



FICHE DE COURS GRATUITE

VBA EXCEL | FORMULAIRES

Créer un premier UserForm

Un formulaire de saisie, c'est la meilleure façon d'éviter les erreurs dans un tableau partagé : l'utilisateur remplit des champs propres, la macro contrôle et range tout au bon endroit. Voici comment construire le premier, de bout en bout.

Version	V1.2
Dernière mise à jour	29/10/2026
Niveau	3 / 3, Avancé
Durée de lecture	10 min
Auteur	Clément Delobel



Version en ligne
mesetudesdecas.fr/vba/userform-premier-formulaire.html

01 Définition

EN UNE PHRASE

Un UserForm est une **fenêtre personnalisée** que vous dessinez dans l'éditeur VBA : zones de texte, listes, cases à cocher et boutons, reliés à du code.

02 Créer le formulaire

- Ouvrez l'éditeur avec `Alt + F11`, puis **Insertion > UserForm**.
- La **Boîte à outils** s'affiche. Si ce n'est pas le cas : **Affichage > Boîte à outils**.
- Dessinez vos contrôles : un intitulé (Label) et une zone de texte (TextBox) pour le nom, une liste déroulante (ComboBox) pour la ville, deux boutons (CommandButton).
- Dans la fenêtre **Propriétés** (`F4`), renommez chaque élément : propriété **Name** pour le code, propriété **Caption** pour le texte affiché.

💡 ASTUCE

Nommer avec un préfixe

`txtNom`, `cboVille`, `btnValider`, `btnAnnuler` : le préfixe indique le type de contrôle, et le code devient lisible sans retourner voir le formulaire.

Contrôle	Name	Caption
UserForm	frmClient	Nouveau client
TextBox	txtNom	(aucune)
ComboBox	cboVille	(aucune)
CommandButton	btnValider	Valider

Contrôle	Name	Caption
CommandButton	btnAnnuler	Annuler

03 Ouvrir le formulaire

Le formulaire s'affiche avec la méthode **Show**, depuis une macro classique. Reliez cette macro à un bouton dans la feuille (**Développeur > Insérer > Bouton**).

DANS UN MODULE STANDARD

```
Sub OuvrirFormulaire()  
    frmClient.Show  
End Sub
```

04 Préparer le formulaire à l'ouverture

Double-cliquez sur le fond du formulaire pour ouvrir son code, puis choisissez l'événement **Initialize** : il s'exécute juste avant l'affichage.

REPLIR LA LISTE DES VILLES

```
Private Sub UserForm_Initialize()  
    With Me.cboVille  
        .AddItem "Lille"  
        .AddItem "Lens"  
        .AddItem "Arras"  
        .AddItem "Valenciennes"  
    End With  
End Sub
```

Me désigne le formulaire lui-même.

05 Le bouton Valider

Double-cliquez sur le bouton Valider dans le formulaire : l'éditeur crée la procédure de son événement **Click**.

CONTRÔLER, ÉCRIRE, FERMER

```
Private Sub btnValider_Click()  
    Dim ligne As Long  
  
    ' 1. Contrôler la saisie  
    If Trim(Me.txtNom.Value) = "" Then  
        MsgBox "Le nom est obligatoire.", vbExclamation  
        Me.txtNom.SetFocus  
        Exit Sub  
    End If  
  
    ' 2. Écrire dans la première ligne libre  
    With Worksheets("Clients")  
        ligne = .Cells(.Rows.Count, 1).End(xlUp).Row + 1  
        .Cells(ligne, 1).Value = Trim(Me.txtNom.Value)  
        .Cells(ligne, 2).Value = Me.cboVille.Value  
        .Cells(ligne, 3).Value = Date  
    End With  
  
    ' 3. Fermer le formulaire  
    Unload Me  
End Sub  
  
Private Sub btnAnnuler_Click()  
    Unload Me  
End Sub
```

06 Pièges à éviter

⚠ PIÈGE

Unload ou Hide

`Unload Me` ferme le formulaire et efface ses valeurs. `Me.Hide` le cache seulement : à la prochaine ouverture, les anciennes saisies sont encore là.

⚠ PIÈGE

Le code au mauvais endroit

Le code des boutons se trouve dans le module du formulaire (double-clic sur le contrôle), pas dans un module standard. La macro qui fait `frmClient.Show`, elle, va dans un module standard.

⚠ PIÈGE

Renommer après avoir codé

Si vous renommez `CommandButton1` en `btnValider` après avoir écrit son code, la procédure `CommandButton1_Click` ne sera plus jamais appelée. Renommez d'abord, codez ensuite.

⚠ PIÈGE

Tout est du texte

Une TextBox renvoie toujours du texte. Pour un montant ou une quantité, vérifiez avec `IsNumeric` puis convertissez avec `Cdbl` ou `CLng` avant d'écrire dans la feuille.