

Programmation Excel/VBA

INTRODUCTION GÉNÉRALE À EXCEL VBA

MASTER SES

UNIVERSITÉ NANTERRE LA DÉFENSE
SANA BEN HAMIDA

Pourquoi VBA ?

2

- *Non pas le meilleur des langages!*
 - *Non pas le plus utilisé!*
 - *Langage propriétaire!*
-
- ➔ Utilisation massive de la distribution Microsoft Office
 - ➔ Utilisation massive dans les entreprises
 - ➔ Beaucoup de fonctions pré-définies : intérêt pour la manipulation de données etc...
 - ➔ Possibilité de créer du code avec l'enregistreur de macro

Historique

3

- Basic : Beginner's All-purpose Symbolic Instruction Code (63)
- **Visual Basic for Applications** (début des années 90s)
- Implémentation Microsoft du langage **VB**
- Langage interprété
- Langage procédural tendant vers l'objet

Caractéristiques

4

VBA, Langage interprété

- Se distingue des langages compilés
 1. Compilation
 2. Exécution
- Interprétation \Leftrightarrow Compilation à la volée \Leftrightarrow exécution ligne à ligne.
- Interprété : moins efficace, plus facile à gérer

Caractéristiques

5

Langage Procédural

- Approche procédurale : « Le programmeur conçoit un algorithme comme solution à son problème, c'est à dire **une suite d'opérations sur des variables**.
- Pour toute sous suite des opérations répétitives ou formant un ensemble cohérent, on crée une **sous-fonction** ou une **sous-procédure** (Par exemple, pour évaluer un polynôme, il est nécessaire de disposer d'une fonction puissance).

Caractéristiques

6

- **Langage Objet**
- Approche objet : conception d'objets et attribution de comportements
- VBA- Excel dispose d'une hiérarchie d'objets MS
 - ➔ Création et utilisation d'objets prédéfinis
 - ➔ L'utilisateur ne définit pas ses propres objets

Pourquoi utiliser VBA

7

- Pour **automatiser** des tâches (macros)
- Pour **étendre et personnaliser** une application Microsoft Office : on peut ajouter ou modifier des commandes, des menus, des barres d'outils, des boîtes de dialogue
- VBA permet de **réutiliser les mêmes instructions** ou codes de programmation dans les différentes applications Microsoft Office.
- Il permet de **communiquer** entre les différentes applications hôtes, ce qui permet de profiter de la puissance de deux applications.

VBE: Visual Basic Editor

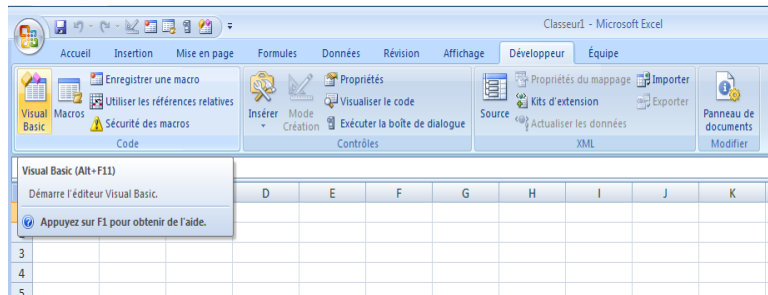
8

- permet de créer des commandes personnalisées à la fois souples et performantes contenant des instructions VBA. On a accès à toutes les possibilités de l'application hôte

VBE: Visual Basic Editor

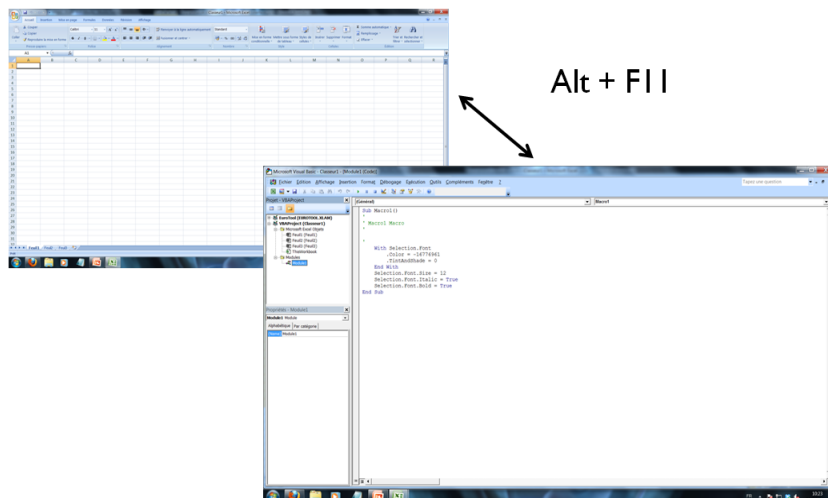
9

- Pour y accéder, sélectionner Visual Basic dans le menu « Développeur »



VBE: Visual Basic Editor

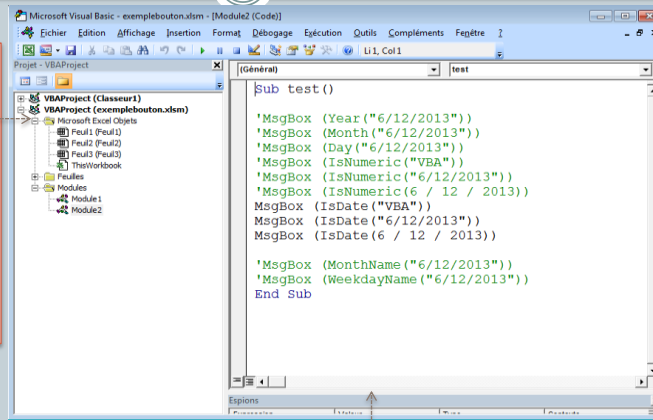
10



VBE: Visual Basic Editor

11

Fenêtre VBAProject. Elle présente les différents projets ouverts et permet de naviguer facilement entre vos différentes feuilles de codes VBA.



Fenêtre Code. C'est l'endroit où vous allez saisir votre code VBA

VBE: Visual Basic Editor

12

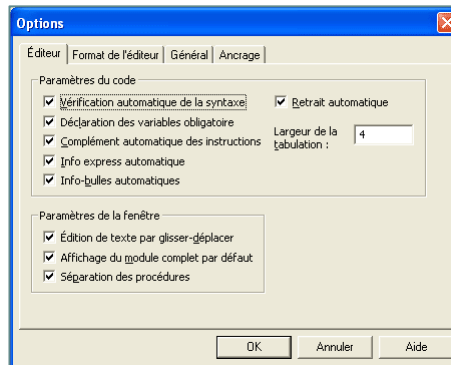
L'aspect de votre éditeur de macros peut être différent : Il est personnalisable car chaque fenêtre peut être masquée puis réaffichée par le menu "Affichage".

Il est important de bien configurer l'éditeur de macros. En effet, VBE peut vous aider dans l'écriture de votre code et le mettre en forme de façon à ce qu'il soit plus facile à lire.

Sous VBE, lancer le menu "Outils-Options"

VBE: Visual Basic Editor

13



Vérification automatique de la syntaxe :

Vérification automatiquement de la syntaxe lors de la saisie d'une ligne de code.

VBE: Visual Basic Editor

14

Déclarations de variables obligatoires :

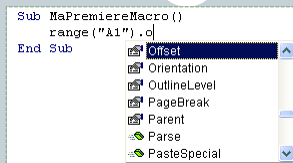
Sous VB, la déclaration de variables n'est pas obligatoire : je vous conseille de cocher cette option.

Si la case est cochée, l'instruction "Option Explicit" est ajoutée dans les déclarations générales de tout nouveau module.

Complément automatique des instructions :

Cette option permet à VBE de vous aider dans la saisie de votre code

VBE: Visual Basic Editor



Vous comprendrez très vite son utilité lorsque vous saisirez vos premières lignes de codes.

Info express automatique :

Encore une option très utile. Elle affiche les différents arguments que possède la fonction que vous venez de taper :

```
Sub MaPremiereMacro()  
  range(|  
End Range(Cell1, (Cell2) As Range
```

VBE: Visual Basic Editor

16

Info-bulles automatique :

Indispensable lors d'un débogage pas à pas. Elle permet l'affichage de vos variables.

Retrait automatique :

Permet à VBE de placer chaque ligne de code au même niveau que la ligne précédente. Le retrait de lignes se fait par les touches "Tab" et "Shift+Tab". Cette option est nécessaire pour une bonne lecture du code VBA.

Paramètres de la fenêtre :

L'affichage du module complet par défaut permet l'affichage de toutes les procédures d'un même module et la séparation des procédures oblige VBE à créer des traits entre chaque procédures.

Ecrire un programme VBA/Excel

19

Débogage:

- **F8** pour exécution pas à pas
- **Maj + F9** pour « espionner » le code (voir les valeurs courantes des variables)

Éléments de base

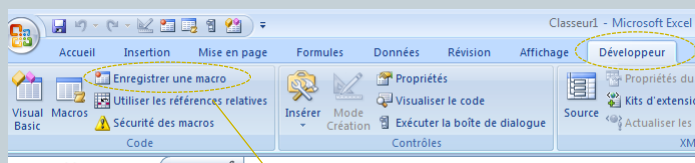
20

- Tout programme est une suite d'instructions, avec une syntaxe précise et dépourvue de toute ambiguïté
- **1 ligne = 1 instruction**
- Si une instruction ne tient pas sur une ligne, il faut mettre en fin de ligne un blanc souligné (underscore _) et passer à la ligne suivante
- Commentaire: le précéder par une apostrophe « ' »
Exemple: ' ceci est un commentaire

L'enregistreur de macro

L'enregistreur de macro

22



Enregistrer une macro

Nom de la macro :
Macro1

Touche de raccourci :
Ctrl+ ☐

Enregistrer la macro dans :
Ce classeur

Description :

OK Annuler

L'enregistreur de macro

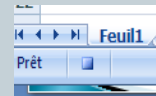
23

Exemple 1

1. Lancer l'enregistreur de macro
2. Mettre la cellule active en gras, taille 14, couleur bleu foncé
3. Arrêter l'enregistrement
4. Visualiser le code correspondant



	A
1	Titre
2	



L'enregistreur de macro

le code

```
Sub MacroTitre()
'
' MacroTitre Macro
'
    With Selection.Font
        .Name = "Calibri"
        .Bold = True
        .Size = 14
        .Strikethrough = False
        .Superscript = False
        .Subscript = False
        .OutlineFont = False
        .Shadow = False
        .Underline = xlUnderlineStyleNone
        .ThemeColor = xlThemeColorLight2
        .TintAndShade = 0.399975585192419
        .ThemeFont = xlThemeFontMinor
    End With
End Sub
```

Par défaut