dans le formulaire de données, mais vous ne pouvez pas modifier la formule à l'aide du formulaire de données.

Pour ma part je ne le trouve pas agréable d'utilisation et son inconvénient c'est qu'il est limité en terme de champ soit exactement 32.

Ces champs correspondent aux colonnes **juxtaposées** que vous avez **sélectionnées**.

	A	B	C	D	E	F	G	H	I
1	NOM	Prénom	Adresse1	Adresse2	Code Postal	VILLE	N° Tel. Fixe	N° Tel. Mobile	adresse mail
2	DURAND	Patrick	2, rue de la Fontaine	Zone Artisanale xxx	44000	NANTES	01 02 03 04 05	06 07 08 09 10	durand.patrick@free.fr
3	DUPONT	Jean	4, avenue de la Forge		50000	CHERBOURG			j.dupont@sfr.fr
4									
5									
6									
7									
8									
9									
10									
11									
12									
13									
14									
15									
16									
17									
18									
19									

? X

FICHER ADRESSES

NOM: 1 sur 2

Prénom: Nouvelle

Adresse1: Supprimer

Adresse2: Restaurer

Code Postal: Prédécédente

VILLE: Suivante

N° Tel. Fixe: Critères

N° Tel. Mobile: Fermer

adresse mail:

FICHER ADRESSES ACCUEIL

Donc pas de possibilité d'avoir un formulaire avec des champs masqués et une visualisation agréable, ni de liste déroulante, ni de choix d'option et etc. comme dans l'image ci-après issue du même fichier.

The image shows a Microsoft Excel spreadsheet with the following data:

1	NOM	Prénom	Adresse1
2	DURAND	Patrick	2, rue de la Fontaine
3	DUPONT	Jean	4, avenue de la Forge
4			
5			
6			
7			
8			
9			
10			
11			
12			
13			
14			
15			
16			
17			
18			
19			

The dialog box 'GESTION DU FICHIER ADRESSES' contains the following fields and buttons:

- Rechercher par NOM: [Dropdown menu]
- NOM: [Text input]
- Prénom: [Text input]
- Adresse1: [Text input]
- Adresse2: [Text input]
- Code Postal: [Text input]
- VILLE: [Text input]
- N° Tel. Fixe: [Text input]
- N° Tel. Mobile: [Text input]
- Label1: [Text input]
- Buttons: MODIFIER, NOUVEAU, QUITTER

Créer un formulaire de ce type n'est pas compliqué et relativement simple si vous suivez scrupuleusement les explications à suivre.... Pas de limitation de champs, ergonomie personnalisée etc.

Donc au travail si vous le souhaitez.....

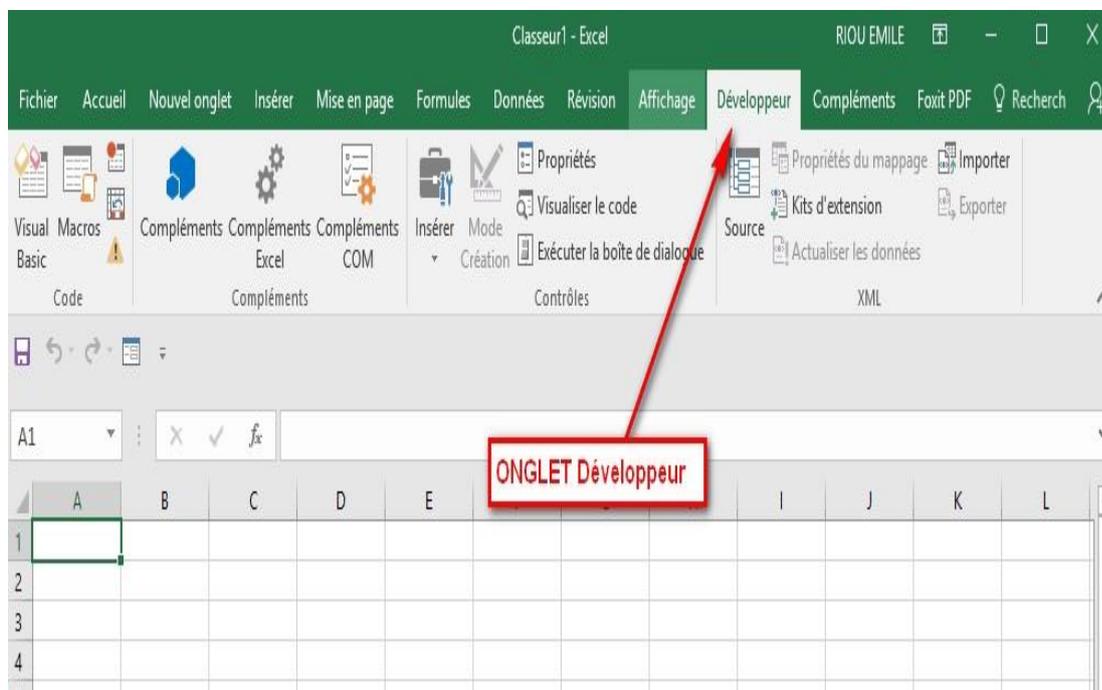
PARAMETRAGES D'EXCEL (Version 2016)

1) Version d'enregistrement du fichier Excel :

Vous devez enregistrer votre fichier sous "**Classeur Excel (prenant en charge les macros)(*.xlsm)**" Recommandé

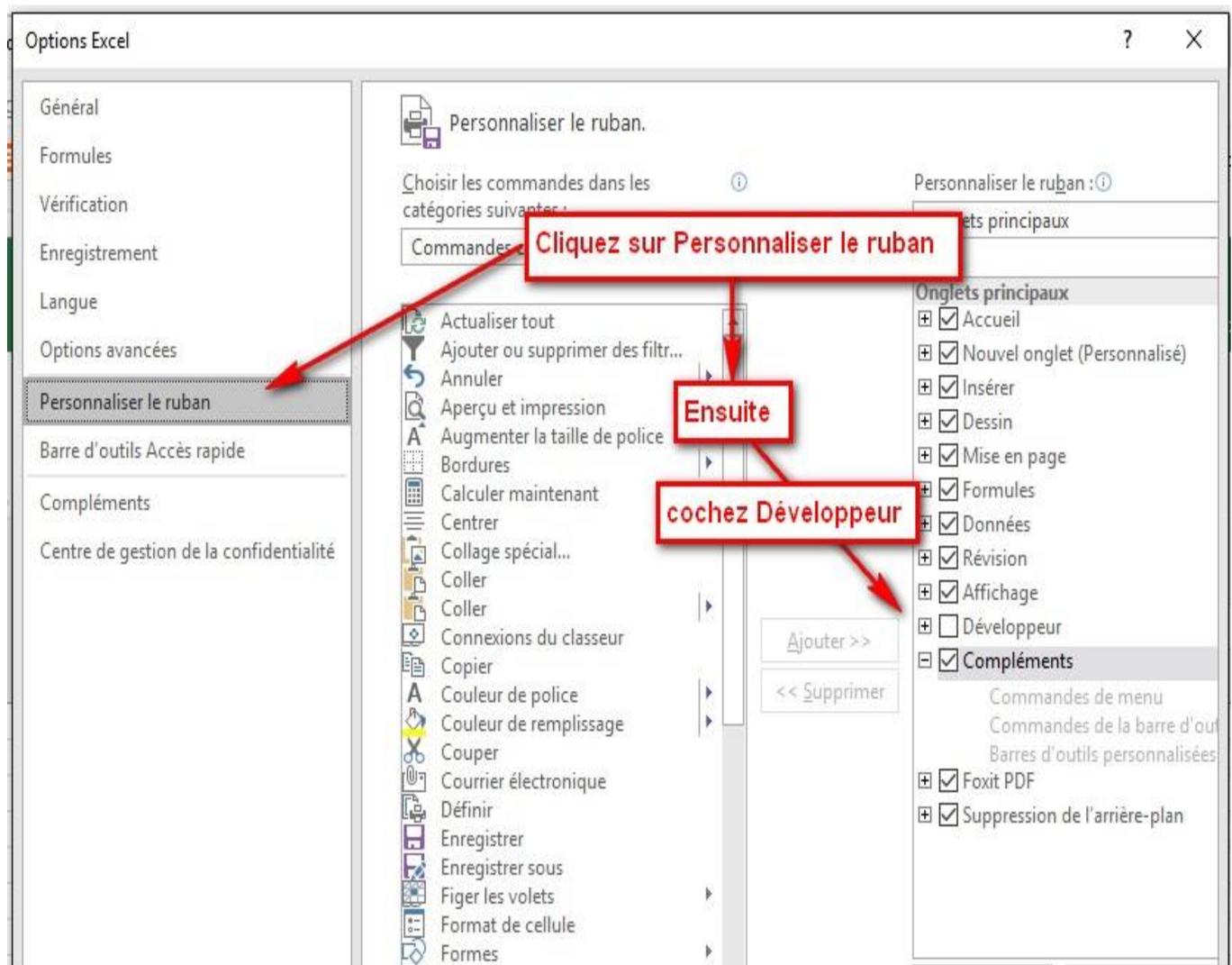


2) Vous devez avoir l'onglet "Développeur" apparent dans Excel :



Pour afficher l'onglet **Développeur** dans Excel 2016, procédez de la manière suivante :
(Pour les autres versions vous trouverez facilement la méthodologie sur la toile.)

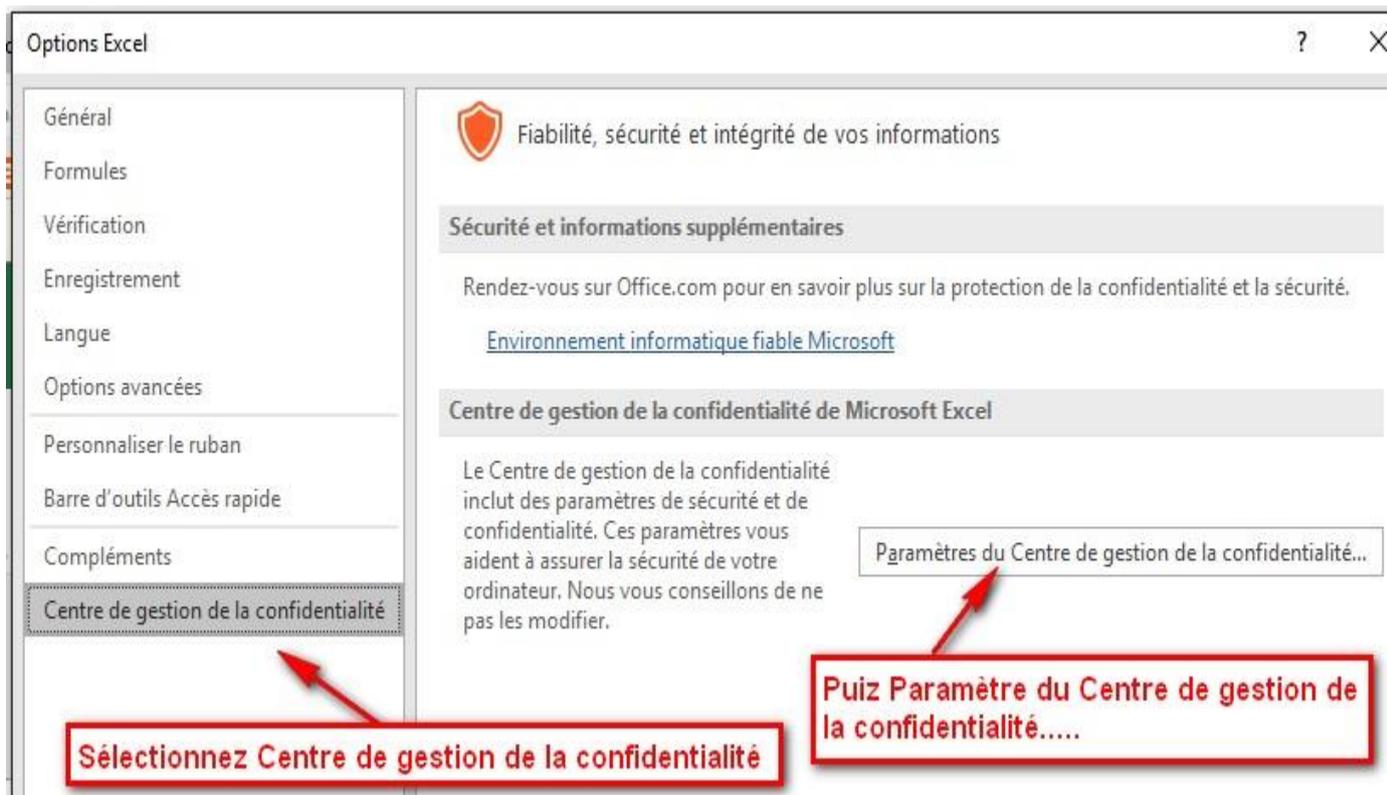
- Dans le ruban supérieur cliquez sur **Fichier** , puis sur **Option**.
- Dans la fenêtre qui s'ouvre sélectionnez **Personnaliser le ruban** et cochez **Développeur** comme sur l'image suivante.



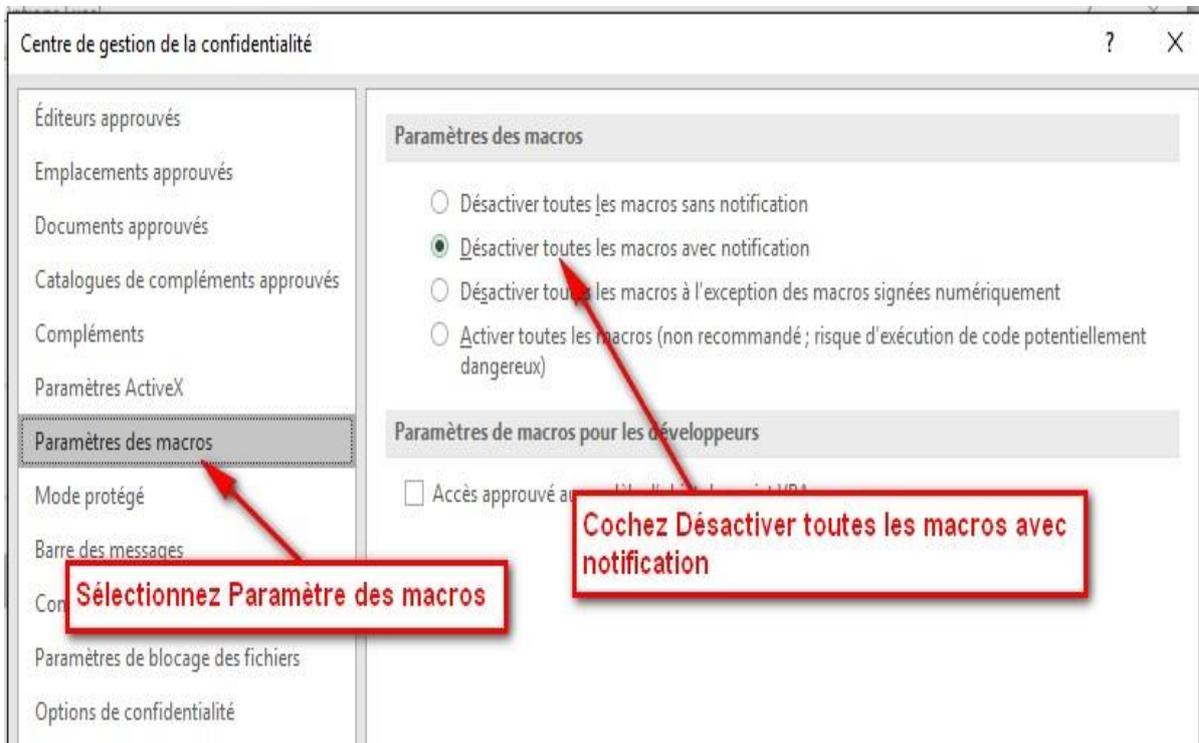
Remarque Le ruban est un élément de l'Interface utilisateur Microsoft Office Fluent

3) Les Macros doivent pouvoir être activées automatiquement :

- Dans le ruban supérieur cliquez sur **Fichier** , puis sur **Option**.
- Dans la fenêtre **Options Excel** sélectionnez **Centre de gestion de confidentialité**



- puis dans la nouvelle fenêtre Centre de gestion de la confidentialité
- Sélectionnez **Paramètres des macros**
- Puis cocher **Désactiver toutes les macros avec notification.**



CREATION DU FORMULAIRE

A ce stade votre fichier est prêt pour la création d'un formulaire mais encore faut-il entrer

quelques informations dans votre fichier.

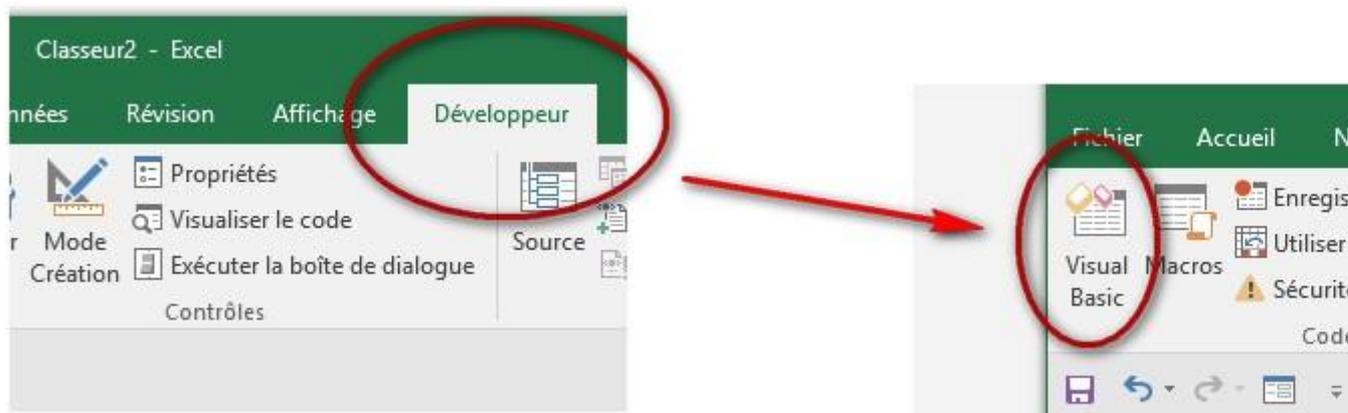
Créez les colonnes suivantes (exemple sur lequel nous allons travailler) :

- **NOM**
- **Prénom**
- **Adresse**
- **Code Postal**
- **Ville**
- **N° de Téléphone**
- **N° de Fax**
- **E-mail**

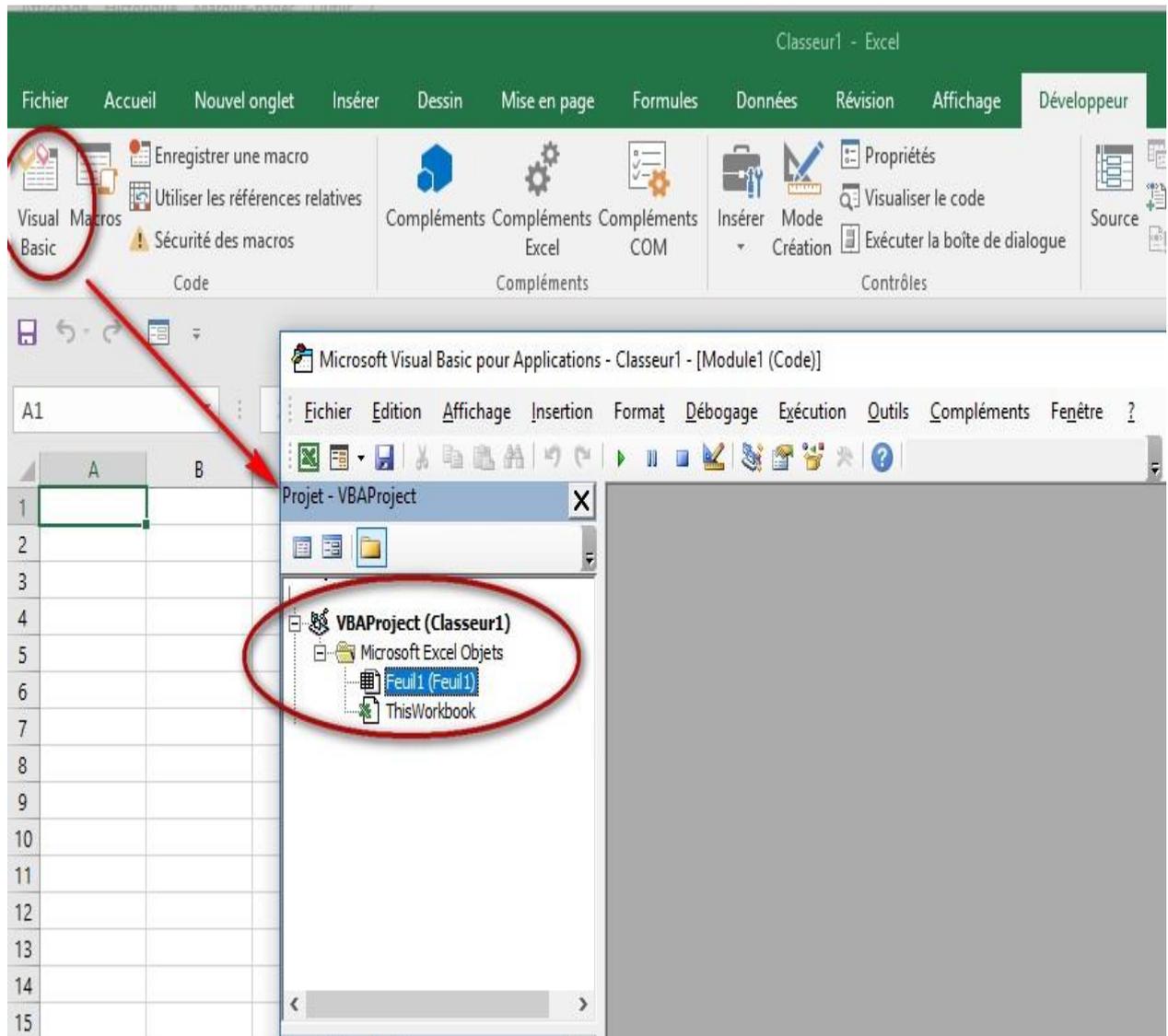
et bien plus si vous le souhaitez.... et renseignez quelques champs.

	A	B	C	D	E	F	G	H
1	NOM	Prénom	ADRESSE	Code Postal	VILLE	N° Tel.	N° Fax	E-MAIL
2	DURAND	Patrick	2, rue de la Fontaine	44000	NANTES		02 99 00 20 22	durand.patrick@free.fr
3	DUPONT	JEAN	4, avenue de la Forge	50000	Cherbourg			j.dupont@sfr.fr
4								
5								

Maintenant nous allons cliquer sur l'onglet "Développeur" pour accéder à la programmation via Visual Basic.



Nous accédons à l'environnement Microsoft Visual Basic :



Sur ce lien vous accéder aux différents

MENUS de l'Editeur VBA

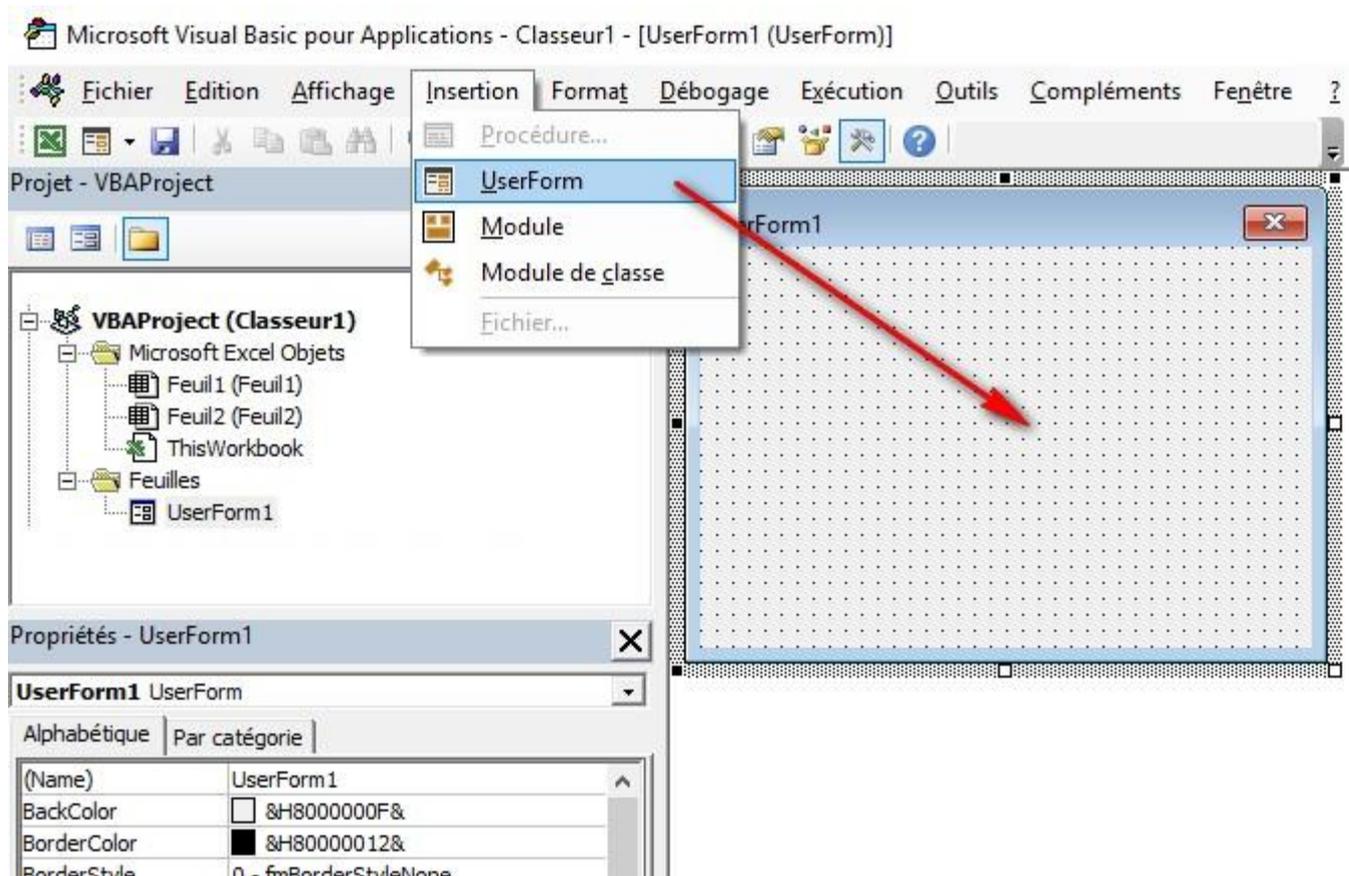
Le formulaire sera composé de :

- * **d'un USERFORM**
- * **de deux boutons de commande (CommandButton)**
- * **de 8 zones de texte (Textbox)**
- * **d'une zone liste modifiable (ComboBox)**
- * **de 8 intitulés (Label)**

1) USERFORM

Dans un premier temps nous allons insérer un UserForm :

- Choisir l'onglet **I**nsertion
- Cliquez sur **U**serform

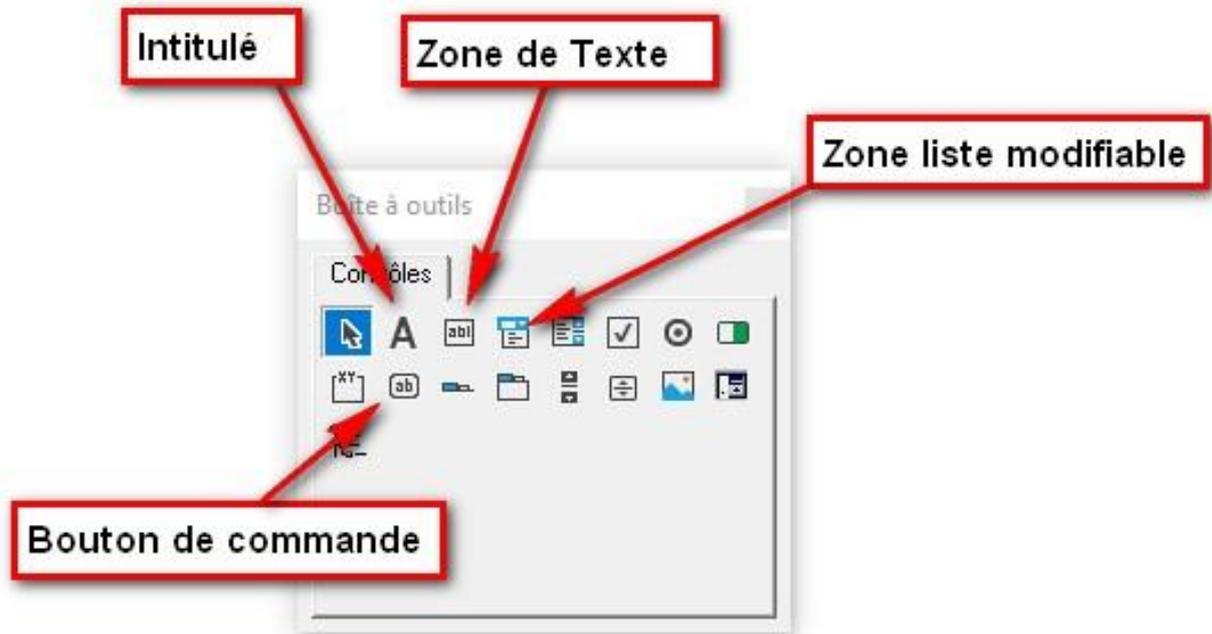


Votre premier UserForm va apparaître et il sera nommé **UserForm1**

Si la boîte à outils n'apparaît pas choisir l'onglet **Affichage** et cliquez sur **Boîte à outils**

Cette boîte à outils va nous permettre d'ajouter tous les éléments nécessaire au fonctionnement de notre formulaire à savoir :

- Une liste déroulante de choix ou de recherche
- Les champs d'informations
- Les titres des champs d'informations
- Les boutons "Modifier" et "Quitter"



2) BOUTON DE COMMANDE

Définition d'un BOUTON DE COMMANDE :

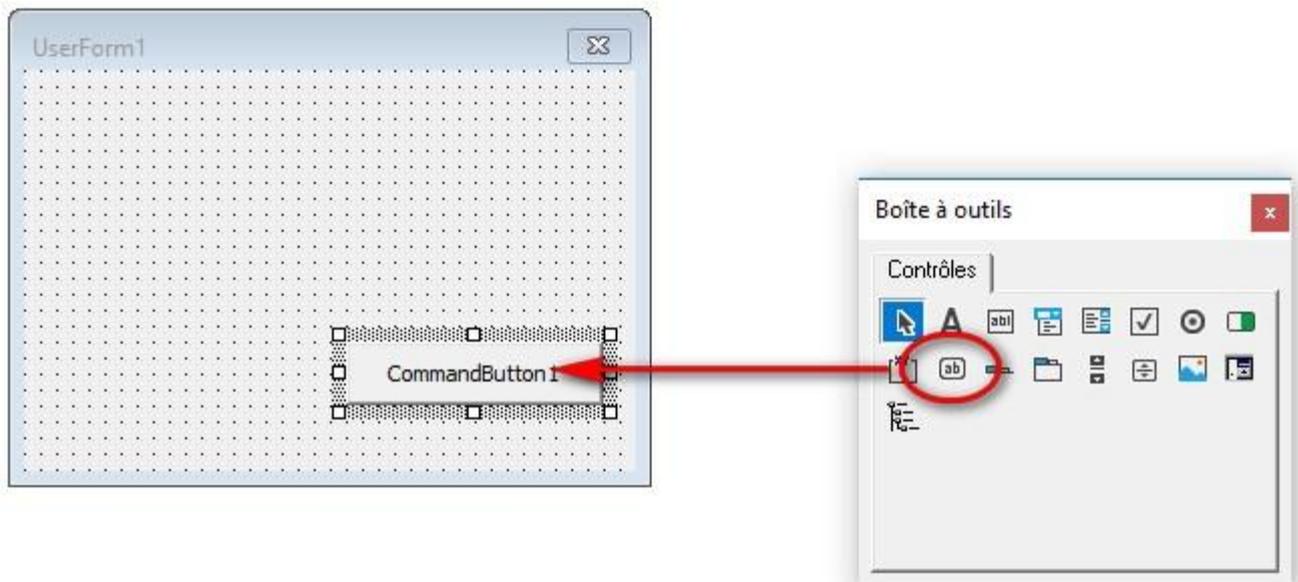
*En langage VBA appelé **CommandButton** est utilisé pour lancer les traitements d'un programme.*

C'est le plus important des contrôles.

Il ne transmet pas de données mais son événement "click" déclenche des actions.

La caption apparaît sur le bouton et doit annoncer sommairement ce qu'il fait.

Procédons à l'insertion des boutons "MODIFIER" et "QUITTER", pour cela cliquez sur le **bouton de commande** une première fois.



Votre bouton se positionnera dans l'UserForm sous le nom de **CommandButton1** auquel nous allons lui attribuer un nom.

Dans la fenêtre Propriétés suivi du nom de l'objet sélectionné (dans le cas présent Propriétés - CommandButton1) remplacer le nom **CommandButton1** en face de **Caption** par **QUITTER**

Projet - VBAProject

- VBAProject (Classeur1)
 - Microsoft Excel Objets
 - Feuil1 (Feuil1)
 - Feuil2 (Feuil2)
 - ThisWorkbook
 - Feuilles
 - UserForm1

Propriétés - CommandButton1

CommandButton1 CommandButton

Alphabétique | Par catégorie

(Name)	CommandButton1
Accelerator	
AutoSize	False
BackColor	<input type="checkbox"/> &H8000000F&
BackStyle	1 - fmBackStyleOpaque
Cancel	False
Caption	QUITTER
ControlTipText	
Default	False
Enabled	True

Sélectionnez le bouton "CommandButton1" et dans "Propriétés" à la ligne "Caption" changez le nom CommandButton par "QUITTER"

Procédez de la même manière pour le deuxième bouton de commande (**CommandButton2 = MODIFIER**)

A ce stade vous pouvez déjà visualiser votre formulaire avec ces deux boutons en appuyant sur la touche **F5** pour fermer le formulaire cliquer sur la croix en haut à droite.

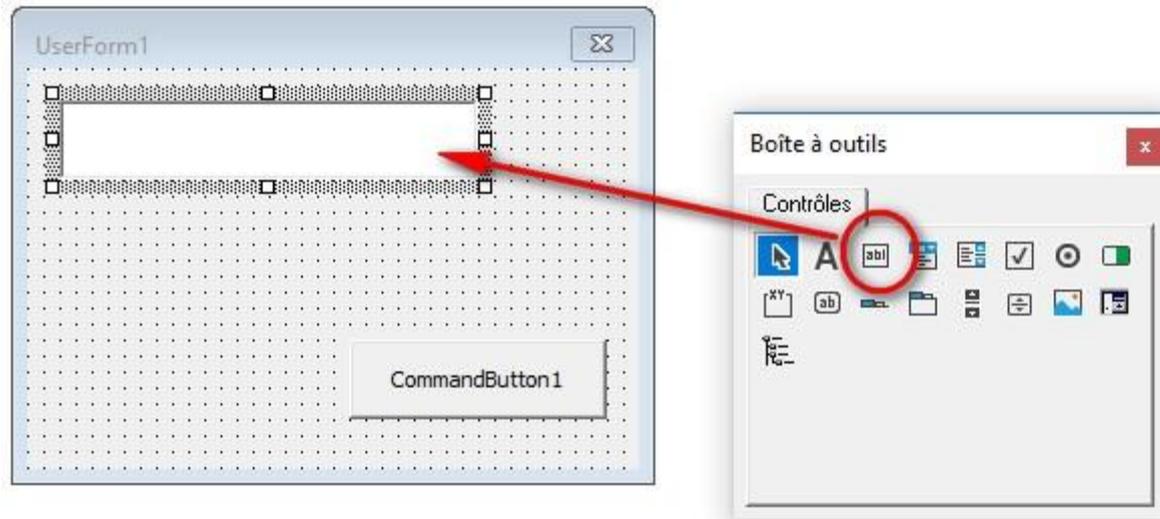
3) ZONE DE TEXTE ou ZONE D'ENTREE TEXTE

Définition d'une ZONE DE TEXTE :

*En langage VBA appelé **Textbox** est utilisé pour afficher des informations saisie par l'utilisateur qui peut être récupérée par sa propriété *Text* ou *Value*.*

Le contrôle doit être accompagné d'un label qui donne un minimum de description.

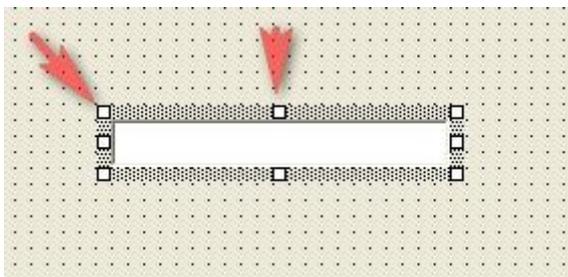
Nous allons maintenant insérer les zones de texte.



Insérer autant de ZONE de TEXTE que de nombre de colonne que vous avez dans votre fichier Excel.

Dans notre fichier d'exemple l'UserForm comporte 8 zones de texte.

Disposez les un par un dans l'UserForm en modifiant à votre gré leur format en les sélectionnant puis en utilisant les poignées pour les étirer, raccourcir ou autre.



Ne faites rien d'autres et surtout pas changer de nom. Ils doivent rester intitulé TextBox1, TextBox2 et etc.

4) ZONE LISTE MODIFIABLE

Définition d'une ZONE DE LISTE MODIFIABLE :

*En langage VBA appelé **ComboBox** est un contrôle qui réunit les caractéristique de la zone de liste et de la zone de texte. L'utilisateur peut choisir un élément sans la liste ou saisir une valeurs dans la zone de texte.*

Procédez de la même manière que précédemment pour insérer cette Zone Liste Modifiable, et positionnez la où bon vous semble.

Elle s'appelle par défaut **ComboBox1**. Ne changer pas son nom dans la fenêtre Propriétés.

5) INTITULE

Définition d'INTITULE:

*En langage VBA appelé **Label** est un contrôle qui permet d'afficher un texte non modifiable, par l'utilisateur, par exemple des titres ou des légendes, etc.*

Il peut servir à communiquer un résultat mais le plus souvent à informer sur un autre contrôle.

Ajouter autant d'INTITULE que vous souhaitez dans l>UserForm pour repérer vos zone de texte.

Le premier INTITULE s'appelle Label1 et ainsi de suite. Vous pouvez changer le nom dans l>UserForm en procédant de la manière suivante :

Dans la fenêtre propriétés remplacer le nom **Label1** en face de **Caption** par "**NOM**" et ainsi de suite selon vos souhaits.

Et voilà vous avez normalement un résultat visible en pressant sur la touche **F5**

Mon exemple

GESTION DU FICHIER ADRESSES X

Rechercher par NOM

NOM : Prénom :

Adresse1 :

Adresse2 :

Code Postal : VILLE :

N° Tel. Fixe : N° Tel. Mobile :

Label1

Cliquez sur **la croix en haut à droite** pour fermer le formulaire car le bouton QUITTER n'est pas associé à un programme pour l'instant.

PROGRAMMATION de l'USERFORM

Vous avez sans doute remarqué, dès lors que vous aviez pressé la touche **F5**, le formulaire se présentait à l'écran et par curiosité vous avez sûrement cliqué sur la touche **QUITTER** ou **MODIFIER**, et essayer d'activer la liste déroulante.

SANS RÉSULTAT

Et pour cause, le formulaire et tous les éléments qui y sont intégrés ne sont pas associés à une programmation.

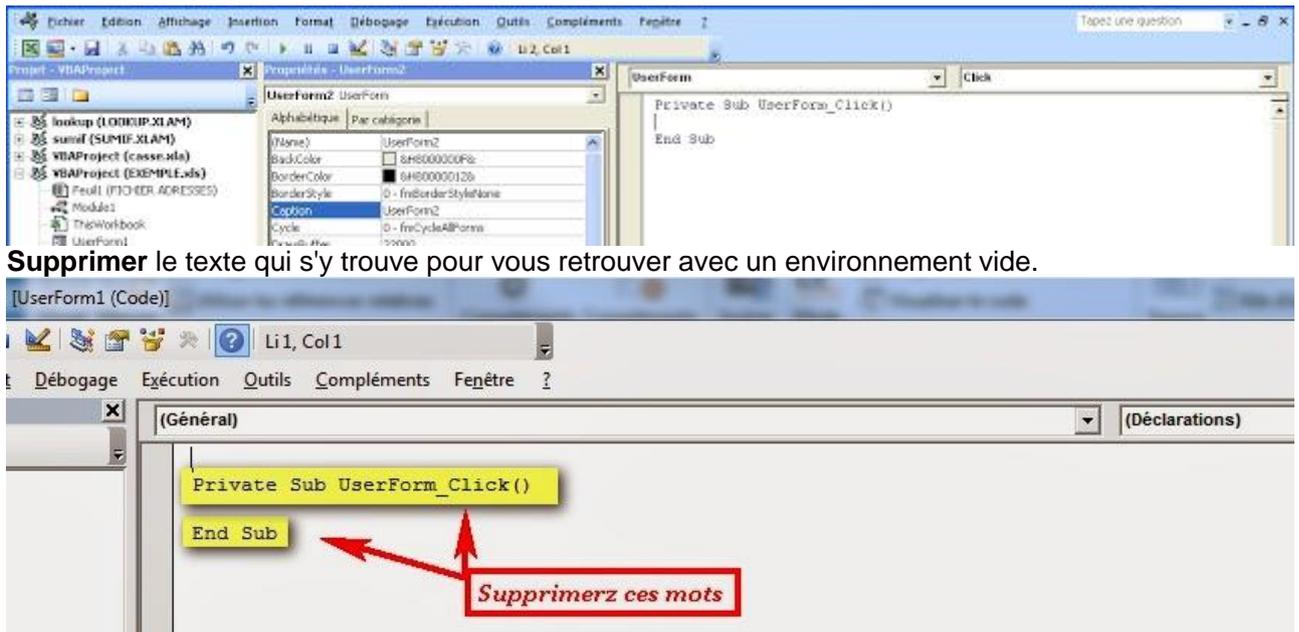
Ce que nous allons effectuer maintenant. Vous n'avez pas besoin de connaissance particulière en programmation simplement de suivre à la lettre les explications ci-après.

Pour les plus curieux n'hésitez pas tester voire même consulter d'autres sites, mais je développerai plus tard quelques trucs et astuces.

Donc dans un premier temps retourner dans l'environnement Microsoft Visual Basic.

Cliquez n'importe où sur votre UserForm.

Votre UserForm va disparaître et laisser place à un nouvel environnement :



Supprimer le texte qui s'y trouve pour vous retrouver avec un environnement vide.

[UserForm1 (Code)]

Li1, Col1

Débugage Exécution Outils Compléments Fenêtre ?



Sélectionnez le texte ci-après et collez-le dans cet environnement.

' Les lignes qui débutent par une apostrophe identifient un commentaire c'est-à-dire du texte exploitable par le lecteur et non par le programme VBA

' Objectif : documenter le code VBA pour le rendre plus compréhensible.

'Attention les deux lignes suivantes doivent être impérativement placées en tout début de programme

Option Explicit 'Activation de la déclaration explicite des variables

Dim Ws As Worksheet 'Variable pour un Objet Worksheet en PUBLIC pour tous les Controls de cet UserForm

'Correspond au programme du bouton QUITTER

Private Sub CommandButton1_Click()

Unload Me

'Unload => Permet de fermer l'Userform et de l'effacer de la mémoire, les valeurs de ses contrôles sont à alors perdues.

' Me => Référence à l'objet en cours

End Sub

'Correspond au programme du FORULAIRE

'Cet évènement est important car il permet de définir les propriétés des objets et les valeurs par défaut des variables, lors du lancement du Userform.

Private Sub UserForm_Initialize() 'à l'initialisation de l'userform

' Déclaration des variables grâce à Dim

' 1° Choisir un nom

' 2° Choisir un TYPE (= définir la nature du contenu)

'-----

Dim J As Long 'déclare la variable J

Dim I As Integer

' Affecter une valeur à une variable (= fixer la valeur de la variable)

Set Ws = Sheets("FICHER ADRESSES") 'Attention ce nom doit correspondre au nom de votre ONGLET

'On stock le nom de la feuille "FICHER ADRESSES" dans une variable nommée WS

```

With Me.ComboBox1
For J = 2 To Ws.Range("A" & Rows.Count).End(xlUp).Row
.AddItem Ws.Range("A" & J)
Next J
End With

```

```

For I = 1 To 8 'on lance une boucle de valeur mini 1 et maxi 8
Me.Controls("TextBox" & I).Visible = True 'affiche les données dans les textbox
Next I
End Sub

```

```

'Correspond au programme du bouton MODIFIER
Private Sub CommandButton2_Click()
If MsgBox("Etes-vous certain de vouloir modifier ce produit ?", vbYesNo, "Demande de confirmation") = vbYes Then
Dim Ligne As Long
Dim I As Integer

```

```

If Me.ComboBox1.ListIndex = -1 Then Exit Sub 'On sort si pas de sélection
Ligne = Me.ComboBox1.ListIndex + 2
For I = 1 To 8
If Me.Controls("TextBox" & I).Visible = True Then
Ws.Cells(Ligne, I + 1) = Me.Controls("TextBox" & I)
End If
Next I
End If 'fin de la condition

```

```

'Code permettant de modifier le format de la plage de cellule en format nombre
With Ws.Range("D2:d10")
.NumberFormat = "0"
.Value = .Value
End With
End Sub

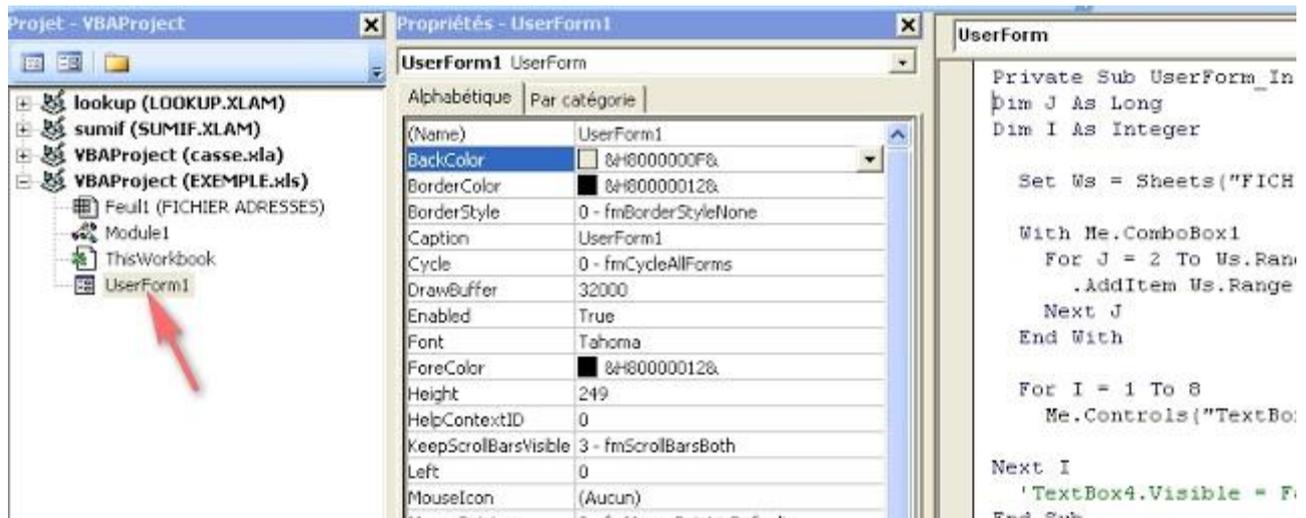
```

```

'Correspond au programme de la LISTE DEROULANTE
Private Sub ComboBox1_Change()
Dim Ligne As Long
Dim I As Integer
If Me.ComboBox1.ListIndex = -1 Then Exit Sub
Ligne = Me.ComboBox1.ListIndex + 2
For I = 1 To 8
Me.Controls("TextBox" & I) = Ws.Cells(Ligne, I + 1)
Next I
End Sub

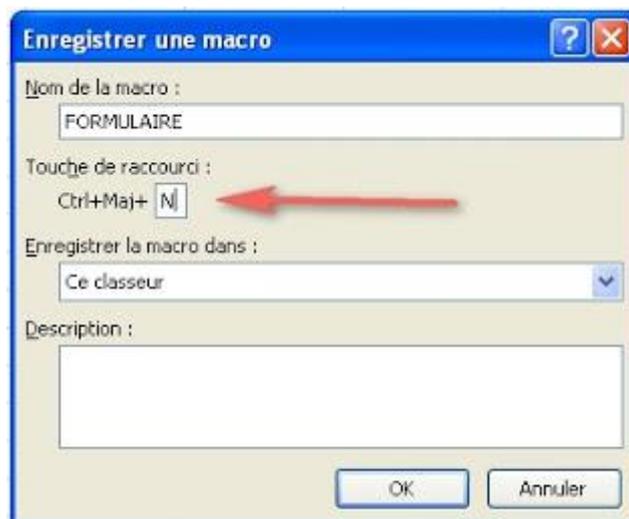
```

Maintenant que vous avez insérer cette programmation cliquez sur l'UserForm dans le projet pour faire apparaître votre UserForm.



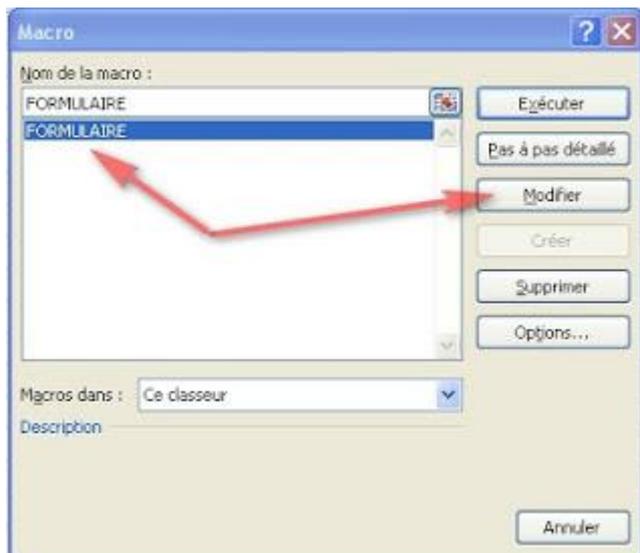
Dès que votre UserForm apparaît appuyer sur **F5** et..... **Et YOUPI !!!** Tout fonctionne par l'intermédiaire de l'environnement Vba Project.

Donc si vous souhaitez faire apparaître le formulaire depuis le fichier Excel il vous faut créer une macro avec une touche de raccourci :



Dans cet exemple j'ai donné à macro le nom de "FORMULAIRE"

Donnez à cette macro le nom que vous souhaitez.
Ensuite allez modifier cette macro



et insérer le texte suivant :

```
Sub FORMULAIRE() 'ou le nom que vous avez défini pour la macro
UserForm1.Show vbModeless '(*)
End Sub
```

(*) On distingue deux types d>Userform :

1) En mode MODAL => **Userform1.show**

Par défaut un Userform est en Modal, c'est-à-dire que l'utilisateur doit d'abord fermer l'interface pour pouvoir travailler sur une feuille de calcul.

2) En mode Non MODAL => **Userform1.show vbmodeless**

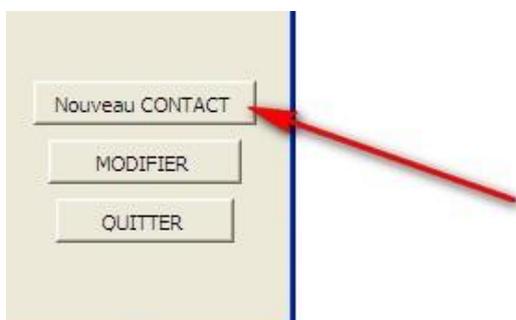
L'utilisateur peut intervenir sur une feuille de calcul en dehors de l'interface et sans la fermer.

Et voilà c'est fini.... Avec ce formulaire vous pouvez remonter toutes informations insérer dans votre fichier Excel et modifier celles-ci.

SUITE => Insérer une nouvelle ligne via le formulaire

Pour insérer une nouvelle ligne via le formulaire voici la méthodologie

Tout d'abord vous ajouter un 3ème bouton de commande (CommandButton3) dans votre Userform que vous allez nommer "Nouveau CONTACT" :



Ensuite vous double-cliquez sur ce bouton pour lui attribuer un code de procédure pour l'insertion d'un nouveau contact. A l'écran s'affichera le texte suivant :

```
Private Sub CommandButton3_Click()
```

```
End Sub
```

A ce niveau vous allez insérer le code suivant entre les deux lignes affichées

Dim L As Integer

```
If MsgBox("Etes-vous certain de vouloir INSERER ce nouveau contact ?", vbYesNo,  
"Demande de confirmation") = vbYes Then 'condition : si oui au message  
L = Sheets("FICHER ADRESSES").Range("a65536").End(xlUp).Row + 1 'Permet de se  
positionner sur la dernière ligne de tableau NON VIDE
```

```
Range("A" & L).Value = ComboBox1 'Insère la donnée de la liste déroulante dans la colonne A  
Range("B" & L).Value = TextBox1 'Insère la donnée de la textbox1 dans la colonne B  
'et à suivre....  
Range("C" & L).Value = TextBox2  
Range("D" & L).Value = TextBox3  
Range("E" & L).Value = TextBox4  
Range("F" & L).Value = TextBox5  
Range("G" & L).Value = TextBox6  
Range("H" & L).Value = TextBox7  
Range("I" & L).Value = TextBox8  
End If
```

```
'Code permettant de modifier le format de la plage de cellule en format nombre  
With Ws.Range("D2:d10")  
.NumberFormat = "0"  
.Value = .Value  
End With
```

' Affiche une boîte de message

```
MsgBox ("Produit inséré dans fichier sélectionné") 'Vous informant que le présent contact est  
insérer dans votre tableau Excel.
```

```
Unload Me ' Vide et ferme l'Userform ( formulaire)  
UserForm1.Show 'Affiche le formulaire
```

Enregistrez votre fichier Excel et lancer votre formulaire.

A son ouverture, renseignez celui-ci...

Formulaire RECHERCHE ou MODIFICATION

Rechercher NOM Prénom

ADRESSE

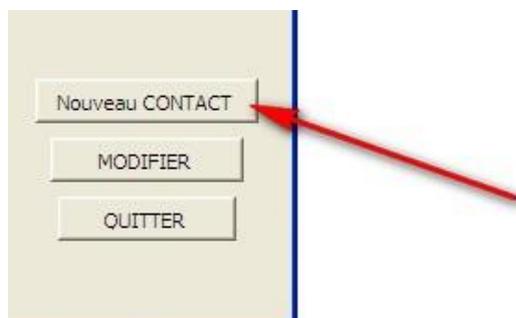
Code postal Ville

N° Téléphone fixe N° de fax

Adresse mail

Observations

et cliquez sur le bouton "Nouveau CONTACT"



Un message vous demandera de confirmer l'insertion de ce nouveau contact...



Après confirmation voici le résultat.

	A	B	C	D	E	F	G	H	I	J
1	NOM	Prénom	ADRESSE	Code postal	VILLE	N° TEL	N° FAX	ADRESSE MAIL		
2	DURAND	Patrick	2, rue de la Fontaine	44000	NANTES		02 99 00 02 00	p.durand@truc.fr		
3	DUPONT	JEAN	4, avenue de la Forge	50000	Cherbourg					
4	ZORRO	Diégo	3, rue de la Cape Noir	83000	TOULON	02 00 00 00 00		d.Zorro@machin.net	contact difficile	
5										
6										
7										
8										
9										
10										
11										
12										
13										
14										
15										
16										
17										
18										
19										
20										
21										
22										
23										

SUITE => Supprimer une ligne erronée ou une ligne obsolète via le formulaire

Pour supprimer une ligne via le formulaire voici la méthodologie :

Tout d'abord vous ajouter un 4ème bouton de commande (CommandButton4) dans votre Userform que vous allez nommer "SUPPRIMER".

Ensuite vous double-cliquez sur ce bouton pour lui attribuer un code de procédure pour l'insertion d'un nouveau contact. A l'écran s'affichera le texte suivant :

Private Sub CommandButton4_Click()

End Sub

A ce niveau vous allez insérer le code suivant entre les deux lignes affichées

'Supprimer la ligne sélectionnée

Dim L As Long

If Me.ComboBox1.ListIndex = -1 Then Exit Sub

L = Me.ComboBox1.ListIndex + 2

Ws.Rows(L).Delete

Enregistrez votre fichier Excel et lancer votre formulaire. et à vous de jouer.....

SUITE =>FORMULAIRE MULTIPAGES

Si votre formulaire comporte un nombre très important de Textbox, vous pouvez opter pour la version formulaire Multipage.

Formulaire sera réduit et comportera des onglets comme ceci.

The image shows a Windows-style dialog box titled "contact". It features a tabbed interface with three tabs: "ADRESSES", "CONTACTS", and "OBSERVATIONS". The "ADRESSES" tab is currently selected and highlighted in yellow. Within this tab, there is a search section with a dropdown menu labeled "Rechercher NOM" containing the text "DURAND", and a text input field labeled "Prénom" containing "Patrick". Below this is a large text area labeled "ADRESSE" containing "2, rue de la Fontaine". Underneath the address area are two smaller text input fields: "Code Postal" with "44000" and "Ville" with "NANTES". At the bottom of the dialog, there are two buttons: "MODIFIER" on the left and "QUITTER" on the right. The dialog has a standard Windows window border with a close button (X) in the top right corner.

contact

Rechercher NOM: DURAND Prénom: Patrick

ADRESSES | CONTACTS | OBSERVATIONS

N° Téléphone fixe: XXXXXX N° Fax: 02 99 00 02 00

Adresse mail: p.durand@truc.fr

MODIFIER QUITTER

contact

Rechercher NOM: DURAND Prénom: Patrick

ADRESSES | CONTACTS | OBSERVATIONS

BLA BLA

MODIFIER QUITTER

Pour cela cliquez sur le bouton Multipage de votre boîte à outils.



Ensuite positionnez vos Textbox comme bon vous semble dans les onglets.

Pour ajouter un nouvel onglet, clic droit sur le multipage et choisissez "**Nouvelle Page**" et clic droit vous pouvez la **Renommer**.

La programmation du formulaire ne change pas.

SUITE =>Envoyer un mail à votre contact via le formulaire.

Cette programmation fonctionne aussi bien avec Outlook Express ou Lotus Notes, pour les autres applications je ne les ai pas exploiter. Si par le plus grand des hasards c'était le cas...pour vous, merci de me faire un retour afin que je puisse en faire profiter tout le monde.

Donc sur ce même formulaire, ajouter un 5ème bouton de commande (CommandButton5) que vous allez nommer par exemple "E-MAIL".

Ensuite vous double-cliquez sur ce bouton pour lui attribuer un code de procédure pour l'envoi d'un mail à l'interlocuteur choisi dans le formulaire :

Private Sub CommandButton5_Click()

End Sub

A ce niveau vous allez insérer le code suivant entre les deux lignes affichées

```
Dim MailAd As String  
Dim Msg As String  
Dim Subj As String  
Dim URLto As String
```

'Sélectionne la cellule correspondant à l'adresse mail de la ligne :
If ComboBox1.ListIndex <> -1 Then Cells(ComboBox1.ListIndex + 2, 8).Select
'Le mail est adressé sur la base de la cellule active :
MailAd = ActiveCell

'On peut sélectionner une plage de cellule pour effectuer un copier dans le mail :
'Range("B8:j32").Select
'Selection.Copy

'On peut mettre une personne en copie en ajoutant une textbox
'Copie = TextBox .Text 'bien mentionner le n° de la textbox
Subj = "Votre Texte pour l'objet" 'Objet du message automatique mais on peut faire référence à une TextBox
Msg = Msg & "Bonjour " & TextBox1.Text & ",%0D%0A %0D%0A" 'Corps du message
Msg = Msg & " Votre texte " & ",%0D%0A %0D%0A" & "Votre prenom ou NOM" & "%0D%0A %0D%0A" 'Corps du message
URLto = "mailto:" & MailAd & "?subject=" & Subj & "&body=" & Msg & "&Cc=" ' & copie
ActiveWorkbook.FollowHyperlink Address:=URLto

Enregistrez votre fichier Excel et lancer votre formulaire. et à vous de jouer.....

SUITE => Insérer ScrollBars (ou barre de défilement) :

Maintenant que vous maîtriser la création d'un formulaire, nous allons aménager celui-ci afin qu'il soit plus agréable en utilisation.

Surtout si vous avez une multitude de Textbox et autres... Ce qui nous fait apparaître à l'écran un énorme Formulaire, car il faut bien tout y mettre, mais ce n'est pas forcément esthétique.

Mais il y a toujours une solution.

Celle-ci consiste à réduire le formulaire (éventuellement) et d'utiliser des barres de défilement.

Verticale, ou Verticale et horizontale, ou uniquement horizontale.

Donc voici la procédure :

Accéder aux propriétés du formulaire avec **Alt + f11**.

Dans cette fenêtre Propriétés accéder à la ligne Scrollbars.

Sur cette même ligne vous avez la mention **0 - fmScrollBarsNone**.

MouseIcon	(Aucun)
MousePointer	0 - fmMousePointerDefault
Picture	(Aucun)
PictureAlignment	2 - fmPictureAlignmentCenter
PictureSizeMode	0 - fmPictureSizeModeClip
PictureTiling	False
RightToLeft	False
ScrollBars	0 - fmScrollBarsNone
ScrollHeight	0
ScrollLeft	0
ScrollTop	0
ScrollWidth	0
ShowModal	True
SpecialEffect	0 - fmSpecialEffectFlat
StartPosition	1 - CenterOwner

Sélectionnez dans la liste déroulante :

1 - fmScrollBarsHorizontal pour le choix d'une barre de défilement Horizontal (en bas du formulaire)

2 - fmScrollBarsVertical pour le choix d'une barr de défilement Vertical (à droite du formulaire)

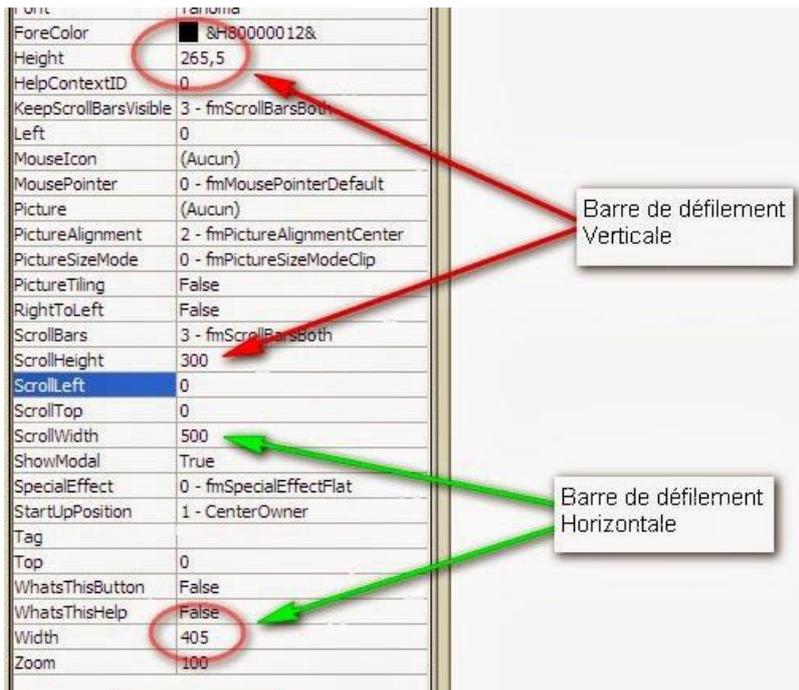
3 - fmScrollBarsBoth pour le choix des deux barres de défilement (Horizontal et vertical)

Ensuite paramétrez la taille des barres de défilement dans ScrollHeight ou ScrollWidth.

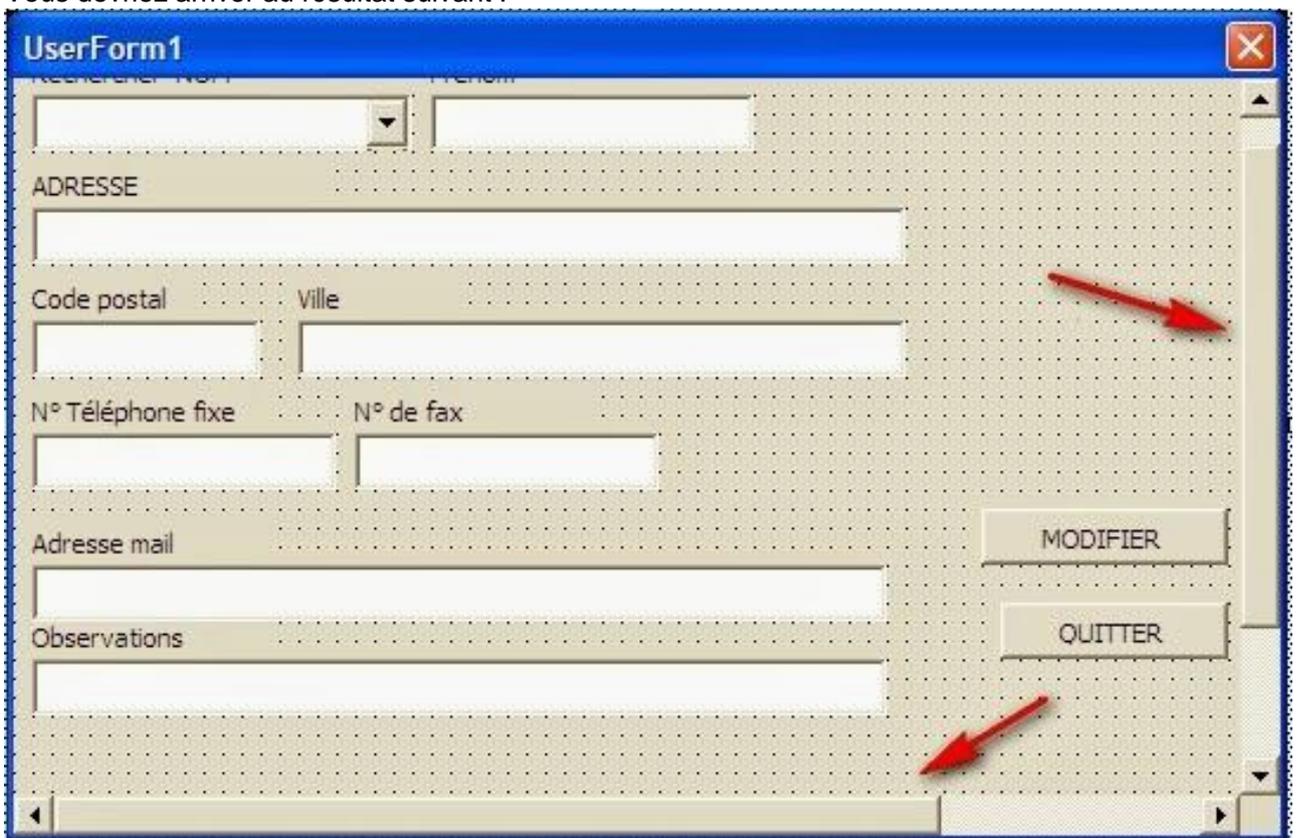
RightToLeft	False
ScrollBars	3 - fmScrollBarsBoth
ScrollHeight	0
ScrollLeft	0
ScrollTop	0
ScrollWidth	0
ShowModal	True
SpecialEffect	0 - fmSpecialEffectFlat
StartPosition	1 - CenterOwner

Attention les tailles de ces barres doivent toujours être supérieures à la taille du formulaire

Comme dans l'exemple ci-après :



Vous devriez arriver au résultat suivant :



Vous pouvez aménager à votre guise la taille sur formulaire et la taille des barres de défilement. A vous de jouer !!!

Agrandir ou réduire son formulaire avec un outils Zoom.

J'ai longtemps chercher sur la toile, mais j'ai trouvé et épuré le procédé.

Il m'est très utile dans le cadre de mon travail car mes collègues n'ont pas forcément un environnement PC, et un formulaire créer sur PC n'est pas très beau sur un environnement MAC.

De plus suivant la grandeur de l'écran de travail, celui-ci n'est non plus très esthétique.

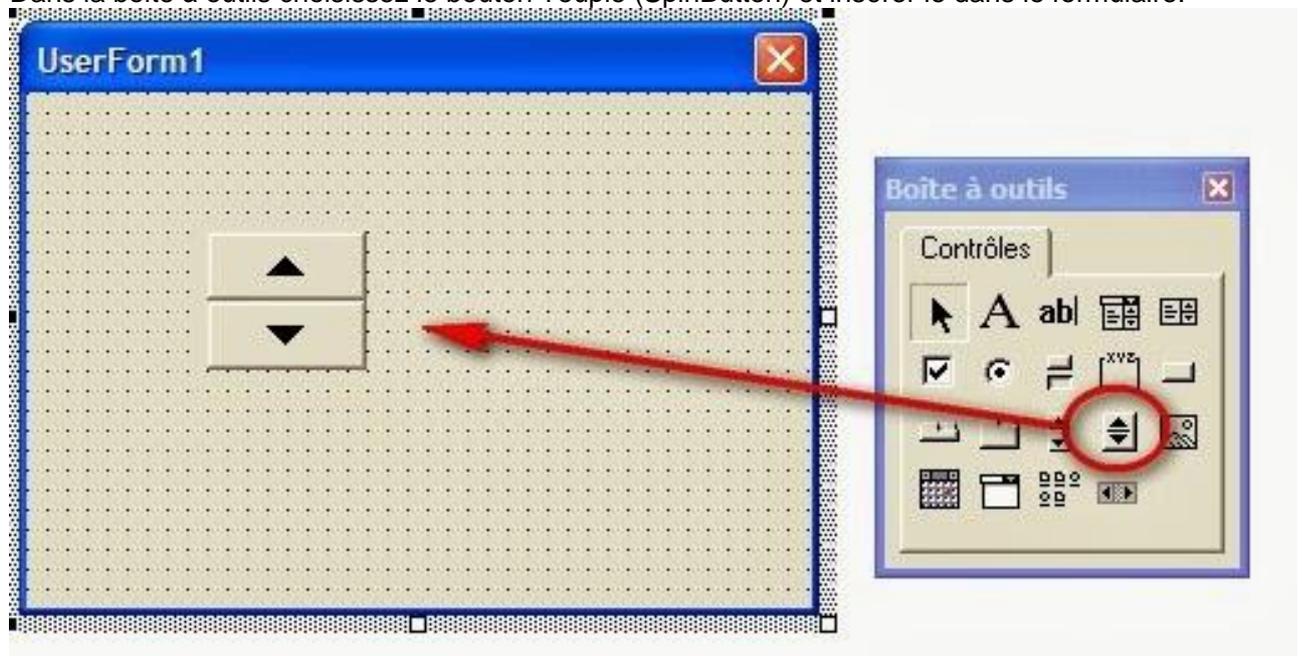
Pour ma part je travaille sur deux écrans simultanément mais pas de même taille (1280x800 pixels et 1680x1050 pixels). Donc ce procédé ZOOM m'est très utile pour afficher mon formulaire à la taille souhaitée suivant mon écran de travail.

Tous les éléments du formulaire seront réduit ou augmenter en taille suivant votre choix via le bouton "ZOOM"

Donc voici la procédure.

Accéder aux propriétés du formulaire avec **Alt + f11**.

Dans la boite à outils choisissez le bouton Toupie (SpinButton) et insérer le dans le formulaire.



1) Ajouter le programme suivant juste en dessous **Option Explicit**
(Si celle-ci n'existe pas il faut simplement l'ajouter)

```
'-----  
'Option Explicite pour le ZOOM soit SpinButton1  
Dim Coef As Long  
Dim ActionSpin As String  
Dim ActionList As String  
'-----
```

2) Dans le programme **Private Sub UserForm_Initialize()**
(Si celui-ci n'existe pas il faut simplement le créer)

Ajouter le code suivant :

```
'-----Zoom à 100% + valeur Minimale et maximale du Zoom -----  
With Me.SpinButton1
```

```
.Value = 100 'valeur de base
.Min = 90 'Valeur mini du zoom en %
.Max = 125 'Valeur maxi du zoom en %
```

```
End With
Reglage
```

3) Pour le réglage de l'ouverture de l'Userform à 100%

Ajouter le code suivant :

```
'-----
'REGLAGE de l'userforme SpinButton
Private Sub Reglage()
With Me
Coef = .SpinButton1 - 100
.Height = ((280 / 100) * Coef) + 280 'correspond à la Hauteur de l'userform
.Width = ((425 / 100) * Coef) + 425 'correspond à la largeur de l'userform
.Zoom = .SpinButton1
End With
End Sub
'-----
```

4) Pour l'action Zoom + et Zoom- sur le bouton SpinButton (ou toupie)

Ajouter le code suivant :

```
'-----
'ZOOM "PLUS" avec SpinButton1
Private Sub SpinButton1_SpinUp()
ActionSpin = "Plus"
Reglage
End Sub

'ZOOM "MOINS" avec SpinButton1
Private Sub SpinButton1_SpinDown()
ActionSpin = "Moins"
Reglage
End Sub
'-----
```

5) et pour le fun, si vous souhaitez connaître le pourcentage de réduction ou d'augmentation, ajoutez le code suivant :

```
'-----
'TITRE du formulaire avec % du zoom
'Facultatif
Private Sub SpinButton1_Change()
'Remarque : l'action Change étant un événement en amont de SpinUp ou SpinDown
Me.Caption = " FORMULAIRE Zoom à : " & Me.SpinButton1 & " %"
End Sub'-----
```

Résultat :

Titre de formulaire avec pourcentage d'augmentation ou de réduction

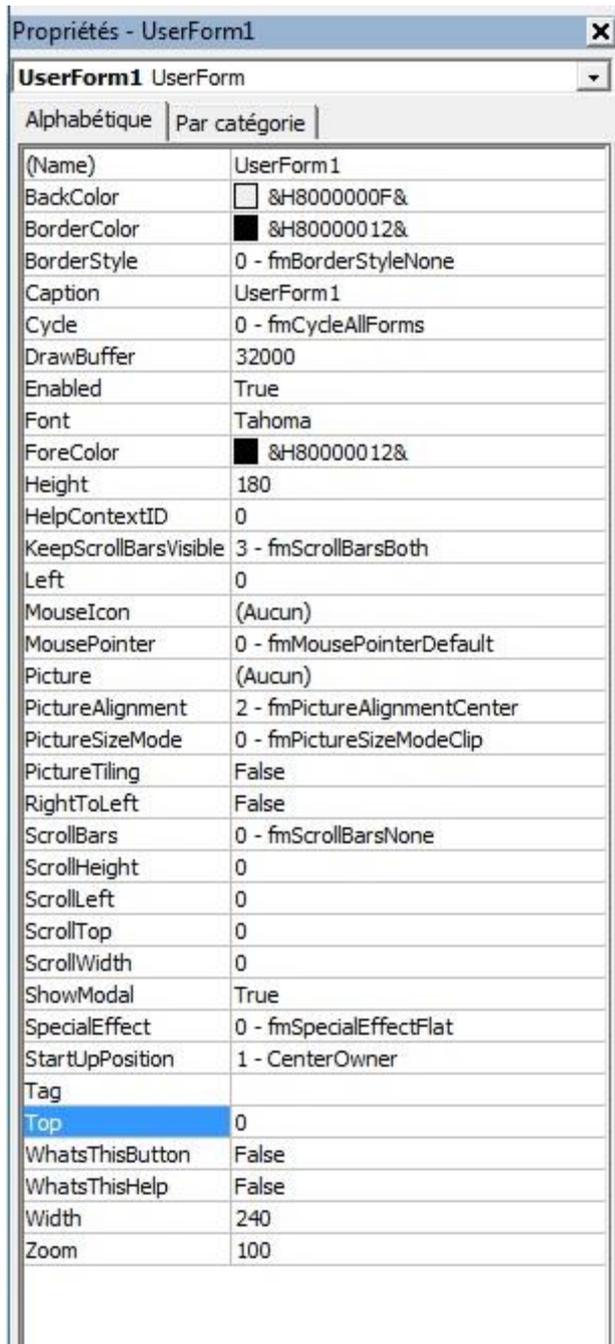
Appuyez sur ce bouton pour augmenter ou diminuer la taille du formulaire afin de l'adapter à la taille de votre écran

The screenshot shows a software window titled "FORMULAIRE Zoom à : 92 %". The window contains a form with the following fields and controls:

- Rechercher NOM (dropdown menu)
- Prénom (text input)
- ADRESSE (text input)
- Code postal (text input)
- Ville (text input)
- N° Téléphone fixe (text input)
- N° de fax (text input)
- Adresse mail (text input)
- Observations (text input)
- MODIFIER button
- QUITTER button
- Zoom control (up and down arrows)

Personnaliser votre FORMULAIRE :

Si vous souhaitez personnaliser votre USERFORM (Formulaire), utiliser la fenêtre de propriétés. Changer manuellement le nom, une fonctionnalité, l'apparence du formulaire...etc...



Description succincte des différentes propriétés :

=> **BackColor** Spécifie la couleur de fond
Choix par couleur "Système" ou "Palette"

=> **BorderColor** Spécifie la couleur de bordure
Choix par couleur "Système" ou "Palette"

=> **BorderStyle** Spécifie le style de bordure
Choix : 0 - fmBorderStyleNone => par défaut
1 - fmBorderStyleSingle

=> **Caption** Spécifie le texte affiché dans la barre de titre

=> **Cycle** Spécifie l'action à effectuer lorsque l'utilisateur quitte le dernier contrôle
Choix : 0 - fmCydeAllForms => par défaut
2 - fmCydeCurrentForm

=> **DrawBuffer** Spécifie le nombre de pixels de la mémoire tampon vidéo utilisée pour le rafraîchissement de l'image

=> **Enabled** Spécifie si l'objet peut recevoir le focus et répondre aux événements générés par l'utilisateur
Choix : Fasl
True

=> **Font** Spécifie la type de police de caractère
Choix par sélection de la police de caractère, Style, Taille, Barré et/ou souligné

=> **Forecolor** : Spécifie la couleur de la police de caractère

=> **Height** : Définit la dimension verticale

=> **HelpContextID** : Spécifie la rubrique d'un fichier d'aide

=> **KeepScrollBarsVisible** : Spécifie si les barres de défilement sont visibles

=> **Left** : Définit la position par rapport au bord gauche de l'application

=> **Mouseelcon** : Affecte un icône personnalisé

=> **MousePointer**..... Spécifie le type de pointeur

=> **Picture**..... Spécifie l'image de fond de l'userform

=> **PictureAlignment** : Spécifie l'alignement de l'image de fond

=> **PictureSizeMode** : Spécifie comment afficher l'image de fond

=> **Picture Tiling** Permet d'afficher l'image en mosaïque

=> **RightToLeft** Indique le sens d'affichage du texte et contrôle l'apparence visuelle sur un système bidirectionnel.

=> **ScrollBars** Indique si les barres de défilement verticales et horizontales doivent être affichées

=> **ScrollHeight** Spécifie la hauteur de la zone totale pouvant être affichée en déplaçant la barre de défilement

=> **ScrollLeft** Spécifie la distance, du bord gauche de la partie visible, à partir du bord gauche de l>UserForm

=> **ScrollTop** Spécifie la distance, du bord supérieur de la partie visible, à partir du bord supérieur de l>UserForm

=> **ScrollWidth** Spécifie la largeur de la zone totale pouvant être affichée en déplaçant la barre de défilement

=> **ShowModal** Indique si la boîte de dialogue est modale

=> **SpecialEffect** Spécifie l'aspect du UserForm à l'écran

=> **StartupPosition** Indique la position du UserForm lors de sa première apparition

=> **Tag** Permet de stocker des informations supplémentaires

=> **Top** Définit la position par rapport au bord supérieur de l'application

=> **WhatsThisButton** Indique si le bouton d'aide "Qu'est-ce que c'est" apparaît sur la barre de titre

=> **WhatsThisHelp** Indique si l'aide contextuelle utilise la fenêtre automatique fournie par l'aide de Windows ou la fenêtre d'aide principale

=> **Width** Définit la dimension horizontale

=> **Zoom** Spécifie le changement de taille de l'objet

=> **Visible** Spécifie si l'objet est masqué ou affiché.

A vous de jouer...