

FORMATION VBA EXCEL 2003

Livret 1 – Les macros – introduction à la programmation

Niveau 2

Microsoft®

Visual Basic®
for Applications



Thierry TILLIER

Ce support de cours est réservé à un usage personnel. Toute utilisation et diffusion dans le cadre d'une activité de formation fait l'objet d'une licence de droit de copie. Veuillez nous contacter si vous souhaitez utiliser ce cours dans ce cadre.

Retrouvez tous nos cours bureautique sur <http://www.e-presse.ca/> au Canada

- Cours Access 2007
- Cours Access 2010
- Cours Windows 7
- Cours Windows Vista
- Cours Publisher 2010
- Cours PowerPoint 2007
- Cours PowerPoint 2010
- Cours Outlook 2007
- Cours Outlook 2010
- Cours Excel 2007
- Cours Word 2007

- Cours Word 2010
- Cours VBA Excel 2003

Prochaines parutions :

- Cours Excel 2010
- Cours VBA Excel 2007
- Cours VBA Excel 2010
- Cours VBA Access 2007
- Cours VBA Access 2010

Sommaire

Chapitre 1 INTRODUCTION.....	4
Chapitre 2 Généralités sur les macros.....	5
Chapitre 3 L'enregistreur de macro	6
3.1 Afficher la barre d'outils Visual Basic	6
3.2 Description de la barre d'outils	7
Chapitre 4 Les macros	8
4.1 Créer une macro.....	8
4.2 Arrêter l'enregistrement d'une macro	9
4.3 Exécuter une macro.....	9
Chapitre 5 Les macros simples sur Excel	10
5.1 Exo 1 : Formater une feuille	10
5.2 Le code de la macro.....	11
5.3 Exo 2 Mise à jour d'un document.....	13
Chapitre 6 Les zones de noms	16
Chapitre 7 Les Macros complexes	18
Chapitre 8 Boutons & Menus	20
8.1 Ajouter un menu	20
8.2 Ajouter un bouton	24
Chapitre 9 Vocabulaire de base	26
Chapitre 10 Conclusion	27
Chapitre 11 Corrigés.....	28
11.1 Chapitre 2 exo 2.....	28
11.2 Chapitre 7	28

Chapitre 1 INTRODUCTION

Dans ce livret vous apprendrez à créer des macros simples et des boutons de commandes. Si vous savez créer des macros avec l'enregistreur, ce livret sera davantage une révision avec des compléments d'explications sur certaines fonctionnalités.

Pré-requis : Maîtriser Excel 2003 et ses fonctions de base, savoir faire des macros simples avec Excel 2003. Avoir des notions d'anglais.

Objectifs :

- ☞ Créer des macros simples
- ☞ Modifier des macros
- ☞ Créer des macros complexes en VBA
- ☞ Créer des boutons de commandes

Signalétique



Attention



Information complémentaire

RECOMMANDATION : PRENEZ LE TEMPS DE BIEN LIRE LES INSTRUCTIONS.

Chapitre 2 Généralités sur les macros

VBA est un langage de programmation pour les applications Office (Word - Excel – Access - PowerPoint – Outlook).

Il permet d'automatiser des tâches répétitives ou encore d'effectuer des opérations événementielles.

Par exemple : rechercher des doublons d'une liste dans une feuille ou supprimer des lignes blanches, effectuer une série de calculs complexes, chargés des données à l'ouverture...

Une macro est une suite d'actions enregistrées qui peuvent être ensuite exécutées automatiquement par Excel. Les macros simples peuvent être conçues avec l'enregistreur de macros. Les macros sont écrites dans un langage de programmation : le Visual Basic pour Application (VBA) qui est le langage de développement commun aux applications Microsoft (Word, Access...)

La programmation en VBA permet :

- D'automatiser des travaux répétitifs.
- Développer des applications complètes pour faciliter et guider le travail d'utilisateurs ne connaissant pas Excel.
- De créer des fonctions de calculs en complément de celles d'Excel ou des fonctions personnalisées
- D'automatiser des actions longues (mettre en forme un document, insérer le nom de votre société sur les pieds de pages ...)
- Créer des boîtes de dialogue (pour contrôler la saisie dans les cellules)
- Créer des barres d'outils personnalisées.

Il existe deux types de macros : Les procédures **Sub** et les procédures **Function** :

- ◆ Sub Elles s'exécutent en lançant une commande de macro (par un bouton ou le menu)

Ex : Sub MettreEnPage()

....

End Sub

- ◆ Function Elles retournent une seule valeur – comme le résultat d'un calcul. On les utilise de la même manière que les fonctions d'Excel, en tapant « =NomDeLaFonction(arg) » dans la cellule. Elles apparaissent dans la boîte de dialogue « Insérer un fonction »

Ex : Function RacineCubique(Cellule)

....

End Function

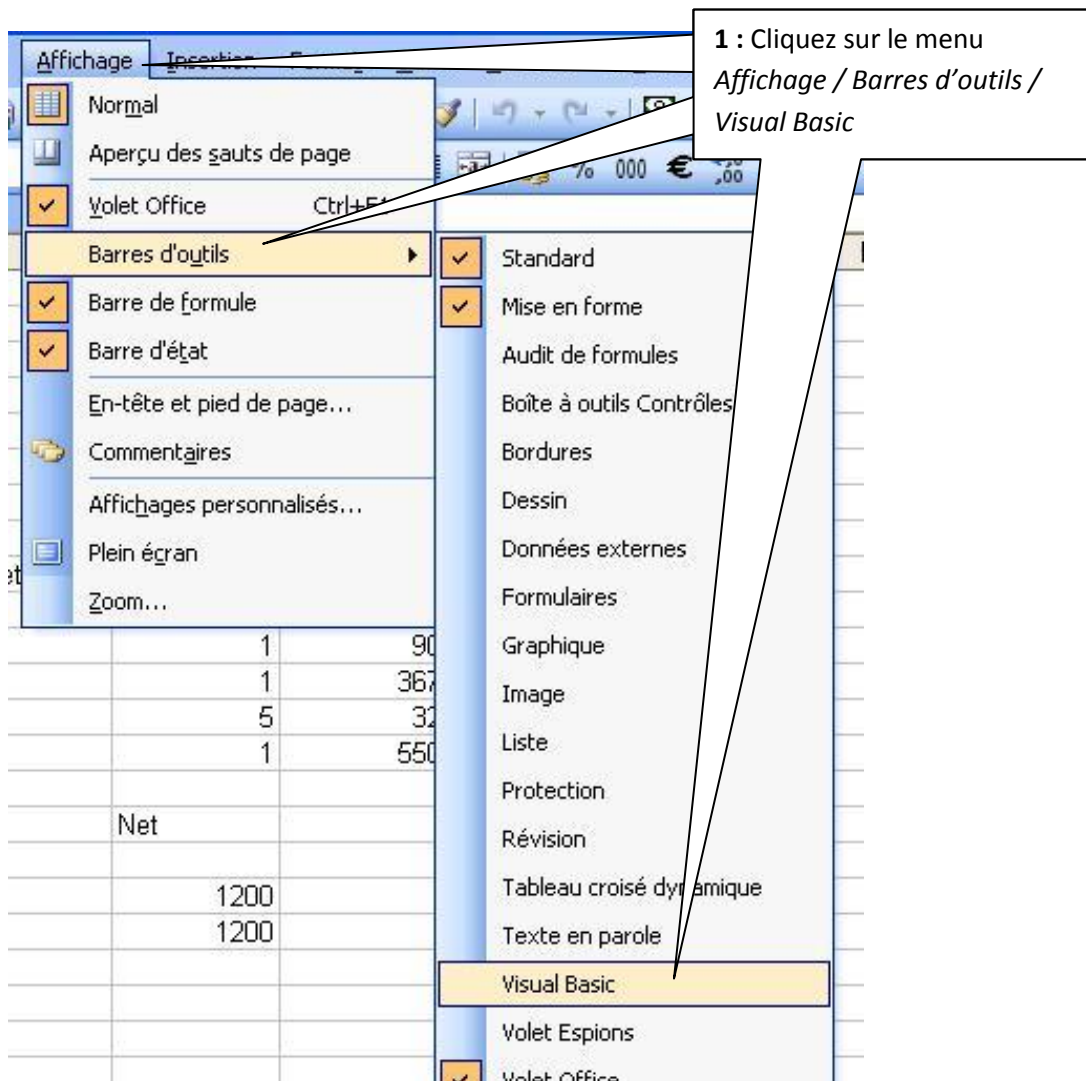
Vous allez utiliser des fonctions et des structures de programme que nous détaillerons dans le livret suivant. Dans ce livret vous faites une première approche des manipulations. Ne cherchez pas maintenant à savoir **pourquoi** on fait les choses mais attachez-vous à mémoriser **comment** on fait les choses. Les questions que vous vous poserez au fil des TP trouveront leurs réponses dans les cours suivants.

Chapitre 3 L'enregistreur de macro

3.1 Afficher la barre d'outils Visual Basic

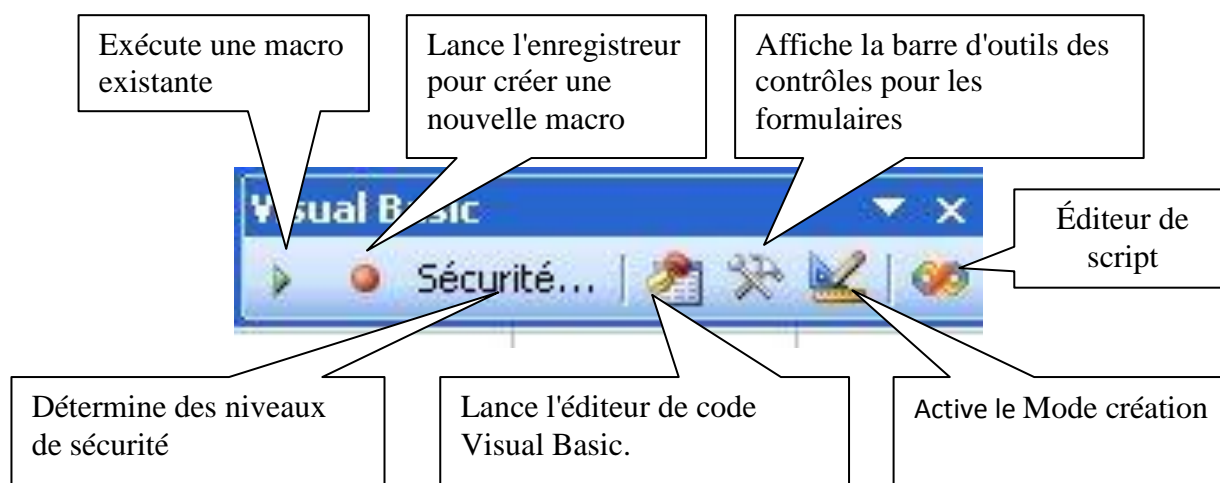
Nous allons apprendre à lancer une macro. Commençons par afficher la barre d'outils de création des macros.

Pour afficher la barre d'outils :

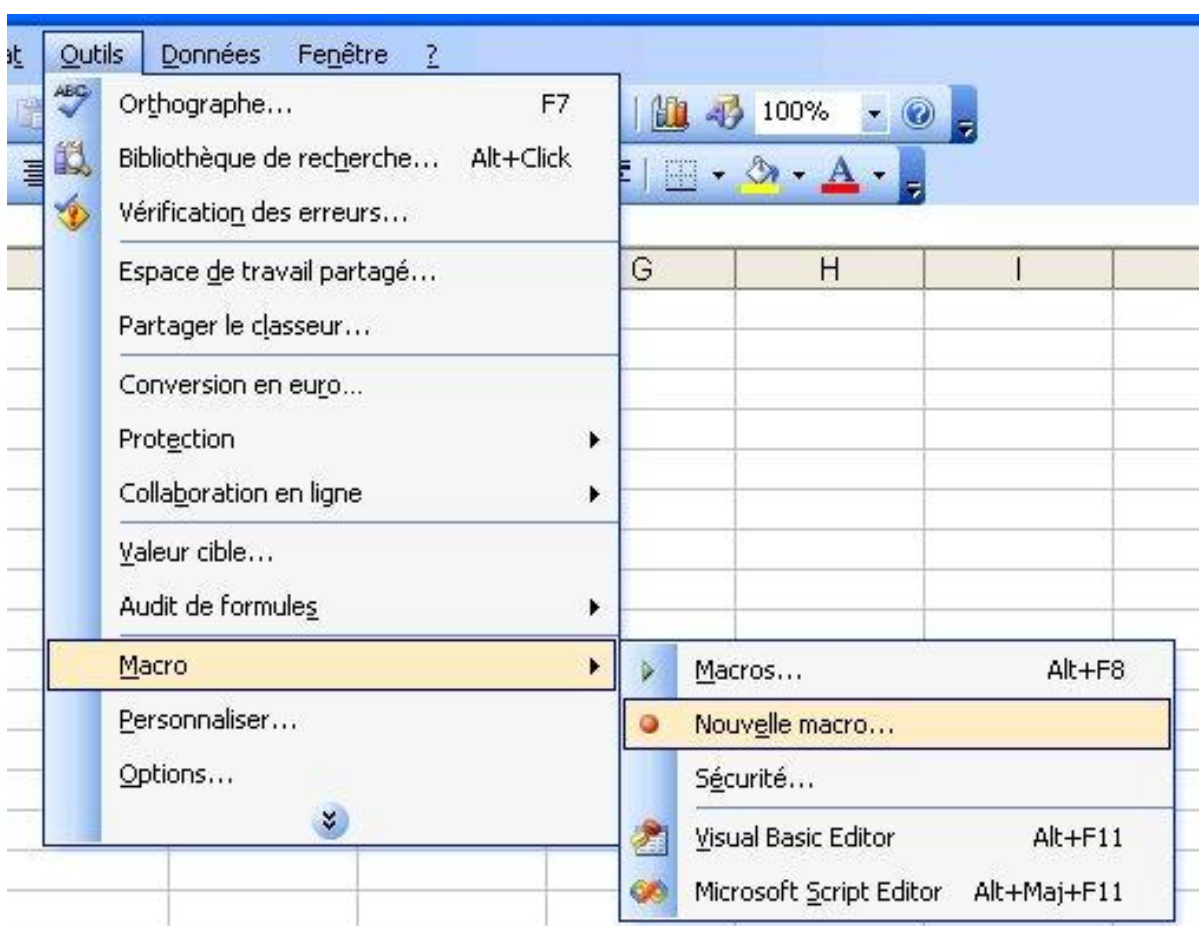


3.2 Description de la barre d'outils

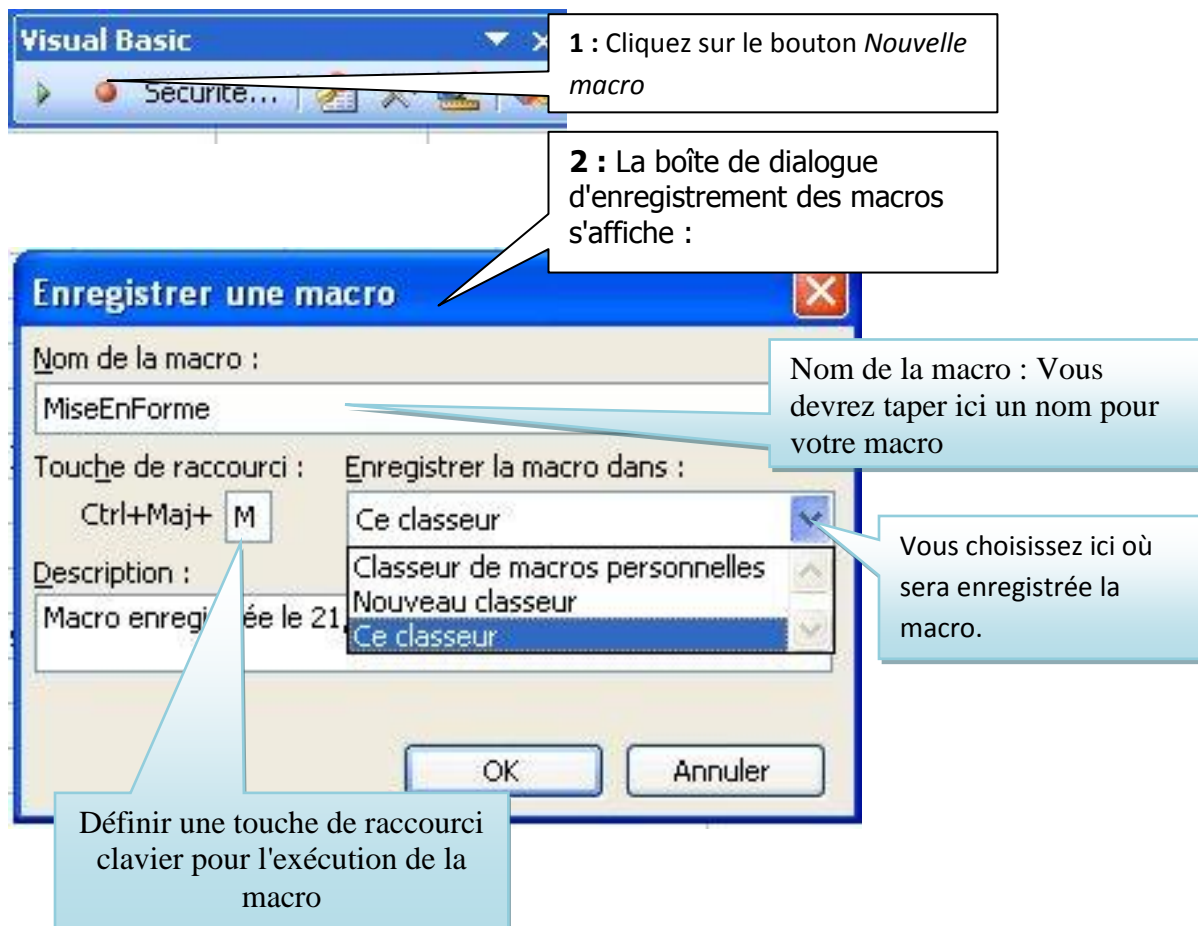
Voici un descriptif des différents boutons de la barre d'outils Visual Basic.



Autre méthode pour créer une nouvelle macro avec le menu : *Outils/Macro/Nouvelle macro*.



4.1 Créer une macro



1 Définition d'un raccourci

Vous pouvez désigner une combinaison de touches qui servira à l'exécution de la macro.

Ex C ou Shift+C. La touche Ctrl étant toujours active, il faudra appuyer sur Ctrl+R ou Ctrl+Shift+R pour exécuter la macro.

Attention : si vous choisissez de ne pas utiliser la touche Shift (facultative), sachez que des combinaisons comme Ctrl+C, Ctrl+G ... remplaceront les raccourcis d'Excel (ex :Ctrl+C = Copier).

2 Nom des macros

Le nom de la macro ne peut être composé que de lettres et/ou de chiffres. Le premier caractère doit être une lettre. Choisissez des noms explicites.

3 Affectation des macros

On détermine ici si la macro est affectée au classeur, à un nouveau classeur ou bien si la macro sera affectée à votre classeur Perso.xls qui contient les macros disponibles pour l'ensemble de vos applications Excel. Il est disponible dans le répertoire XLStart (c'est un fichier caché). Une fois créé, les macros seront accessibles dès l'ouverture d'Excel. Pour afficher ce fichier cliquez sur : Fenêtre / Afficher. Lorsque vous choisissez d'affecter les macros au classeur actif vous pourrez aussi les utiliser dans les autres classeurs mais à chaque fois ce classeur source s'ouvrira.

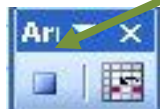
Lorsque vous cliquez sur le bouton *OK*, toutes vos actions clavier et souris seront enregistrées à la manière d'un magnétophone.



Avant d'enregistrer ou d'écrire une macro, planifiez les étapes et les commandes que la macro doit effectuer. Si vous commettez une erreur pendant l'enregistrement de la macro, ces erreurs sont également enregistrées.

4.2 Arrêter l'enregistrement d'une macro


Pour arrêter l'enregistrement de la macro vous cliquez sur le **bouton arrêt** de la barre d'outils *Arrêt enregistrement*.



Si cette barre d'outils n'est pas visible, vous pouvez l'afficher en faisant : *Affichage / Barre d'outils / Arrêter l'enregistrement*. (visible uniquement quand une macro est en cours d'enregistrement).

4.3 Exécuter une macro



Une fois la macro créée, cliquez sur  ou bien *Outils/Macro/Macros* pour l'exécuter.

Pour arrêter l'exécution de la macro avant la fin, cliquez sur Echap ou Ctrl + Pause.

Remarque :

Si vous souhaitez créer une macro en utilisant les références absolues (le fameux \$), cliquez sur le **boutons de références absolues**. Il reste actif jusqu'à ce que vous cliquiez à nouveau dessus.



Maintenant que vous avez appris comment lancer l'enregistrement d'une macro, nous allons faire quelques exercices pratiques.

5.1 Exo 1 : Formater une feuille

Nous allons créer une macro qui permet d'automatiser le centrage dans les cellules et d'agrandir la taille des cellules. La macro sera créée avec l'enregistreur de macro.

1. Ouvrez un classeur Excel
2. Créez une nouvelle macro. *Nom de la macro* : FormatCentré.
3. Sélectionner la feuille entière.
4. Cliquez sur Format cellule.

Onglet *alignement* : Horizontal et vertical = centré.

Onglet *police* : Choisissez la police de votre choix.

5. Augmentez légèrement la hauteur des lignes
6. Sélectionnez la cellule *A1*
7. Arrêter l'enregistrement de la macro

Chaque fois que vous exécuterez cette macro, le contenu de vos cellules sera centré verticalement et horizontalement. Grâce à ce type de macro, vous pouvez ainsi automatiser le formatage de vos feuilles Excel, surtout si elles comportent de nombreuses manipulations.

Examinons le code informatique.

FORMATION VBA EXCEL 2003

Livret 2 – L'éditeur Visual Basic

Niveau 2

Microsoft®

Visual Basic®
for Applications



Sommaire

Chapitre 1 INTRODUCTION.....	4
Chapitre 2 Présentation de l'éditeur	5
Chapitre 3 Les explorateurs.....	6
Chapitre 4 L'explorateur de projet.....	8
Chapitre 5 L'explorateur de propriétés	8
Chapitre 6 L'explorateur d'objets.....	9
Chapitre 7 Afficher l'aide.....	11
Chapitre 8 Le debug.....	12
8.1 Message de bug.....	12
Erreur de compilation.....	12
Erreur de paramètre.....	12
Erreur de syntaxe	12
Fenêtre de débogage.....	13
8.2 Barre d'outils de l'éditeur.....	13
8.3 La fenêtre d'exécution.....	14
8.4 Point d'arrêt (Break Point)	14
8.5 Fenêtre espion.....	15
8.6 Fenêtre des variables locales	17
8.7 Pas à pas détaillé	18
8.8 Pas à pas principal	18
Chapitre 9 Exercices	19

Chapitre 1 INTRODUCTION

Dans ce cours, vous allez apprendre à utiliser l'interface de l'éditeur de Visual Basic. Vous apprendrez les commandes servant à déboguer une macro, à rechercher de l'aide et découvrir les objets de VBA. Lisez simplement le contenu et essayer de bien vous repérer dans l'interface de l'éditeur de Visual Basic. Ce livret est une première approche, dans les cours suivants, vous aurez amplement l'occasion d'utiliser ce que vous apprendrez dans ce cours.

Pré-requis : Maitriser EXCEL et ses fonctions de base, livret 1 Introduction à VBA. Avoir fait le livret 18 Excel 2003 sur les macros

Objectifs :

- ☞ Se repérer dans l'éditeur Visual Basic
- ☞ Exécuter une macro pas à pas
- ☞ Utiliser les points d'arrêts

Signalétique



Attention



Information complémentaire

RECOMMANDATION : PRENEZ LE TEMPS DE BIEN LIRE LES INSTRUCTIONS.

FORMATION VBA EXCEL 2003

Livret 3 – Le langage VBA

Niveau 3

Microsoft®

Visual Basic®
for Applications



Sommaire

Chapitre 1 INTRODUCTION.....	5
Chapitre 2 Syntaxe de base	6
2.1 Différentes zones d'un module	6
2.2 Les commentaires	6
2.3 Les identificateurs	8
2.4 Les options.....	8
2.5 Les mots réservés	9
2.6 Les constantes	12
Les nombres réels.....	12
Les caractères et chaînes de caractères.....	12
Quelques constantes VB.....	13
2.7 Les types de base.....	15
2.8 Les variables	16
Les variables locales	17
Les variables statiques.....	17
2.9 Exercice.....	18
Exécuter la macro.....	19
Pas à Pas	19
Questions :.....	20
Chapitre 3 Les tableaux	22
3.1 Création de tableau	22
3.2 Naviguer dans un tableau.....	23
3.3 Exercice.....	24
Chapitre 4 Les instructions de contrôles	25
4.1 Généralités	25
4.2 If .. Then .. Else.....	25
4.3 Select Case.....	26
Chapitre 5 Les boucles.....	27
5.1 For .. To .. Next	27
5.2 For Each Next.....	28
Exercice.....	28
5.3 While .. Wend	29

5.4 Do While ... Loop	29
5.5 Do .. Loop While	30
5.6 Do Until ...Loop	30
5.7 Do . Loop Until.....	30
5.8 Exit.....	31
Chapitre 6 Instruction Enum	32
Chapitre 7 Les opérateurs	33
7.1 Opérateurs arithmétiques.....	33
7.2 Opérateurs logiques.....	33
7.3 Opérateurs de comparaison.....	34
7.4 Opérateurs de concaténation	34
7.5 Opérateur d'affectation d'objet SET	34
Chapitre 8 Les procédures.....	35
8.1 Les fonctions.....	35
Créer une fonction avec l'éditeur.....	38
8.2 Les arguments optionnels	40
8.3 Les propriétés	42
Chapitre 9 La gestion d'erreur.....	43
Chapitre 10 Notions d'objets	46
10.1 Description	46
10.2 Encapsulation	46
10.3 Les objets.....	46
10.4 Syntaxe des méthodes	48
Chapitre 11 Exercice	49
Chapitre 12 Corrigé	50

Chapitre 1 INTRODUCTION

Dans ce cours, vous allez apprendre les **instructions** nécessaires à la programmation Visual Basic. Les méthodes de programmation mentionnées dans ce livret sont celles utilisées dans la programmation de langages plus évolués comme le C++ ou le VB.NET. Bien que le VBA ne soit pas un langage orienté objet, nous nous efforcerons de nous en rapprocher. L'ensemble de ce livret vous servira de soutien pour les applications. Vous ne retiendrez pas tout du premier coup, aussi, n'hésitez pas à y revenir lorsque vous aborderez le livret suivant.

Pré-requis : Maitriser EXCEL et ses fonctions de base, livret 1 Introduction à VBA et Livret 2.

Objectifs :

- ☞ Maîtriser le langage Visual Basic
- ☞ Utiliser les instructions de contrôles
- ☞ Utiliser les boucles
- ☞ Appliquer des règles de programmation

Signalétique



Attention



Information complémentaire

RECOMMANDATION : PRENEZ LE TEMPS DE BIEN LIRE LES INSTRUCTIONS.

Beaucoup de mots nouveaux seront utilisés dès le début, ne vous focalisez pas trop dessus, ils seront largement expliqués dans le déroulement du cours, l'objectif est dans un premier temps de vous familiariser avec ces mots.

Chapitre 2 Syntaxe de base

2.1 Différentes zones d'un module

Un module est la page qui contient le code de programmation de nos macros. Les modules sont visibles avec l'éditeur de code de Visual Basic. Les différentes zones d'un module devraient se découper ainsi :

1. Entête (Présentation du module en commentaire)
2. Déclaration des options
3. Déclaration des constantes
4. Les procédures.

Chaque **procédure** commence par un mot clé : **Sub** et se termine par **End Sub**

Chaque **fonction** par : **Function** et se termine par **End Function**.

La différence entre une procédure et une fonction est qu'une fonction renvoie un résultat mais pas la procédure.

Ex :

Sub MettreEnRouge (Éventuellement des paramètres ici)

... Les instructions...

End Sub

Function SommeDe (Éventuellement des paramètres ici)

... Les instructions...

'je renvoi la somme trouvée'

End Function

2.2 Les commentaires

Voici deux méthodes pour écrire du texte qui sera considéré comme un commentaire et non comme du code Basic :

action ' commentaires (l'apostrophe désigne une zone de commentaire)

ou

REM commentaires

Les commentaires vous permettront de clarifier votre code. Notez toutefois que si votre code est bien fait et que les noms de vos variables et procédures sont bien choisis, vos commentaires seront peu nombreux.

Voici un exemple de ce que vous devrez faire :

```
Classeur1 - Module1 (Code)
(Général) MaTVA

'*****
'***** ENTETE *****
'Nom du Projet :
'Date de réalisation :
'Nom du développeur :
'Version du projet :
'Nom du module :
'remarques diverses :
'*****

'*****DECLARATION DES OPTIONS *****
'---
'*****

Function MaTVA(ByVal je_met_mes_paramètres_si_besoin)
'
' MaTVA Macro
' Macro enregistrée le 21/09/2009 par coursinfo.fr
'
'
'je met les instructions ici
End Function
```

La zone d'entête – facultative mais fort utile pour toutes modifications futures et suivies du développement

Cette zone pourra contenir des déclarations spécifiques

Déclaration d'une fonction (nous étudierons cela plus loin)

2.3 Les identificateurs

Les identificateurs sont des **noms** permettant de référencer les différents objets utilisés par le programme. Ces objets peuvent être des constantes, des variables ou des fonctions.

Pour suivre la logique Microsoft, chaque mot de l'identificateur commencera par une majuscule.

ex **Le**Volume **Calculer****La**Circonference()

Les **Procédures** étant des actions, il convient d'utiliser des verbes dans leur identificateur. Pour les fonctions, le nom de *l'objet* renvoyé (*en général c'est le résultat d'un calcul*) suffit comme identificateur.

Exemple pour une procédure : **Sub MettreEnGras(...)**

Exemple pour une fonction : **Function MaTVA(...)**

Le caractère '*espace*' est un **séparateur**, il ne peut donc **pas faire partie d'un identificateur**.

On peut aussi utiliser le caractère '_' (underscore) si on veut séparer les mots constituant un identificateur.

2.4 Les options

Le mot clé **Option** permet de configurer des options par défaut.

- **Option Base** : Cette option, par défaut à 0, permet de définir une base à un tableau.
- **Option Explicit** : Cette option oblige le *compilateur* (simplifions en disant *VBA*) à signaler des erreurs lorsque des variables ne sont pas déclarées.
- Déclarer une variables signifie faire connaître à VBA le nom et le type de la variable AVANT de l'utiliser.

Lorsqu'on choisi d'utiliser des options, celles-ci doivent être déclarée en tout premier dans le module.

2.5 Les mots réservés

Comme dans tout langage évolué VB possède un jeu de mots nécessaires à son fonctionnement.

- *Spécificateurs du type d'objet* : Integer, Char ...
- *Opérateurs symbolique* : =, And...
- *Instructions de contrôle* : if, do-while
- *Spécificateurs de classe* : static, extern

Le tableau suivant contient une liste des mots clés du langage Visual Basic. Tous les mots clés du langage sont réservés et ne peuvent pas servir d'identificateur.

Jetez-y un œil, les termes en gras sont ceux qui sont le plus utilisés.

Quelques mots clés Visual Basic		
As Le mot clé As introduit une clause As , qui identifie un type de données.	Call Appelle une fonction ou une procédure ou une procédure de bibliothèque de liaison dynamique (DLL, <i>Dynamic-Link Library</i>).	Date Les variables Date sont stockées sous la forme d'entiers IEEE 64 bits (8 octets) qui représentent des dates comprises entre le 1 ^{er} janvier de l'année 1 et le 31 décembre 9999 et des heures comprises entre 0:00:00 (minuit) et 23:59:59. Les valeurs Date doivent être placées entre des signes numériques (#) et leur format doit être de type d/m/yyyy, par exemple #31/5/1993#. Si vous convertissez une valeur Date en type String , la date est affichée en fonction du format de date courte reconnue par votre ordinateur, et l'heure est affichée en fonction du format d'heure (12 ou 24 heures) en vigueur sur votre ordinateur.
Dim Utilisée au niveau du module, de la classe, de la structure, de la procédure ou du bloc pour déclarer et allouer un espace de stockage	End Termine l'application en cours	Enum Permet de faire des énumérations (des listes de valeurs).

Quelques mots clés Visual Basic		
(en mémoire) pour les variables.		
<p>Exit</p> <p>Quitte une procédure ou un bloc et passe immédiatement le contrôle à l'instruction qui suit l'appel de procédure ou la définition de bloc.</p>	<p>False</p> <p>Valeur égale à 0. Utilisée pour les variables de types booléennes.</p>	<p>Get</p> <p>Déclare une procédure de propriété Get utilisée pour assigner une valeur à une propriété.</p>
<p>GetType</p> <p>Retourne le type de l'objet spécifié.</p>	<p>GoTo</p> <p>Effectue un branchement inconditionnel vers une ligne spécifiée d'une procédure. Utiliser dans la gestion d'erreur.</p>	<p>In</p> <p>Le mot clé In spécifie le groupe que doit traverser la variable de boucle dans une boucle For Each.</p>
<p>Is</p> <p>Instruction de test qui permet de comparer des objets.</p>	<p>Like</p> <p>Instruction de test qui permet de comparer des valeurs de types différents.</p>	<p>Me</p> <p>Le mot clé Me sert de variable objet faisant référence à l'instance en cours d'une classe. Lorsqu'une classe peut posséder plusieurs instances, Me offre la possibilité de faire référence à l'instance spécifique de la classe où s'exécute actuellement le code. L'utilisation de Me est particulièrement utile pour le passage des informations concernant l'instance d'une classe en cours d'exécution, à une procédure se trouvant dans une autre classe ou un autre module.</p>
<p>Nothing</p> <p>Le mot clé Nothing représente la valeur par défaut d'un type de données. L'assignation de Nothing à une variable la définit à sa valeur par défaut pour son type déclaré. Si ce type contient des membres de variable, ils ont tous leurs valeurs par défaut.</p>	<p>Optional</p> <p>Le mot clé Optional indique qu'un argument de procédure peut être omis lorsque la procédure est appelée.</p>	

Quelques mots clés Visual Basic

<p>Preserve</p> <p>Le mot clé Preserve empêche l'effacement du contenu d'un tableau lors du redimensionnement de ce dernier.</p>	<p>Private</p> <p>Le mot clé Private confère l'accès private sur un ou plusieurs éléments de programmation déclarés. Les éléments privés sont uniquement accessibles à partir de leur contexte de déclaration, y compris à partir de membres d'un type imbriqué, par exemple à partir d'une procédure imbriquée ou d'une expression d'assignation dans une énumération imbriquée.</p>	<p>Property</p> <p>Déclare le nom d'une propriété, ainsi que les procédures de propriété utilisées pour stocker et extraire la valeur de la propriété.</p>
<p>Protected</p>	<p>Public</p>	
<p>ReDim</p> <p>Utilisée au niveau de la procédure pour réallouer l'espace de stockage pour une variable tableau.</p>	<p>Resume</p> <p>Le mot clé Resume indique l'endroit où doit se poursuivre l'exécution après le traitement d'une erreur.</p>	<p>Set</p> <p>Déclare une procédure de propriété Set utilisée pour assigner une valeur à une propriété.</p>
<p>Stop</p> <p>Interrompt l'exécution en cours.</p>	<p>Variant</p> <p>Définit une variable dont la valeur peut-être de n'importe quel type et même changer de type.</p>	

FORMATION VBA EXCEL 2003

Livret 4 – VBA pour Excel

Niveau 3

Microsoft®

Visual Basic®
for Applications



Sommaire

Chapitre 1 INTRODUCTION.....	5
Chapitre 2 Les classes d'objets.....	6
2.1 Les collections.....	6
2.2 Les objets.....	7
Chapitre 3 VBA sur les cellules	10
3.1 Exercice 1 – Format de cellule et changement de valeur	10
3.2 Exercice 2 – Compter la récurrence d'une valeur	21
3.3 Exercice 3 – Table de multiplication.....	30
3.4 Exercice 4 – Tableau chronologique.....	31
Chapitre 4 VBA sur la feuille.....	35
4.1 Exercice 1 – Changer le nom d'une feuille	35
4.2 Exercice 2 – Photo employés.....	37
4.3 Exercice 3 – Contrôler l'insertion de doublons	43
4.4 Exercice 4 – Ajouter 12 feuilles mensuelles	46
Chapitre 5 VBA sur le classeur.....	49
5.1 Exercice 1 – Enregistrement de classeur.....	49
Chapitre 6 VBA sur les applications.....	52
6.1 Exercice 1 – Graphique Excel dans Word	52
Chapitre 7 Création de formulaires.....	54
7.1 Créer des boutons	54
7.2 Modifier la taille des boutons	59
7.3 Créer le code des boutons.....	61
7.4 Créer le code du classeur	62
7.5 Exercice.....	63
<i>Chapitre 8 Les formulaires UserForm</i>	<i>64</i>
8.1 Formulaire de saisie	64
8.2 Créer le formulaire UserForm	66
8.3 La boîte à outils	67
8.4 Modifier la taille du cadre formulaire	68
8.5 Titre du formulaire	69
8.6 Insérer des contrôles – les Labels (intitulés)	70
8.7 Aligner les contrôles	73

8.8 Insérer des contrôles – Les zone de texte	75
8.9 Insérer des contrôles – Liste déroulante (ComboBox)	76
8.10 Insérer des contrôles – Les cadres	77
8.11 Insérer des contrôles – les cases à cocher (CheckBox)	78
8.12 Insérer des contrôles – Les boutons.....	79
8.13 Préparation de la feuille de calcul	81
8.14 Code – Afficher le formulaire	81
8.15 Code VBA sur les contrôles.....	86
8.16 Initialisation du code des listes déroulantes (ComboBox)	89
8.17 Événement click sur la commande du bouton <i>Annuler</i>	90
8.18 Événement click sur la commande du bouton <i>Fermer</i>	90
8.19 Méthode d’insertion des données	91
8.20 Événement Click sur le bouton <i>Valider</i>	92
Chapitre 9 Annexe	94
9.1 Objet Sheet.....	94
9.2 Objet WorkBooks	97
9.3 Objet WorkSheets	99
9.4 Objet WorksheetFunction	101
9.5 Objet Range.....	102
9.6 Objet Cells.....	108
9.7 Les fonctions Texte.....	109
Chapitre 10 Corrigés.....	111
10.1 Exercice 3.1.....	111
10.2 Exercice 3.2 Analyser tableau.....	112
10.3 Exercice 4.2 – Photo Employés.....	113
10.4 Exercice 7.5 – Notes 3eme Aquitaine.....	116
Chapitre 11 Index	117

Chapitre 1 INTRODUCTION

Dans ce cours, vous allez apprendre les instructions nécessaires à la programmation Visual Basic. Les méthodes de programmation mentionnées dans ce livret sont celles utilisées dans la programmation de langages plus évolués comme le C++ ou le VB.NET. Bien que le VBA ne soit pas un langage orienté objet, nous nous efforcerons de nous en approcher.

Pré-requis : Maitriser EXCEL et ses fonctions de base, livret 1, 2 et 3 des cours VBA.

Objectifs :

- ☞ Créer des macros sur les cellules
- ☞ Créer des macros sur les feuilles et les classeurs
- ☞ Insérer des boutons de commandes
- ☞ Créer des formulaires
- ☞ Créer des fonctions
- ☞ Utiliser les objets Excel

Signalétique



Attention



Information complémentaire

Vous découvrirez VBA au travers de plusieurs TP. Ces TP vont aller croissant en difficultés.

Chapitre 2 Les classes d'objets

Avant de commencer, éclaircissons un point sur le fonctionnement du VBA et notamment ce qui concerne les objets et les collections.

2.1 Les collections

Une collection désigne un ensemble d'éléments ayant une relation entre eux. On accède à ces éléments grâce à leur numéro d'index ou leur nom. Les collections peuvent ainsi être hiérarchisées (l'application Excel contient une collection de classeur, chaque classeur contient une collection de feuilles, chaque feuille contient une collection de cellules...)

Les actions de base (appelées **méthodes**) de la plupart des collections, sont les méthodes suivantes :

Add	Pour ajouter un élément à la collection
Delete (ou Remove)	Pour supprimer un élément de la collections
Item	Pour accéder à un élément de la collection par son numéro d'index ou encore son nom.

La propriété **Count** permet de connaître le nombre d'éléments de la collection.

On devine ici que les collections prennent beaucoup de place en mémoire. Si on doit l'utiliser fréquemment, il est préférable de passer par une variable:

```
Dim MesCellules As Range  
Set MesCellules = ActiveSheet.Cells(1,1)
```

Sheets signifie *Feuilles*

Dans cet exemple, l'objet `MesCellules` prendra moins de place. En effet, `ActiveSheet` contient toutes les cellules de la feuille active et pas seulement celle qui nous intéresse (`Cells(1,1)`).

Un classeur se compose de différents éléments ou objets.

Les feuilles

Les graphiques

Les cellules

Les boîtes de dialogue

Les Formulaires

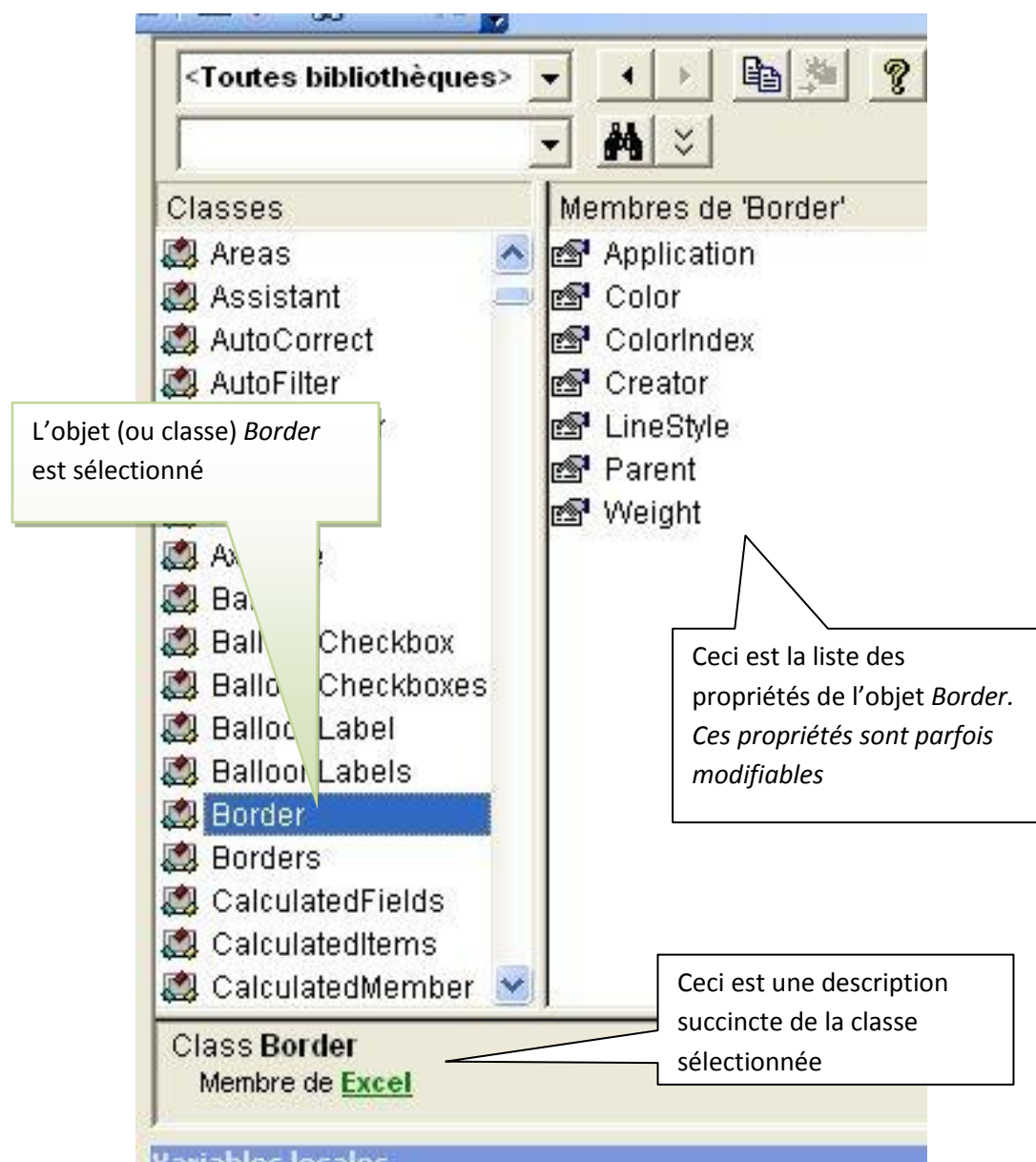
Un classeur peut également contenir des contrôles divers : image, bouton, liste déroulante... Ces contrôles sont aussi des objets qui ont des caractéristiques (propriétés) et sur lesquelles on peut agir.

2.2 Les objets

Les objets désignent des éléments spécifiques qui possèdent des propriétés et des méthodes. Par exemple, l'objet *Porte de la maison* possède des propriétés : dimension, type de bois, avec ou sans vitre... avec elle on peut réaliser des actions (*méthodes*) : ouvrir, fermer, verrouiller dégonder...

Il en va de même pour les objets Excel (qu'on appelle aussi des classes). On trouve l'objet *Cellules*, *feuilles*, *classeurs*...

Vous avez appris dans un livret précédent à afficher l'explorateur d'objet. Cet explorateur donne la liste des objets et des propriétés et méthodes.



La notation et l'utilisation des objets est à retenir également.

Dans les livrets précédents vous avez utilisé l'objet Range qui désigne une ou plusieurs cellules. Range("A1") correspond donc à la cellule A1.

Si on veut utiliser une méthode ou une propriété, nous taperons un point(.) après l'objet.

Voici un exemple pour désigner la valeur de la cellule A1 : **Range("A1").Value**

Les propriétés peuvent également s'enchaîner. Voici un exemple pour parler d'une cellule dont le contenu est mis en gras : **Range("A1").Font.Bold**

Font = Police d'écriture

Bold = gras (en anglais).

Vous devez absolument bien comprendre cette utilisation des objets. La suite du chapitre approfondira cet aspect.

