

Apprenez le langage VBA

et devenez un expert sur Excel

Jean-Emmanuel CHAPARTEGUI

Fichiers complémentaires
à télécharger



Solutions Business



Chapitre 1

Introduction

A. Introduction	9
1. Pourquoi apprendre VBA ?	9
2. Pourquoi ce livre ?	10
3. Quel est le niveau nécessaire pour lire ce livre ?	11
4. Quel sera votre niveau à la fin de la lecture de ce livre ?	11
5. Comment lire ce livre ?	11
6. L'auteur	12
B. Avant de commencer	13
1. Vocabulaire spécifique	13
2. Raccourcis-clavier	14
a. Manipulation d'un classeur	14
b. Manipulation du tableur	14
3. Versions de Microsoft Office Excel	15
a. Excel 2003	15
b. Excel 2007	16
c. Excel 2010	16
d. Excel 2013	16
e. Excel 2016	16
f. Office pour Mac	17
g. OpenOffice.org	17
h. Version du livre	17
i. Langue du produit Office	17

Chapitre 2

Gestion des employés : exploitation de données brutes

A. Formules Excel avancées	21
1. Description de l'exemple	21
a. Présentation de l'exemple	21
b. Présentation du classeur	22
c. Fonctionnalités	24
2. Notions de cours	25
a. Liste déroulante dans une cellule – Validation de données	25
b. Formule de recherche	29
c. Structure conditionnelle sur Excel : conditions et SI	32
d. Formule conditionnelle	32

e.	Gestion des cas d'erreur	34
f.	Calcul matriciel	35
3.	Réalisation de l'exemple	36
a.	Nommage des plages	36
b.	Fonctions de recherche : afficher le poste et le salaire de l'année précédente	38
c.	Gestion des erreurs et formules conditionnelles	42
d.	Gérer l'erreur sur le salaire des nouveaux arrivants	43
e.	Salaire moyen par grade et filière : calcul matriciel contre formule conditionnelle	44
f.	Création de la pyramide des âges	49
B.	Indicateurs clés et partages	57
1.	Description de l'exemple	57
a.	Présentation de l'exemple	57
b.	Présentation du classeur	57
c.	Fonctionnalités	57
2.	Notions de cours	57
a.	Formules Excel	57
b.	Création d'un graphique Sparkline	60
c.	Mise en forme conditionnelle simple	61
d.	Introduction au tableau	67
3.	Réalisation de l'exemple	68
a.	Mise en place du tableau	68
b.	Formules statistiques	69
c.	Mise en valeur des données	71
d.	Sparkline contre graphique classique	73

Chapitre 3

Gestion des ventes et formulaires VBA

A.	Formulaire de gestion des ventes	79
1.	Description de l'exemple	79
a.	Présentation de l'exemple	79
b.	Présentation du fichier	80
c.	Fonctionnalités	81
2.	Notions de cours	82
a.	Concept de programmation	82
b.	Concept de formulaire	84
c.	Rédaction du code	86

d. Le fonctionnement de l'éditeur Visual Basic	90
3. Réalisation de l'exemple	92
a. Création du formulaire	92
b. Création des contrôles sur le formulaire	96
c. Définition des procédures et événements	101
d. Rédaction du code : procédures et événements	103
B. Protéger le classeur	132
1. Description de l'exemple	132
a. Présentation de l'exemple	132
b. Présentation du fichier	133
c. Fonctionnalités	133
2. Notions de cours	133
a. Afficher/masquer une feuille	133
b. Protéger la structure	135
c. Protéger la feuille et ses cellules	137
d. Protéger le code VBA	140
3. Création de l'exemple	142
a. Masquer les feuilles Factures et Produits	142
b. Protéger la structure du classeur	143
c. Afficher les stocks via un formulaire	143
d. Protéger les cellules de la feuille Accueil	149
e. Protéger le code VBA	152

Chapitre 4

Gestion d'une campagne de test

A. Création de tableaux et graphiques croisés dynamiques (TCD et GCD)	155
1. Description de l'exemple	155
a. Présentation de l'exemple	155
b. Présentation du fichier	156
c. Fonctionnalités	159
2. Notions de cours	159
a. Créer un tableau croisé dynamique simple	159
b. Créer un tableau croisé dynamique avec l'assistant	162
c. Champs calculés et éléments calculés	169
d. Créer un graphique croisé dynamique	172
3. Réalisation de l'exemple	174
a. Stock d'anomalies	174
b. Nombre d'anomalies par projets (et par priorité)	178

c.	Avancement des cas de tests	184
d.	Revue des cycles de test.....	189
e.	Indicateur de situation des tests	194
B.	Automatisation de la création d'un rapport PowerPoint	200
1.	Description de l'exemple	200
a.	Présentation de l'exemple.....	200
b.	Présentation du fichier	201
c.	Fonctionnalités.....	202
2.	Notions de cours	203
a.	Enregistrement de macro	203
b.	Créer un tableau croisé dynamique avec VBA	205
c.	Créer un graphique avec VBA.....	207
d.	Manipuler PowerPoint	208
3.	Réalisation de l'exemple	210
a.	Actualiser et copier les graphiques.....	210
b.	Nombre de tests par personne.....	214
c.	Mise en forme du rapport.....	218
d.	Création du rapport PowerPoint	223
e.	Finalisation	227

Chapitre 5

Gestion des employés

A.	Calcul de la durée et du planning.....	233
1.	Description de l'exemple.....	233
a.	Présentation de l'exemple.....	233
b.	Présentation du fichier	233
c.	Fonctionnalités.....	235
2.	Notions de cours	235
a.	Formules de date	235
b.	Mise en forme conditionnelle avancée	237
3.	Réalisation de l'exemple	240
a.	Calcul de la durée de chaque tâche	240
b.	Mise en forme du diagramme de Gantt	249
B.	Gestion des présences - Outil d'administration.....	254
1.	Description de l'exemple	254
a.	Présentation de l'exemple.....	254
b.	Présentation du fichier	254
c.	Fonctionnalités.....	257

2. Notions de cours	258
a. Création dynamique de contrôle	258
b. Tableaux VBA	259
3. Réalisation de l'exemple	260
a. Initialisation du formulaire	260
b. Bloquer l'accès à la feuille Planning	270
c. Calculer le coût du projet	271

Chapitre 6

Consolidation et partage de données

A. Consolidation de données diverses	283
1. Description de l'exemple	283
a. Présentation de l'exemple	283
b. Présentation des classeurs	284
c. Fonctionnalités	286
2. Notions de cours	287
a. Manipulation de feuilles et classeurs	287
b. Sélection et ouverture d'un classeur Excel	287
c. Les boucles	288
d. Format de la cellule	289
e. Formule Excel dans le code VBA	289
f. Select Case et structure conditionnelle	290
3. Réalisation de l'exemple	291
a. Structure du code	291
b. Déclaration des variables feuille et classeur	292
c. Définition de la boîte de dialogue d'ouverture de fichier	293
d. Parcours des feuilles	294
B. Partage des données	309
1. Description de l'exemple	309
a. Présentation de l'exemple	309
b. Présentation des classeurs et outils utilisés	309
c. Fonctionnalités	310
2. Notions de cours	311
a. Formulaire de tableau	311
b. OneDrive	316
c. Enquêtes	317
d. Envoyer un e-mail avec VBA via Outlook	319

6

Apprenez le langage VBA

et devenez un expert sur Excel

3. Réalisation de l'exemple	322
a. Créer un formulaire de saisie automatique pour faciliter la saisie des données	322
b. Créer une enquête partagée via OneDrive et la diffuser	326
c. Envoyer un e-mail avec les statistiques des ventes aux agences	335
Index	343

A. Formulaire de gestion des ventes

1. Description de l'exemple

a. Présentation de l'exemple

Nous sommes l'entreprise **SacEni**, un distributeur qui commercialise des sacs de sports, un sac en taille L, un autre en taille XL. L'activité débute et l'entreprise souhaite se doter d'un simple fichier pour suivre ses ventes et son stock.

L'outil que vous allez mettre en place va permettre au vendeur de cette petite entreprise de créer, stocker et télécharger des factures.

L'outil va se présenter comme un formulaire à remplir par le vendeur. Il sera accessible à partir d'un fichier Excel et du bouton **Accéder à l'outil de gestion des ventes** qui sera positionné sur la feuille d'ouverture du classeur.

Voici l'outil une fois qu'il sera finalisé :

The screenshot shows a window titled "Logiciel de gestion des ventes" with a close button (X) in the top right corner. The interface is divided into several sections:

- Input Fields:** A "Quantité" field containing the number "5" and a "Produit" dropdown menu currently set to "Sac taille XL". A dropdown menu is open below the "Produit" field, showing options "Sac taille L" and "Sac taille XL".
- Action Button:** A button labeled "Ajouter à la ligne d" is positioned below the input fields.
- Item List:** A list box contains two entries: "4*Sac taille L=80€" and "5*Sac taille XL=150€".
- Management Button:** A button labeled "Supprimer la ou les lignes sélectionnées" is located below the list box.
- Summary Section:**
 - "Total HT" is 230 €, with a checkbox for "Remise 10 % pro" which is currently unchecked.
 - "Total après remise" is 230 €, "TVA" is 46 €, and "Total TTC" is 276 €.
- Final Button:** A button labeled "Sauvegarder et imprimer la commande" is at the bottom of the window.

b. Présentation du fichier

Pour réaliser cet exemple, vous allez utiliser le fichier Excel **Enoncé_3-A.xlsm**. Le format XLSM signifie que c'est un fichier Excel qui prend en charge les macros (permettant l'utilisation de VBA, souvent désactivée par défaut).

Ce fichier contient trois feuilles Excel (appelées *sheets* en Visual Basic) :

Feuille Accueil

Cette feuille est destinée à contenir uniquement le bouton **Accéder à l'outil de gestion des ventes**. Il permettra l'accès à l'outil de gestion des ventes.

Initialement cette feuille est vide.

Feuille Produits

La feuille **Produits** recense les produits vendus par la société. Elle contient trois colonnes :

	Colonne A	Colonne B	Colonne C
Ligne 1	Nom du produit	Prix hors taxe	Quantité disponible
Ligne 2	Sac taille L	20 €	90
Ligne 3	Sac taille XL	30 €	50

Feuille Factures

Cette feuille contiendra les factures créées avec l'outil de gestion des ventes. Elles seront référencées par numéro et vous trouverez également la date/heure de l'édition, ainsi que le montant.

	Colonne A	Colonne B	Colonne C
Ligne 1	Numéro de facture	Date et heure	Montant de la facture

c. Fonctionnalités

L'objectif est donc d'avoir un outil qui permet de gérer les ventes des sacs de l'entreprise. Voici la retranscription des besoins sous forme d'exigences.

Exigences métiers

Les exigences métiers correspondent à la description des fonctionnalités globales de l'application, c'est le niveau de détail le plus faible :

- ▶ Créer une facture ;
- ▶ Tracer la facture ;
- ▶ Mettre à jour les stocks de produits.

Retranscription de ces exigences métiers en fonctionnalités

Les fonctionnalités permettent de détailler les exigences métiers en fonctionnalités qui correspondent aux actions utilisateurs et aux traitements du système. Cette liste doit être exhaustive afin de permettre de réaliser ces fonctionnalités sous forme d'application.

- ▶ Ajouter une ligne de commande
 - ▶ Choisir la quantité ;
 - ▶ Choisir le produit ;
 - ▶ Valider la ligne de commande ;
- ▶ Afficher la commande.
- ▶ Supprimer une ligne de commande.

- ▶ Faire le total et permettre l'application d'une remise de 10 %.
- ▶ Mettre à jour les stocks.

2. Notions de cours

Cet exemple contient de nombreux nouveaux concepts liés à la programmation avec le langage Visual Basic... Quelques repères sont donc utiles pour pouvoir démarrer sereinement.

a. Concept de programmation

Voici une description simplifiée de quelques notions de base de programmation permettant une meilleure approche des exemples proposés.

Objet et classe

Un **objet** est une entité informatique, il peut être de toute forme et chaque objet est unique. Il est caractérisé selon son type.

La **classe** correspond à la définition de l'objet, elle servira de canevas pour la création de nouveaux objets. Par conséquent, tous les objets d'une même classe auront les mêmes propriétés, ils se différencieront par les valeurs de leurs propriétés.

Exemple :

Classe	Objets
Cell (cellule d'une feuille Excel)	<ul style="list-style-type: none"> ▶ Cells("A1") : Cellule A1 de la feuille en cours ; ▶ Cells("C4") : Cellule C4 de la feuille en cours.
Sheet (Feuille de classeur)	<ul style="list-style-type: none"> ▶ Sheets(0) : première feuille du classeur en cours ; ▶ Sheets(1) : deuxième feuille du classeur en cours.
Textbox (zone de texte saisissable dans un formulaire)	<ul style="list-style-type: none"> ▶ Textbox1 : zone de texte saisissable nommée Textbox1 par l'utilisateur ▶ Textbox2 : zone de texte saisissable nommée Textbox2 par l'utilisateur.

Propriétés

Une **propriété** correspond à un attribut d'une classe. Lorsqu'un objet est créé, il a donc des valeurs assignées à ses propriétés.

Exemple : La cellule d'une feuille comporte de nombreuses propriétés, comme par exemple la valeur : `Cells("A1").value`

Méthode

Une **méthode** correspond à une action qui peut être réalisée par un objet. Par exemple, l'objet feuille de calcul (`Sheets`) propose une méthode `Add` qui permet d'ajouter une feuille.

Exemple : `Sheets.Add`

Collections

Une **collection** est une liste d'objets d'une même classe. Par exemple, la collection `Sheets` correspond à l'ensemble des feuilles. En Visual Basic, les collections sont des objets à part entière avec leurs propres méthodes et propriétés.

Variables

Une **variable** est une entité informatique qui permet de stocker des informations au sein de l'application, elle se déclare de la manière suivante :

- ▶ `Dim` : permet de définir la variable (`Public` pour une variable publique) ;
- ▶ `Nom_variable` : permet de donner un nom à la variable ;
- ▶ `As TypeVariable` : permet de typer la variable.

Exemple :

```
Dim MaVariable As String
```

Cela signifie que la variable `MaVariable` est déclarée en tant que chaîne de caractères.

Les variables sont :

- ▶ **Publiques** : elles sont accessibles sur l'ensemble de l'application. Elles sont déclarées en dehors de toute procédure de code ;
- ▶ **Privées** : elles sont accessibles uniquement dans la procédure où elles sont déclarées (sur une procédure donnée).

Les variables sont typées principalement pour les trois motifs suivants :

- ▶ Cela permet d'avoir des méthodes (voir précédemment) adaptées à la variable : une addition de chaînes de caractères correspond à la concaténation alors qu'une addition de nombres correspond à la somme des valeurs :

Opérations	Valeur de la variable chaîne de caractères	Valeur de la variable nombre entier
<code>MaVar = "A" + "E"</code>	"AE"	Erreur
<code>MaVar = 1 + 2</code>	"12"	3

- ▶ Cela facilite le développement et l'usage de variable, le contenu de la variable est attendu.
- ▶ Chaque type de variable a une quantité de mémoire allouée, par conséquent, utiliser le bon type de variable permet d'économiser de la mémoire.

Voici les types de variable et le détail de chacune :

Nom	Type	Détails
Byte	Numérique	Nombre entier de 0 à 255
Integer	Numérique	Nombre entier de -32'768 à 32'767
Long	Numérique	Nombre entier de - 2'147'483'648 à 2'147'483'647
Currency	Numérique	Nombre à décimale fixe de - 922'337'203'685'477.5808 à 922'337'203'685'477.5807
Single	Numérique	Nombre à virgule flottante de -3.402823E38 à 3.402823E38
Double	Numérique	Nombre à virgule flottante de -1.79769313486232D308 à 1.79769313486232D308
String	Texte	Texte
Date	Date	Date et heure
Boolean	Boolean	True (vrai) ou False (faux)
Object	Objet	Objet Microsoft (exemple cellule, plage de cellule, feuille)
Variant	Tous	Valeur par défaut si non déclarée

b. Concept de formulaire

Formulaire

Un **formulaire** (dit *form*) est une fenêtre d'interaction entre l'utilisateur et le système. Il s'agit d'une interface visuelle permettant de restituer et/ou collecter de l'information dans le but de faire des traitements.

L'objet formulaire est le contenant des autres objets visuels. Cela signifie qu'il comporte d'autres objets visuels, ceux-là même proposés dans la boîte à outils.

Une même application peut contenir plusieurs formulaires.

Les contrôles

Un formulaire est un contenant de contrôles. Ces **contrôles** sont des objets visuels qui permettent l'interaction avec l'utilisateur.