

Mikaël Bidault

Programmation Excel avec VBA

Compatible avec toutes
les versions d'Excel

EYROLLES



Programmation Excel avec VBA

MAÎTRISEZ TOUS LES ASPECTS DE LA PROGRAMMATION EXCEL

Qu'il s'agisse de faire face à un besoin immédiat ou de créer des programmes durables, cet ouvrage vous aidera à tirer pleinement profit d'Excel grâce à la programmation VBA.

Vous y apprendrez les principes de la programmation orientée objet, le langage VBA et découvrirez en détail Visual Basic Editor, l'environnement de programmation Excel.

Du simple enregistrement de macros à la conception d'interfaces utilisateur et au débogage de vos programmes, vous trouverez ici toutes les informations nécessaires au développement d'applications VBA pour Excel.

DÉVELOPPEZ UNE APPLICATION EXCEL PROFESSIONNELLE

Vous finirez l'ouvrage par la création d'un programme complet de génération de contrats et de feuilles de paie via une série d'interfaces utilisateur qui vous permettront de récapituler et mettre en œuvre toutes les connaissances acquises lors de votre lecture.

À qui s'adresse cet ouvrage ?

- Aux utilisateurs d'Excel désireux d'améliorer leur productivité
- Aux responsables qui souhaitent créer des solutions sûres et efficaces pour leurs équipes
- Aux personnes qui souhaitent s'initier à la programmation via le tableur de Microsoft

Des compléments web à télécharger

Tous les exemples de programmes du livre sont en téléchargement sur notre site Internet www.editions-eyrolles.com/dl/0067401.

Éditeur et développeur indépendant, **Mikael Bidault** développe des compléments Word et Excel pour des maisons d'édition et des sites Internet. Il est le créateur de l'articho, un complément VBA pour Word dédié à l'édition print et numérique (www.articho.eu).

Sommaire

Découvrir la programmation Excel. Notions fondamentales de la programmation orientée objet (POO) • Premières macros • Déplacement et sélection dans une macro Excel • Découvrir Visual Basic Editor • Programmer en Visual Basic. Développer dans Visual Basic Editor • Variables et constantes • Contrôler les programmes VBA • Fonctions Excel et VBA • Manipuler des chaînes de caractères • Déboguer et gérer les erreurs • Intégrer des applications VBA dans l'interface d'Excel • Développer des interfaces utilisateur. Créer des interfaces utilisateur • Exploiter les propriétés des contrôles • Maîtriser le comportement des contrôles • Notions avancées de la programmation Excel. Programmer des événements Excel • Protéger et authentifier des projets VBA • Exemple complet d'application Excel

Programmation Excel avec VBA

DANS LA MÊME COLLECTION

- K. NOVAK. – **Débuter avec LINUX.**
N°13793, 2017, 522 pages.
- P. MARTIN, J. PAULI, C. PIERRE DE GEYER. – **PHP 7 avancé.**
N°14357, 2016, 732 pages.
- L. BLOCH, C. WOLFHUGEL, A. KOKOS, G. BILLOIS, A. SOULLIÉ, T. DEBIZE. –
Sécurité informatique.
N°11849, 5^e édition, 2016, 648 pages.
- R. GOETTER. – **CSS 3 Flexbox.**
N°14363, 2016, 152 pages.
- W. MCKINNEY. – **Analyse de données en Python.**
N°14109, 2015, 488 pages.
- E. BIERNAT, M. LUTZ. – **Data science : fondamentaux et études de cas.**
N°14243, 2015, 312 pages.
- B. PHILIBERT. – **Bootstrap 3 : le framework 100 % web design.**
N°14132, 2015, 318 pages.
- C. CAMIN. – **Développer avec Symfony2.**
N°14131, 2015, 474 pages.
- S. PITTION, B. SIEBMAN. – **Applications mobiles avec Cordova et PhoneGap.**
N°14052, 2015, 184 pages.
- H. GIRAUDEL, R. GOETTER. – **CSS 3 : pratique du design web.**
N°14023, 2015, 372 pages.
- C. DELANNOY. – **Le guide complet du langage C.**
N°14012, 2014, 844 pages.
- K. AYARI. – **Scripting avancé avec Windows PowerShell.**
N°13788, 2013, 358 pages.
- W. BORIES, O. MIRIAL, S. PAPP. – **Déploiement et migration Windows 8.**
N°13645, 2013, 480 pages.
- W. BORIES, A. LAACHIR, D. THIBLEMONT, P. LAFEIL, F.-X. VITRANT. – **Virtualisation du poste de travail Windows 7 et 8 avec Windows Server 2012.**
N°13644, 2013, 218 pages.
- J.-M. DEFRANCE. – **jQuery-Ajax avec PHP.**
N°13720, 4^e édition, 2013, 488 pages.

SUR LE MÊME THÈME

- D.-J. DAVID. – **VBA pour Excel 2010, 2013 et 2016.**
N°14457, 2016, 324 pages.
- N. BARBARY. – **Excel expert.**
N°13692, 2^e édition, 2014, 444 pages.
- J.-M. LAGODA, F. ROSARD. – **Réaliser des graphiques avec Excel.**
N°56425, 2016, 128 pages.

Retrouvez nos bundles (livres papier + e-book) et livres numériques sur
<http://izibook.eyrolles.com>

Mikaël Bidault

Programmation Excel avec VBA

EYROLLES

The logo for EYROLLES features the word "EYROLLES" in a bold, sans-serif font. Below the text is a horizontal line with a small circle in the center, which is slightly thicker than the rest of the line.

ÉDITIONS EYROLLES
61, bd Saint-Germain
75240 Paris Cedex 05
www.editions-eyrolles.com

En application de la loi du 11 mars 1957, il est interdit de reproduire intégralement ou partiellement le présent ouvrage, sur quelque support que ce soit, sans l'autorisation de l'Éditeur ou du Centre Français d'exploitation du droit de copie, 20, rue des Grands Augustins, 75006 Paris.
© Groupe Eyrolles, 2017, ISBN: 978-2-212-67401-9

Table des matières

Introduction	1
Compléments VBA et compléments Office	2
VBA, pour quoi faire?	2
Des programmes	4
Une application hôte et des projets	4
Un langage de programmation	5
Un environnement de travail.....	7
Conventions typographiques.....	8
Codes sources des exemples du livre	9
PREMIÈRE PARTIE	
Découvrir la programmation Excel	11
CHAPITRE 1	
Notions fondamentales de la programmation orientée objet (POO)	13
Comprendre le concept d'objet	13
Objets et collections d'objets.....	14
Application hôte et modèles d'objets.....	15
Accéder aux objets	18
Les propriétés	20
Les méthodes	25
Les événements	26
Les fonctions	26
Le modèle d'objets d'Excel	27

CHAPITRE 2

Premières macros	31
Créer une macro GrasItalique	32
Afficher l'onglet Développeur	32
Démarrer l'enregistrement	33
Enregistrer les commandes de la macro	35
Exécuter la macro	35
Structure de la macro	36
Améliorer la macro	41
Une autre méthode d'enregistrement	43
Enregistrement	43
Structure de la macro	43
Écrire la macro	44
Exécution de la macro	45
Choisir l'accessibilité des macros	46
Accessibilité globale ou limitée	46
Classeurs et modèles	47
Le classeur de macros personnel	47
Les macros complémentaires	48
Définir le classeur de stockage lors de l'enregistrement d'une macro	50
Accéder aux macros d'un classeur spécifique	52

CHAPITRE 3

Déplacement et sélection dans une macro Excel	55
Méthodes de sélection dans une feuille Excel	56
Clavier	56
Souris	57
Notion de cellule active	57
Références relatives et références absolues	58
Coder les déplacements effectués lors de l'enregistrement d'une macro ...	59
Référence absolue aux cellules	60
Référence relative aux cellules	67
Référence aux cellules en fonction de leur contenu	69
Référence aux plages de cellules nommées	71

CHAPITRE 4

Découvrir Visual Basic Editor	73
Accéder à Visual Basic Editor	73
Les outils et les fenêtres de Visual Basic Editor	76
L'Explorateur de projet	77
L'Explorateur d'objets	80
La fenêtre UserForm	86
La fenêtre Code	89
La fenêtre Propriétés	101
Les barres d'outils	105
Paramétrer Visual Basic Editor	108

DEUXIÈME PARTIE

Programmer en Visual Basic	111
---	-----

CHAPITRE 5

Développer dans Visual Basic Editor	113
Structure des programmes Visual Basic	113
Les modules	113
Les procédures	114
Les instructions	116
Les différents types de procédures	118
Procédures Sub	118
Procédures Function	122
Procédures Property	124
Des projets bien structurés	129
Ajouter un module	129
Supprimer un module	130
Créer une procédure	132
Écrire l'instruction de déclaration	132
La boîte de dialogue Ajouter une procédure	133
La notion de portée	134
Écriture et mise en forme du code	135
Déplacer une procédure	140

Appel et sortie d'une procédure	140
Appel d'une procédure Sub	140
Appels de procédures Function et Property	142
Passage d'arguments	142
Sortie d'une procédure	144
Sortie d'un programme	145
Exécuter du code	146
Aide à l'écriture de code	146
Vérification automatique de la syntaxe	146
Complément automatique des instructions	147
Info express automatique	148
CHAPITRE 6	
Variables et constantes	149
Déclarer une variable	149
Déclaration implicite	149
Déclaration explicite	150
Types de données des variables	153
Chaînes de caractères	153
Valeurs numériques	155
Valeurs booléennes	157
Dates	157
Type Variant	158
Variables de matrice	159
Variables objets	163
Types de données personnalisés	167
Constantes	168
Validation et conversion des types de données	169
Portée et durée de vie des variables	172
Portée de niveau procédure	172
Portée de niveau module privée	172
Portée de niveau module publique	173
Variables statiques	173
Traitement entre applications à l'aide de variables objets	173

CHAPITRE 7

Contrôler les programmes VBA	177
Répéter une série d'instructions : les boucles	177
La boucle While...Wend	178
La boucle Do...Loop	181
La boucle For...Next	185
La boucle For Each...Next	189
Utiliser des instructions conditionnelles	192
La structure de contrôle If...Then...Else	193
La structure de contrôle Select Case	197
Définir l'instruction suivante avec GoTo	198
Interagir avec l'utilisateur via des boîtes de dialogue	198
La fonction InputBox	199
La méthode InputBox	202
La fonction MsgBox	204
Affichage de boîtes de dialogue Excel	209
Utiliser les opérateurs logiques	213
Trier des données	214

CHAPITRE 8

Fonctions Excel et VBA	217
Utiliser les fonctions Excel dans VBA	217
Créer des fonctions Excel personnalisées	218
Intégrer une fonction via l'Explorateur d'objets	219
Insérer une fonction VBA dans votre code	219
Insérer une fonction Excel dans votre code	220
Recommandations pour l'écriture de fonctions Excel	221
Les limites de la cellule	221
Principales fonctions VBA	222

CHAPITRE 9

Manipuler des chaînes de caractères	229
Modifier des chaînes de caractères	229
Concaténer des chaînes	229
Insérer des caractères non accessibles au clavier	231
Répéter une série de caractères	232
Supprimer les espaces superflus d'une chaîne	233
Extraire une partie d'une chaîne	233
Effectuer des remplacements au sein d'une chaîne	234
Modifier la casse des chaînes de caractères	235
Comparer des chaînes de caractères	236
Rechercher dans les chaînes de caractères	237
Rechercher une chaîne dans une chaîne	237
Scinder une chaîne	240
Rechercher une chaîne dans une variable de matrice	241

CHAPITRE 10

Déboguer et gérer les erreurs	245
Les étapes et les outils du débogage	245
Test du projet	246
Exécuter pas à pas	248
La fenêtre Variables locales	249
Les points d'arrêt	250
Modifier l'ordre d'exécution des instructions	251
La fenêtre Exécution	252
Les espions	252
La pile des appels	254
Exemple de débogage	255
Recherche du bogue	256
Résolution du bogue	258
Gestion des erreurs et des exceptions	261
Exemple de gestion d'erreur	262

CHAPITRE 11

Intégrer des applications VBA dans l'interface d'Excel	265
Affecter une macro à un raccourci clavier	265
Personnaliser le ruban et la barre d'outils Accès rapide	266
Affecter une macro à un bouton	268
Affecter une macro à un objet	268

TROISIÈME PARTIE

Développer des interfaces utilisateur	271
--	-----

CHAPITRE 12

Créer des interfaces utilisateur	273
Les phases de développement de feuilles	273
Créer une feuille	274
Les contrôles de la boîte à outils	276
Outil Sélection.....	276
Contrôle Label.....	276
Contrôle TextBox.....	277
Contrôle ComboBox.....	277
Contrôle Frame.....	278
Contrôle ListBox.....	278
Contrôle CheckBox.....	279
Contrôle OptionButton.....	280
Contrôle ToggleButton.....	280
Contrôle CommandButton.....	280
Contrôle TabStrip.....	281
Contrôle MultiPage.....	281
Contrôle ScrollBar.....	282
Contrôle SpinButton.....	283
Placer des contrôles sur une feuille	283
Copier-coller des contrôles.....	285
Sélectionner plusieurs contrôles.....	286
Supprimer des contrôles.....	287

Mise en forme des contrôles	287
La grille	287
Aligner les contrôles	289
Uniformiser la taille des contrôles	290
Uniformiser l'espace entre les contrôles	290
Centrer les contrôles	291
Réorganiser les boutons de commande	292
Grouper ou séparer des contrôles	292
Personnaliser la boîte à outils	293
Ajouter/supprimer un contrôle	293
Ajouter/supprimer une page	296
Afficher/masquer une feuille	298

CHAPITRE 13

Exploiter les propriétés des contrôles	301
Propriété Name	302
Apparence	303
Alignment	303
BackColor	304
Color	304
BorderStyle	305
BorderColor	305
Caption	305
ControlTipText	306
ForeColor	306
SpecialEffect	306
Style	307
Value	307
Visible	309
Comportement	311
AutoSize	311
AutoTab	312
AutoWordSelect	313
Cancel	313
Default	314
Enabled	314
EnterKeyBehavior	316

HideSelection	316
Locked.	317
MaxLength	317
MultiLine	317
SelectionMargin	318
Style.	318
TabKeyBehavior	319
TextAlign.	319
TripleState.	319
WordWrap.	320
Défilement	321
ScrollBars	321
KeepScrollsVisible	322
Delay	322
Max et Min	323
SmallChange	323
LargeChange	324
Divers	324
Accelerator	324
GroupName	325
HelpContextID	325
MouseIcon.	326
MousePointer	326
TabIndex	328
TabStop	329
Tag	329
Emplacement.	330
Height et Width	330
Left et Top.	330
StartUpPosition	330
Image	332
Picture	332
PictureAlignment	333
PicturePosition	334
PictureSizeMode.	334
PictureTiling	334
Police	335
Font	335

CHAPITRE 14

Maîtriser le comportement des contrôles	337
Créer des procédures événementielles	337
Créer une procédure	337
Les événements	343
Exemples d'exploitation des contrôles	348
Label	348
Contrôle TextBox	350
ComboBox	352
ListBox	357
CheckBox et OptionButton	360
ScrollBar	360
SpinButton	362
Exploiter les informations d'une feuille VBA	364

QUATRIÈME PARTIE

Notions avancées de la programmation Excel	367
---	-----

CHAPITRE 15

Programmer des événements Excel	369
L'objet Application	369
Déclaration et instanciation de l'objet Application	370
Création de procédures événementielles de niveau application	371
Propriétés de l'objet Application	372
Méthodes de l'objet Application	373
L'objet ThisWorkbook	374
L'objet Worksheet	376

CHAPITRE 16

Protéger et authentifier des projets VBA	379
Les virus macros	379
Se protéger des virus macros	379
Définir un niveau de sécurité	380
Les signatures numériques	382
Sauvegarder des macros	383
Protéger l'accès aux macros	384
Verrouiller un projet	384
limiter les droits d'exécution d'une macro.....	385
Authentifier ses macros	395
Obtenir une authentification	395
Authentifier une macro	396

CHAPITRE 17

Exemple complet d'application Excel	397
Présentation du projet d'application Excel	397
Avant de commencer.....	398
Identification des informations à recueillir.....	399
Définition de la structure du programme	400
Créer un modèle Excel	405
Définir et créer des interfaces	407
Feuille fmContratAuteur.....	407
Feuille fmContratConditions.....	418
Feuille fmContratDates.....	430
Feuille fmContratImpression	435
Feuille fmContratFin.....	440
Écriture des procédures d'édition de documents	441
Édition des feuilles de paie	442
Mise à jour du tableau Word.....	444

ANNEXE

Mots-clés pour la manipulation de fichiers et de dossiers	447
Index	449

Introduction

Visual Basic pour Applications, VBA, est la solution de programmation intégrée aux applications de la suite Office. La connaissance de VBA permet à l'utilisateur d'Excel de tirer pleinement profit du tableur de Microsoft en en développant les capacités et les fonctionnalités pour ses besoins spécifiques. Maîtriser VBA, c'est à coup sûr améliorer sa productivité.

L'intégration dans Excel de Visual Basic pour Applications, un *environnement de développement intégré* complet et professionnel, remonte à sa version 97. Office 2013 et Office 2016 intègrent la version 7.1 de Visual Basic, tandis qu'Office 2010 propose la version 2010 et que XP, 2003 et 2007 fournissent Visual Basic 6.3. Entre ces versions, les différences sont quasi-inexistantes.

Cet ouvrage traite de la programmation des versions 97 à 2016 d'Excel. Sauf exception signalée, les explications et les exemples proposés sont valides pour toutes les versions d'Excel. En effet, de l'une à l'autre, il n'y a pas eu de révolution. Le modèle d'objets s'est affiné et les nouvelles fonctions d'Excel, apparues au cours des différentes versions du logiciel, peuvent également être manipulées *via* la programmation VBA. Cependant, le langage, la gestion des programmes, l'environnement et les outils au service du développeur – bref, tout ce que vous devez savoir pour programmer Excel et que cet ouvrage se propose de vous apprendre – restent inchangés d'une version à l'autre.

Donc, sachez que vous pourrez appliquer les connaissances acquises lors de la lecture de ce livre, aussi bien avec Excel 2003 sous Windows XP qu'avec la version 2016 et un système Windows 10. Mieux, les programmes développés pour Excel 97 fonctionnent avec toutes les versions ultérieures du tableur et, dans la très grande majorité des cas, les programmes développés dans Excel 2016 devraient fonctionner avec les versions antérieures.

Dans cet ouvrage, vous découvrirez les différentes méthodes de création de projets VBA pour Excel, Visual Basic (le langage de programmation proprement dit) et les outils de développement et de gestion intégrés de Visual Basic pour Applications. Votre initiation à la programmation VBA se fera au moyen d'exemples de programmes détaillés et commentés.

Définition

Vous rencontrerez le terme « projet » tout au long de cet ouvrage. C'est ainsi que l'on nomme un ensemble de programmes développés avec Visual Basic pour Applications.

Compléments VBA et compléments Office

Avec Office 2013, Microsoft a introduit un nouveau type de compléments, les compléments Office. Contrairement à ceux développés en VBA, les compléments Office ne sont pas installés sur l'ordinateur de l'utilisateur, mais hébergés sur un serveur distant à partir duquel ils s'exécutent. Ils sont développés à partir de technologies Web, telles que HTML 5, JavaScript, CSS 3, XML et des API REST.

En termes d'expérience utilisateur, les compléments Office s'apparentent à des applications mobiles auxquelles on s'abonne *via* l'Office store et s'exécutent systématiquement dans un panneau qui leur est dédié. Cependant, si vous souhaitez développer des solutions professionnelles dans le cadre d'une entreprise et non dans le but de les commercialiser *via* l'Office store, VBA reste presque toujours la solution la plus simple et la plus souple à mettre en œuvre et à déployer.

VBA, pour quoi faire ?

Excel offre des possibilités très étendues. Pourtant, quelle que soit la puissance de ses fonctions, elles ne peuvent répondre à toutes les situations. La programmation VBA est la solution de personnalisation offerte par Excel, afin d'ajouter des caractéristiques, des fonctions et des commandes qui répondent précisément à vos besoins.

La programmation VBA peut être définie comme la *personnalisation d'un logiciel afin de s'assurer gain de temps, qualité des documents et simplification des tâches complexes ou fastidieuses*. Voici quelques exemples de ce que permettent les programmes VBA :

- Combiner un nombre indéterminé de commandes. Nous sommes souvent amenés à répéter ou à associer certaines commandes plutôt que d'autres et à ignorer certaines fonctionnalités selon l'usage personnel que nous avons d'un logiciel. VBA permet d'associer un nombre illimité de commandes à une seule. Vous pouvez ainsi ouvrir simultanément plusieurs documents Excel stockés dans des dossiers ou sur des serveurs différents, y insérer des données spécifiques et leur appliquer des mises en forme adaptées, en exécutant une seule commande créée en VBA.
- Ajouter de nouvelles commandes et de nouvelles fonctions à Excel – par exemple, une fonction personnalisée qui calcule les taxes à retenir sur un salaire (ou, mieux, les primes à y ajouter), etc. Vous pouvez, en outre, attacher vos programmes VBA à des raccourcis clavier et à des commandes d'onglets afin d'en améliorer l'accessibilité.

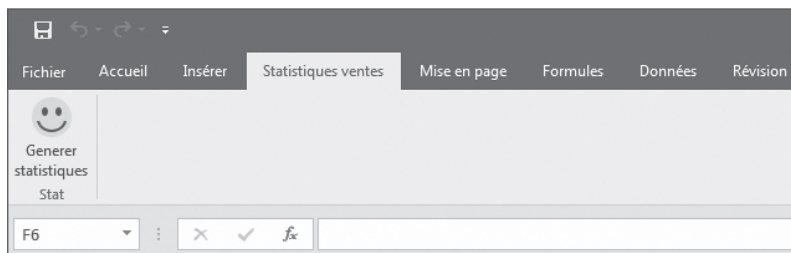


Figure 1 – Vous pouvez affecter les macros VBA à des commandes sur des onglets personnalisés.

- Automatiser des actions répétitives. Nous sommes parfois amenés à répéter certaines opérations plusieurs fois sur un même document ou à réitérer des traitements spécifiques. Un programme VBA peut, par exemple, mettre en forme des cellules dans un classeur Excel, effectuer des séries de calculs, etc.
- Modifier et améliorer les commandes d'une application. Les commandes Excel ne sont pas toujours adaptées à nos besoins ou présentent parfois des limitations gênantes. Un programme VBA peut modifier, brider ou compléter les commandes d'une application. Vous pouvez ainsi intégrer dans un tableau le nom de l'utilisateur, le nombre de pages imprimées et l'imprimante utilisée chaque fois qu'une impression est lancée à partir d'Excel.
- Faire interagir les différentes applications Office. Un programme VBA sait exploiter des données issues de fichiers générés par d'autres programmes et interagir avec ceux-ci de façon transparente pour l'utilisateur. Vous pouvez ainsi créer une commande qui envoie automatiquement le classeur Excel ouvert en fichier joint dans un courriel Outlook à des destinataires définis ou qui génère un rapport Word à partir de données Excel et l'imprime.
- Créer des interfaces personnalisées. Les programmes VBA peuvent ramener des tâches complexes à la simple information de champs dans des boîtes de dialogue personnalisées pour l'utilisateur final, simplifiant ainsi considérablement le travail de celui-ci, tout en vous assurant qu'aucun oubli ou fausse manipulation n'aura lieu.

Visual Basic pour Applications permet le développement de solutions adaptées à vos besoins. Les outils que vous apprendrez à manier vous permettront de développer des programmes simples, sans écrire la moindre ligne de code, comme des programmes complets intégrant une interface utilisateur adaptée.

La fonction d'un programme VBA peut être d'automatiser une tâche répétitive. Cependant, vous pouvez aussi créer très vite un petit programme VBA pour faire face à une nécessité immédiate ; par exemple, afin de généraliser un traitement exceptionnel à l'ensemble d'un document.

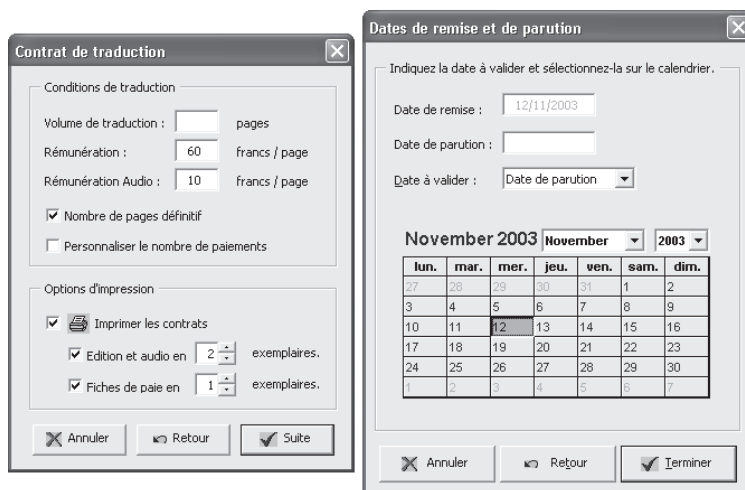


Figure 2 – Visual Basic pour Applications vous permet de développer des interfaces utilisateur évoluées.

Des programmes

Les projets *VBA* sont des programmes ou *macros* écrits dans le langage Visual Basic. Si vous ne possédez aucune expérience préalable en programmation, ne vous inquiétez pas : cet ouvrage aborde le développement de projets VBA à travers l'enregistrement de macros. Lorsque vous l'activez, l'Enregistreur de macro mémorise chacune de vos actions. C'est votre programmeur personnel : vous utilisez simplement les commandes d'Excel et il se charge de traduire les actions exécutées en instructions Visual Basic. Il vous suffit ensuite d'exécuter la macro pour répéter l'ensemble des commandes enregistrées.

Définition

Le terme macro désigne le regroupement d'un ensemble de commandes en une seule. On parle parfois de macrocommande pour désigner un programme qui se résume à l'exécution d'une série de commandes, sans égard pour le contexte. Des macros plus évoluées peuvent répéter des opérations en boucle ou afficher des boîtes de dialogue qui autorisent une interaction avec l'utilisateur. Ces programmes se comporteront différemment en fonction des informations entrées ou de l'état du document sur lequel elles s'exécutent.

Le terme projet est plus large. Il désigne l'ensemble des éléments constituant vos programmes VBA. Il s'agit toujours de macros, mais à celles-ci peuvent s'ajouter des feuilles – qui constituent une interface utilisateur permettant de récolter des informations de tout type –, des modules de classe et autres friandises que vous découvrirez tout au long de cet ouvrage.

L'enregistrement de macros constitue sans aucun doute le meilleur moyen de se familiariser avec la programmation en Visual Basic. Ainsi, sans connaître le langage – les instructions qui le composent et la façon dont elles sont structurées –, vous pouvez créer des programmes VBA et en visualiser ensuite le code.

Une application hôte et des projets

Visual Basic pour Applications est un environnement de développement calqué sur Visual Basic, une solution de développement d'applications Windows. Les structures de contrôle du langage sont les mêmes, et l'environnement proprement dit (Visual Basic Editor) est pour ainsi dire identique à celui de Visual Basic. Cependant, contrairement à Visual Basic, Visual Basic pour Applications est conçu... *pour des applications*. Cela signifie que, tandis que les programmes Visual Basic sont autonomes, les programmes VBA ne peuvent être exécutés qu'à partir d'une application intégrant cet environnement de développement – Excel ou une autre application.

Lorsque vous développez un programme VBA, vous l'attachez à une application. Il s'agit de l'*application hôte* du programme. Plus précisément, vos programmes VBA sont attachés à un document (un fichier ou un modèle Word, une feuille de calcul Excel, une présentation PowerPoint...) spécifique à l'application hôte. L'ensemble des programmes VBA attachés à un document constitue un projet. Un projet regroupe des macros, mais peut également intégrer des interfaces utilisateur, des déclarations système, etc. Un projet constitue en fait la partie VBA d'un document. Si cet ouvrage ne traite que de la programmation pour Excel, sachez qu'un programme VBA peut être attaché

à une autre application. Les concepts et les outils que vous découvrirez au long de cet ouvrage sont valides pour toutes les applications de la suite Office. Pour exécuter une macro VBA, vous devez avoir accès au document auquel elle est attachée. Vous pouvez choisir de rendre certaines macros disponibles à partir de n'importe quel document Excel ou en limiter l'accessibilité à un classeur Excel spécifique. La disponibilité des programmes VBA est abordée au chapitre 2.

Un langage de programmation

Les projets VBA sont développés dans le langage de programmation Visual Basic. Vous découvrirez par la pratique la structure de ce langage et apprendrez rapidement à en discerner les composants et les relations qu'ils entretiennent. Comme nous l'avons dit précédemment, l'enregistrement de macros constitue une excellente initiation à Visual Basic. C'est sous cet angle que nous vous ferons découvrir ce langage.

Visual Basic est un langage de *programmation orientée objet* (POO). Nous présenterons donc les concepts de ce type de programmation. Vous apprendrez ce qu'on appelle un objet, une propriété, une méthode ou un module de classe. Vous verrez comment conjuguer ces éléments pour créer des applications Excel souples et puissantes. Visual Basic pour Applications constitue une bonne approche de la programmation pour le néophyte.

VBA intègre un grand nombre d'instructions, grâce auxquelles vous développerez des macros qui identifient très précisément l'état de l'application et des documents et reproduisent l'exécution de la plupart des commandes disponibles dans l'application hôte.

Vous verrez que certaines instructions sont spécifiques à Excel, par exemple celles qui affectent une formule à une cellule. Vous n'utiliserez probablement qu'un nombre limité de ces instructions, en fonction de votre usage personnel d'Excel ou des besoins de votre entreprise. Cependant, certaines apparaîtront presque toujours dans vos macros. C'est par exemple le cas de la propriété *Range*, qui renvoie un objet Excel tel qu'une cellule ou une plage de cellules.

D'autres instructions sont communes à l'ensemble des applications Office, notamment celles qui règlent le comportement d'une macro : réaliser des opérations en boucle, induire des réactions face à certains paramètres, afficher des boîtes de dialogue simples (figures 3 et 4) ou développer des interfaces utilisateur évoluées (figure 1), etc. Ce sont ces instructions qui constituent véritablement ce qu'il est convenu d'appeler *le langage Visual Basic*. Vous aurez besoin d'y faire appel dès que vous voudrez créer un programme interactif, capable de se comporter différemment selon le contexte. Pour la plupart, ces instructions ne peuvent être générées par enregistrement de macros et doivent donc être éditées manuellement dans Visual Basic Éditeur.

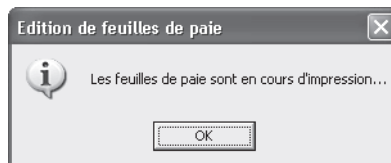


Figure 3 – La fonction VBA MsgBox affiche une boîte de dialogue.

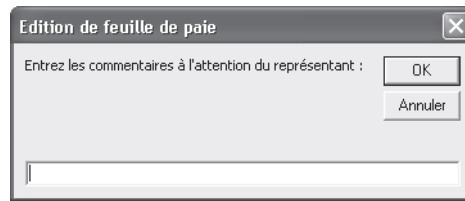


Figure 4 – Il existe une version VBA et une version Excel de la fonction *InputBox*.

Cet ouvrage ne se veut pas un dictionnaire du langage, mais un guide qui vous enseignera le développement de projets VBA de qualité. Vous apprendrez à enregistrer, modifier, exécuter et déboguer des macros, à créer des interfaces utilisateur ainsi qu'à gérer vos projets VBA. Vous découvrirez, à travers les exemples de cet ouvrage, un certain nombre d'instructions spécifiques à la *hiérarchie d'objets* d'Excel, qui vous familiariseront avec la logique de ce langage.

Définition

La hiérarchie d'objets d'une application, encore appelée modèle d'objets, est le rapport qu'entretiennent entre eux les différents objets d'une application. Ce concept ainsi que les notions spécifiques aux langages orientés objet seront développés au chapitre 1, « Notions fondamentales de la programmation orientée objet ».

Ce livre présente et illustre d'exemples commentés l'ensemble des structures de contrôle qui servent à créer très simplement des macros évoluées. Nous vous fournirons les bases du langage Visual Basic. Elles suffisent pour créer une infinité de macros et répondre à vos besoins spécifiques.

Lorsque les principes du développement de projets VBA vous seront acquis et que vous créerez vos propres macros, il vous arrivera sûrement d'avoir besoin d'instructions que vous n'aurez pas rencontrées lors de la lecture de cet ouvrage ; vous pourrez alors utiliser l'Enregistreur de macro ou encore les rechercher dans l'aide de Visual Basic pour Applications ou dans l'Explorateur d'objets – étudié au chapitre 4. Vous verrez que l'aide de VBA fournit une référence complète du langage, facilement accessible et consultable.

Si vous n'avez aucune expérience de programmation, peut-être ce *Visual Basic* vous apparaît-il comme un langage barbare ou inaccessible. Ne vous inquiétez pas : le développement de projets VBA ne requiert ni expérience préalable de la programmation, ni connaissance globale du langage. Contentez-vous, au cours de votre lecture, d'utiliser les fonctions nécessaires aux exercices et que nous vous détaillerons. Cet ouvrage propose un apprentissage progressif et concret : vous développerez vos premiers projets VBA dès les premiers chapitres.

Un environnement de travail

VBA dispose d'un environnement de développement à part entière : Visual Basic Editor.

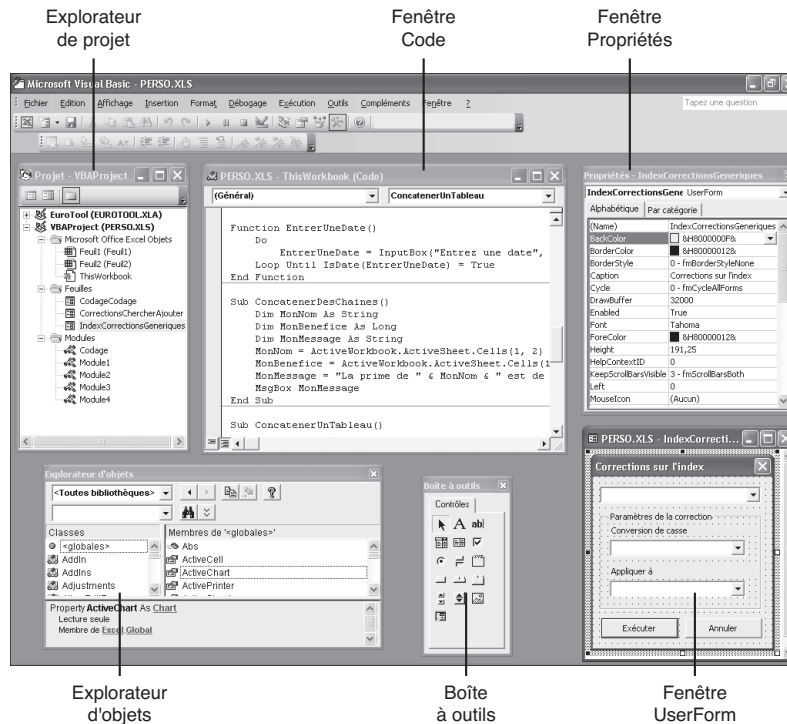


Figure 5 – Visual Basic Editor est l'environnement de développement de Visual Basic pour Applications.

Visual Basic Editor est l'environnement de développement intégré des applications Office. Il permet de visualiser et de gérer les projets VBA, d'écrire, de modifier et de déboguer les macros existantes, de visualiser comment les commandes propres à une application Office sont traduites en langage Visual Basic et inversement. C'est un outil de débogage de vos projets VBA d'une grande efficacité. Il propose nombre d'outils pour tester les macros et en étudier le comportement. Vous pouvez ainsi exécuter les commandes de la macro pas à pas, en suivre le déroulement, insérer des commentaires dans le texte de la macro, etc. Enfin, cet environnement intègre des outils très intuitifs, dédiés au développement d'interfaces graphiques.

Vous apprendrez dans cet ouvrage à utiliser les nombreux outils de Visual Basic Editor à toutes les phases de développement d'un projet VBA.