

Véronique Plihon-Heiwy

# VBA : l'essentiel

Cours et exemples de programmes  
pour *Visual Basic For Applications*



# Chapitre 1 - Introduction

## Prérequis à l'apprentissage de VBA : connaître Excel

Excel est un tableur permettant de créer des classeurs. Un classeur est composé de feuilles de calcul et de graphiques. Dans une feuille de calcul se trouvent des cellules. Chaque cellule est identifiée par le numéro de ligne et par le nom de colonne dans laquelle elle se trouve. A2 par exemple, désigne la cellule positionnée dans la colonne A et à la ligne 2. Les colonnes peuvent également être référencées par un numéro. A2 devient alors L2C1 (la cellule positionnée en deuxième ligne (L2) et en première colonne (C1)). Dans une cellule il est possible de saisir un texte, un nombre ou une formule.

Excel permet de créer des **tableaux de données**, de créer des **formules** dans des cellules et propose une grande variété de **fonctions** organisées en catégories (mathématiques, statistiques, logiques, de recherche, de base de données, de date, etc.). Excel propose également une grande variété de **commandes** pour trier des tableaux de données, filtrer des données, créer des mises en forme conditionnelles, créer des listes de choix, créer des tableaux croisés dynamiques, insérer des graphiques, mettre en forme des tableaux de données ou créer des formats personnalisés d'affichage pour les nombres notamment.

Malgré tout, Excel présente certaines **limites** : il peut arriver que les utilisateurs ne trouvent pas de fonctions d'Excel correspondant à leurs besoins ou qu'ils souhaitent automatiser l'exécution de commandes Excel. La solution pour dépasser ces limites réside dans le langage de programmation VBA.

## I. Importance de Visual Basic for Applications (VBA)

**Visual Basic for Applications** (VBA) est le langage de programmation que Microsoft recommande d'utiliser pour les nouvelles applications depuis la sortie d'Excel 5, en 1993.

La stratégie de Microsoft est de se servir de ce langage VBA pour uniformiser l'automatisation de tous ses produits bureautiques, y compris le traitement de textes Word et le gestionnaire de bases de données Access.

**Visual Basic pour Applications**, en anglais Visual Basic for Applications, est donc un langage d'une grande importance dans la stratégie Microsoft. C'est lui que nous étudions ici. Nous abrégons son nom en **VBA**.

**VBA (Visual Basic for Applications)** est un langage complet, moderne, qui n'a pratiquement plus rien à voir avec les premières versions du langage BASIC. Les actions réalisées par le code VBA s'effectuent à la demande de l'utilisateur, ce code est toujours écrit sous forme de sous-programmes.

- Les sous-programmes de type **procédure** (Subroutine ou **Sub**) servent à réaliser des actions. Le cas le plus simple est celui d'une procédure *sans paramètre*, qui peut être **appelée par un clic sur un bouton** (c'est ainsi que se programme en VBA l'équivalent des anciennes macros). Les procédures *avec paramètres* ne peuvent être appelées que depuis une autre procédure ou une fonction.
- Les sous-programmes de type **fonction (Function)** ont pour but de calculer une valeur et de la renvoyer à l'appelant. Cet appelant peut être une autre procédure ou fonction, ou une formule d'un tableau Excel. C'est ainsi qu'il est possible de créer de nouvelles fonctions de feuille de calcul. L'appel d'une fonction peut également être utilisé comme terme dans une expression.

**Exemple :**

$$z = (\max(x, y) + 3) / 6$$

L'appel de fonction  $\max(x, y)$  est inclus dans une expression où sa valeur est augmentée de 3, puis divisée par 6 avant d'être affectée à z.

**VBA (Visual Basic for Applications)** possède une structure de blocs pour les instructions conditionnelles :

If... End If

Select Case... End Select

et pour les instructions répétitives (boucles) :

Do... Loop

For... Next

For Each... Next

While... Wend

VBA ne crée des blocs que dans les cas d'instructions conditionnelles et des instructions répétitives (boucles) vues plus haut.

**VBA (Visual Basic for Applications)** est un langage de programmation orienté objet. Les objets y ont trois utilités :

- fournir une **interface** entre le programme VBA et le contenu des feuilles de calcul Excel : par exemple les objets de classe Range, qui représentent des ensembles de cellules (il est possible de traduire Range par étendue, zone, ou domaine) ;
- réaliser des **interfaces graphiques** permettant au programme VBA de communiquer avec l'utilisateur (objets **boutons**, objets **boîtes de dialogue**, objets **zones de texte**) ;
- permettre au programme VBA de **communiquer avec d'autres applications** en appelant des objets d'autres applications (Word ou Access) à partir du VBA d'Excel. L'utilisation d'Automation (anciennement OLE automation) ou d'échange dynamique de données (DDE) est nécessaire.

**VBA (Visual Basic for Applications)** permet au programmeur de définir et d'utiliser des **procédures** et des **fonctions**, avec passage **d'arguments** soit **positionnel** (les arguments sont identifiés par leur **position** dans une liste), soit **par nom**. Par ailleurs, ces paramètres peuvent être passés **par valeur** (une copie est passée) ou **par référence** (leur adresse est passée). Les procédures et fonctions sont le moyen naturel d'étendre le langage pour l'adapter aux besoins.

## II. Modules

Microsoft a doté les tableurs d'une véritable structure de langage informatique, où les programmes sont séparés des données (ou presque, car les cellules peuvent toujours contenir des appels de fonctions).

Pour séparer les programmes des données, les **procédures et fonctions VBA (Visual Basic for Applications)** depuis Excel 97 se situent dans des structures particulières nommées **modules**.

Il est également possible d'associer des procédures à des **événements** (par exemple « *clic* ») liés à des **objets** (par exemple « *bouton de commande* ») présents sur des feuilles de calcul ou dans des boîtes de dialogue.

## III. Langue utilisée

L'aide en ligne, quand elle est disponible, n'est plus valable que pour des instructions et des noms d'objets exprimés en **anglais**.

Cette restriction ne vaut que pour les mots-clés du langage, les **noms** définis par le programmeur (**types, variables, procédures et fonctions**) peuvent être des chaînes de caractères quelconques ou presque.

Depuis Excel 97, **l'anglais** s'est imposé comme langue de programmation pour VBA.

## **IV. Présentation de l'environnement VBA**

Cette section est consacrée à la présentation de l'environnement VBA, tant pour les versions d'Excel pour PC que pour Mac.

Prendre connaissance de cet environnement est nécessaire pour avoir les clés pour créer, sauvegarder, mettre au point et exécuter ses propres programmes VBA.

### **IV.1 Prérequis : Disposer du menu « Développeur »**

Pour programmer avec le langage VBA, il faut disposer d'un onglet spécifique dénommé « Développeur » dans le ruban.

Pour faire apparaître le menu « Développeur » :

**Sur PC :** Fichier/Options/Personnaliser le ruban.

Dans les onglets principaux, Cocher « Développeur ».

Puis valider par OK.

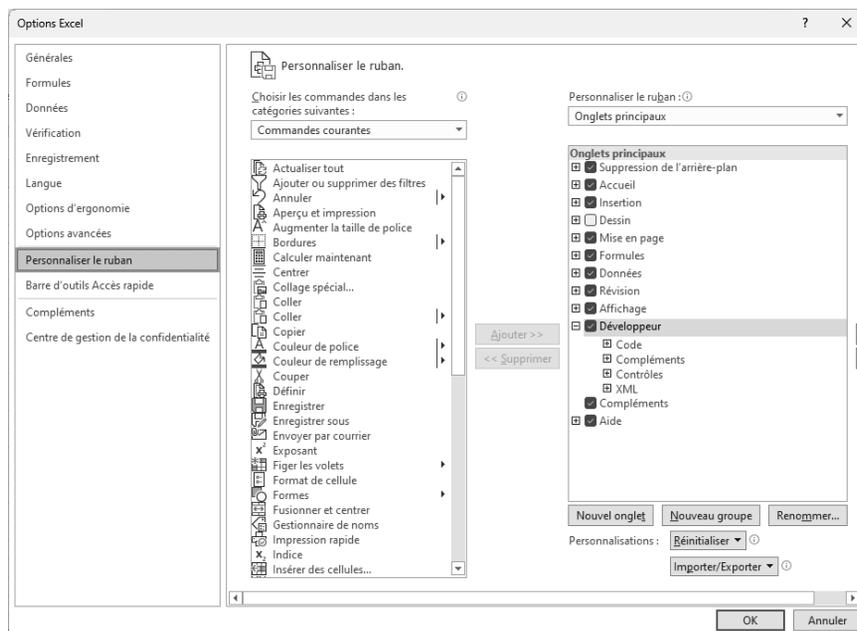


Figure 1 : Personnaliser le ruban (sur PC) pour faire apparaître le menu Développeur

L'onglet « Développeur » apparaît maintenant dans votre barre de menus ou ruban.

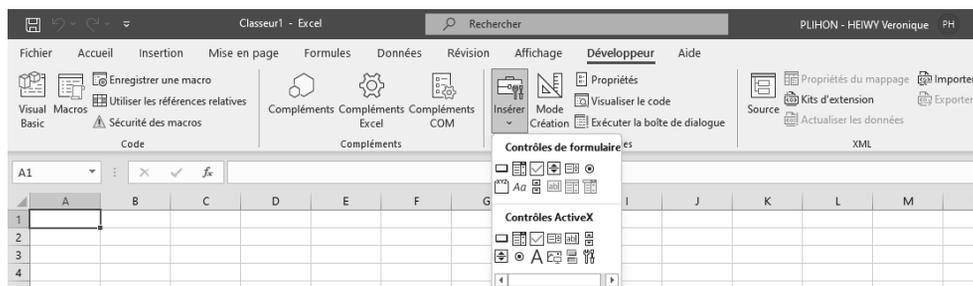


Figure 2 : Le menu Développeur d'Excel pour PC

## Sur Mac : Excel/ Préférences/ Ruban et barre d'outils

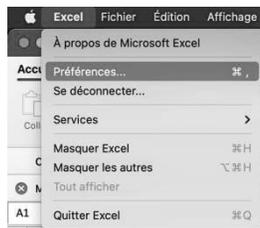


Figure 3 : Excel /Préférences...

Dans les « Préférences d'Excel », choisir, dans le groupe « Création » la troisième icône en partant de la gauche dénommée « Ruban et barre d'outils » (voir Figure 4). Un clic sur cette icône fait apparaître la boîte de dialogue de la Figure 5.

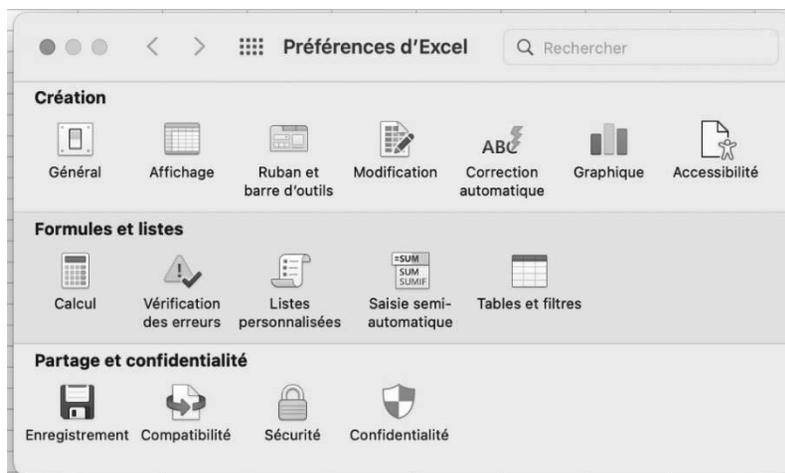


Figure 4 : Préférences d'Excel pour « Ruban et barre d'outils »

Dans l'onglet « Ruban », pour « Personnaliser le ruban » (à droite), choisir « Onglets principaux » et cocher la case « Développeur ».



Figure 5 : Personnaliser le ruban (sur Mac) pour faire apparaître le menu Développeur

L'onglet « Développeur » apparaît maintenant dans votre barre de menus ou ruban.

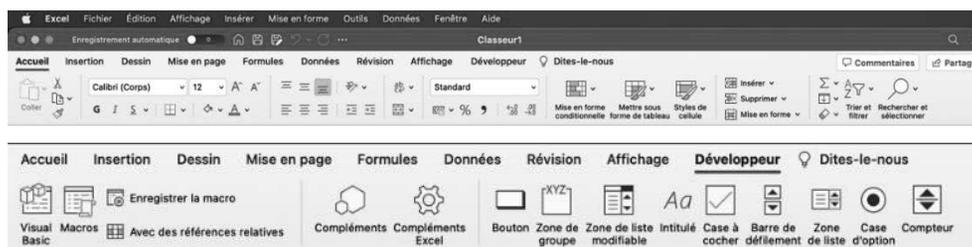


Figure 6 : Le menu Développeur d'Excel pour Mac

## IV.2 Présentation du menu Développeur

Dans Excel pour PC comme dans Excel pour Mac, le menu « Développeur » comporte les icônes :

1. **Visual Basic** : pour accéder à l'éditeur Visual Basic
2. **Macros** : pour afficher la liste des macros qu'il est possible d'exécuter en choisissant celle qui convient dans la liste
3. **Enregistrer une macro** : pour démarrer l'enregistrement de macro. Une fois l'enregistrement démarré, le bouton permet d'arrêter l'enregistrement.

**Dans la version Excel pour PC,**

4. **Insérer** : pour afficher la **boîte à outils** contenant les contrôles graphiques de formulaires qu'il est possible d'insérer dans une feuille de calcul pour créer rapidement une interface graphique.

**Dans la version Excel pour Mac,**

la **liste des contrôles graphiques** (équivalente aux contrôles de formulaire de la boîte à outils de la version Excel pour PC) qu'il est possible d'insérer dans une feuille de calcul apparaît directement.

## IV.3 Présentation de l'interface de l'éditeur Visual Basic

L'éditeur Visual Basic est accessible par un clic sur le bouton « Visual Basic » du menu « Développeur ». Il permet de créer du code VBA, de sauvegarder le projet Excel/VBA au format **.xlsm** (classeur Excel avec macros), d'afficher du code, de compiler le code, de mettre au point le code en l'exécutant pas à pas et d'exécuter le code sans mode pas à pas, directement, autant de fois que nécessaire.