

Chapitre 4

Les objets d'Excel

1. L'objet Application

L'objet **Application** représente l'application Microsoft Excel active. Cet objet étant l'objet par défaut, il est souvent facultatif (exemple : *Version* équivaut à *Application.Version*).

Cet objet contient les éléments suivants :

- Des **propriétés relatives à l'environnement Excel** (options d'Excel...) et à la **présentation de l'interface** (pointeur de la souris, texte de la barre d'état, état et taille de la fenêtre de l'application...).
- Différentes **méthodes** permettant d'effectuer des actions dans l'environnement Excel.
- Des **propriétés** renvoyant les objets et collections de premier niveau (objets et collections du modèle objet Excel : Workbooks, Charts...).
- Des **propriétés spécifiques** faisant directement référence à des objets : ActiveCell, ActiveSheet, ActiveWindow...

■ Remarque

Les propriétés faisant référence aux objets sont détaillées dans le chapitre La programmation objet sous Excel.

122 — VBA Excel (2021 et Microsoft 365)

Programmer sous Excel : macros et langage VBA

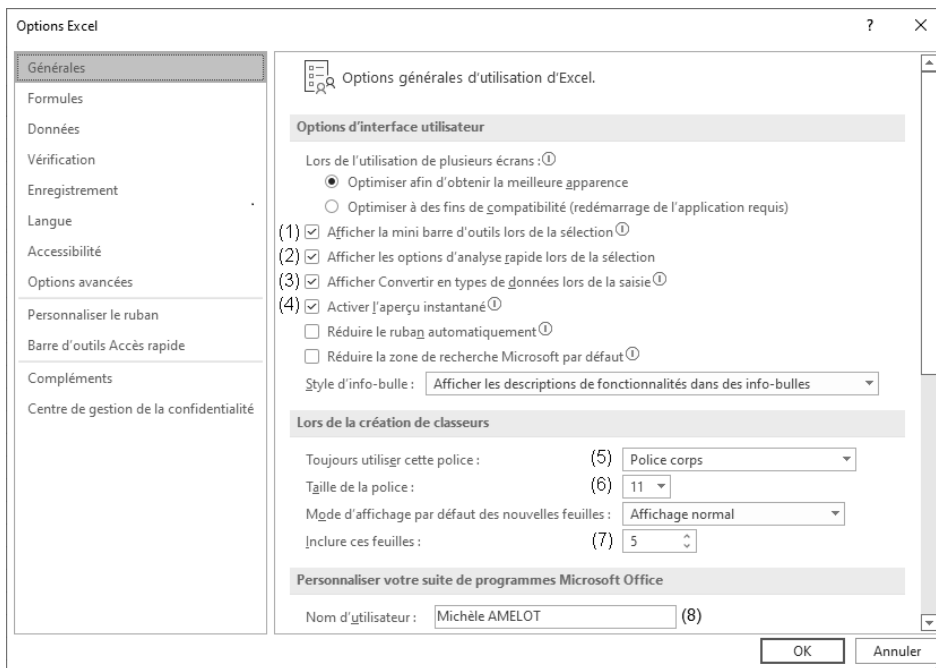
1.1 Propriétés représentant les options d'Excel

Les principales options d'Excel peuvent être renvoyées ou définies à partir de propriétés de l'objet **Application**. La plupart de ces propriétés sont en lecture et écriture.

Remarque

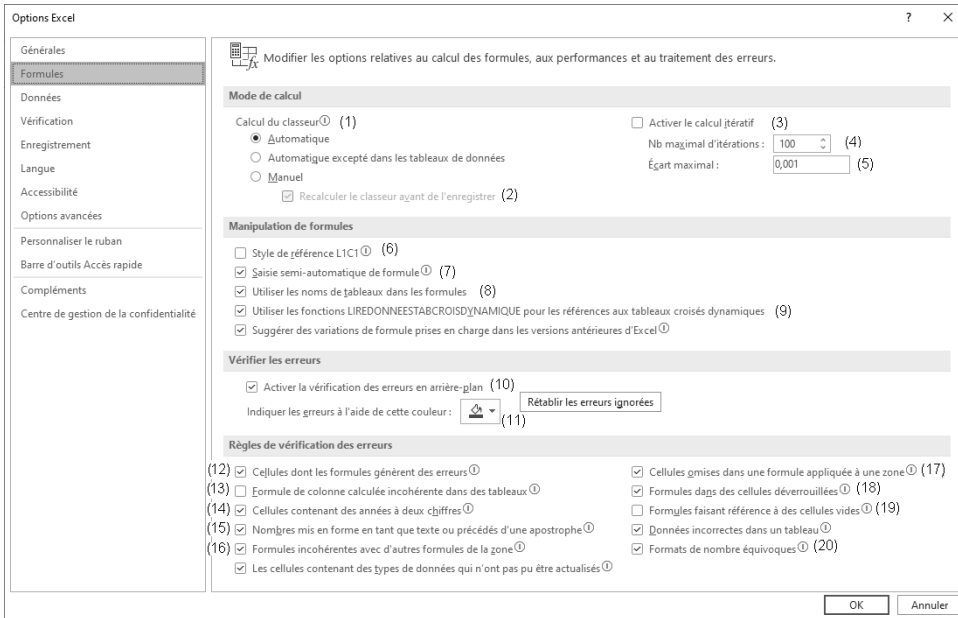
*Pour accéder aux options Excel, cliquez sur l'onglet **Fichier**, puis sur la commande **Options**.*

1.1.1 Options de la catégorie Standard



N°	Propriétés	Valeurs retournées
1	ShowSelectionFloaties	Booléen
2	ShowQuickAnalysis	Booléen
3	ShowConvertToDataType	Booléen
4	EnableLivePreview	Booléen
5	StandardFont	Entier long
6	StandardFontSize	Entier long
7	SheetsInNewWorkbook	Entier long
8	UserName	Chaîne de caractères

1.1.2 Options de la catégorie Formules



124 — VBA Excel (2021 et Microsoft 365)

Programmer sous Excel : macros et langage VBA

Options relatives au Mode de calcul

N°	Propriétés	Valeurs retournées
1	Calculation	Constantes : xlCalculationAutomatic xlCalculationManual xlCalculationSemiautomatic
2	CalculateBeforeSave	Booléen
3	Iteration	Booléen
4	MaxIterations	Entier long
5	MaxChange	Double

Options relatives aux Formules

N°	Propriétés	Valeurs retournées
6	ReferenceStyle	Constantes : xlA1 xlR1C1
7	DisplayFormulaAutoComplete	Booléen
8	GenerateTableRefs	Constantes : xlGenerateTableRefStruct xlGenerateTableRefA1
9	GenerateGetPivotData	Booléen

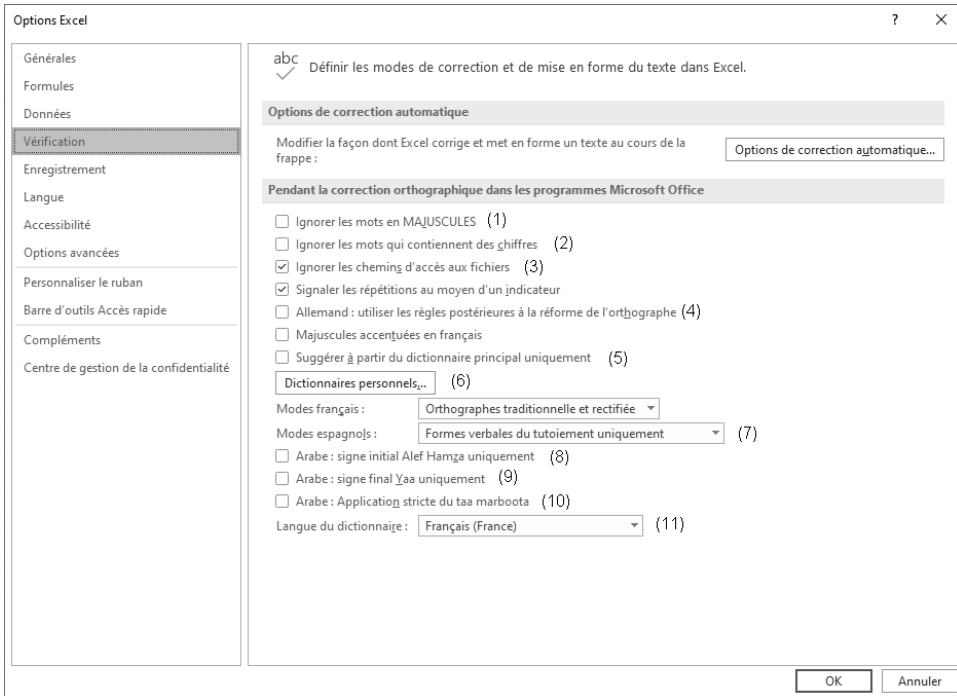
Options de vérification d'erreurs

Les propriétés suivantes dépendent de la propriété **ErrorCheckingOptions** de l'objet **Application**. Cette propriété renvoie un objet **ErrorCheckingOptions**, qui représente les options de vérification d'erreurs pour une application.

N°	Propriétés	Valeurs retournées
10	BackgroundChecking	Booléen
11	IndicatorColorIndex	Constante XIColorIndex

N°	Propriétés	Valeurs retournées
12	EvaluateToError	Booléen
13	InconsistentTableFormula	Booléen
14	TextDate	Booléen
15	NumberAsText	Booléen
16	InconsistentFormula	Booléen
17	OmittedCells	Booléen
18	UnlockedFormulaCells	Booléen
19	EmptyCellReferences	Booléen
20	ListDataValidation	Booléen

1.1.3 Options de la catégorie Vérification



126 — VBA Excel (2021 et Microsoft 365)

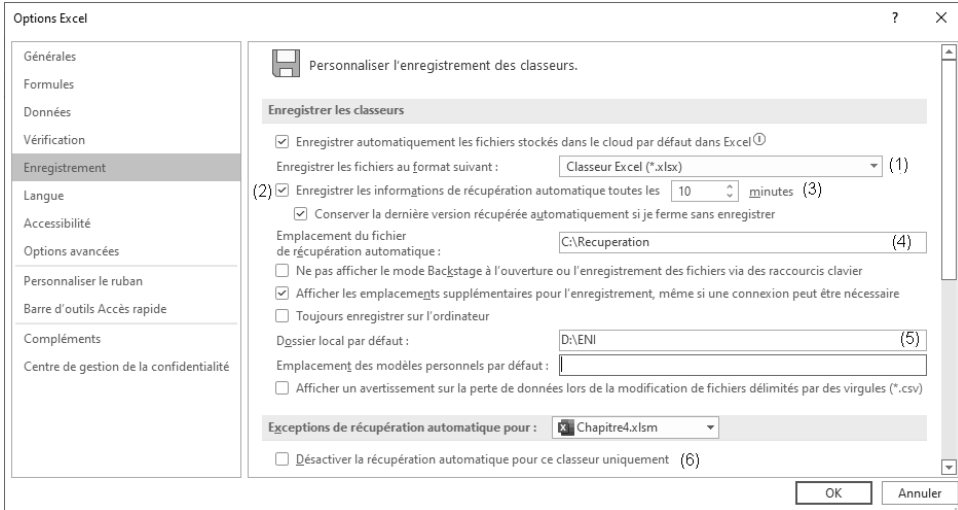
Programmer sous Excel : macros et langage VBA

Options de vérification orthographique

Les propriétés suivantes dépendent de la propriété **SpellingOptions** de l'objet **Application**. Cette propriété renvoie un objet **SpellingOptions**, qui représente les options de vérification orthographique pour une application.

N°	Propriétés	Valeurs retournées
1	IgnoreCaps	Booléen
2	IgnoreMixedDigits	Booléen
3	IgnoreFileNames	Booléen
4	GermanPostReform	Booléen
5	SuggestMainOnly	Booléen
6	UserDict	String
7	SpanishModes	Constante xlSpanishModes
8	ArabicStrictAlefHamza	Booléen
9	ArabicStrictFinalYaa	Booléen
10	ArabicStrictTaaMarboota	Booléen
11	DictLang	Entier long

1.1.4 Options de la catégorie Enregistrement



N°	Propriétés	Valeurs retournées
1	DefaultSaveFormat	Constante xlFileFormat
2	AutoRecover.Enabled	Booléen
3	AutoRecover.Time	Entier long
4	AutoRecover.Path	Chaîne de caractères
5	DefaultFilePath	Chaîne de caractères
6	EnableAutoRecover	Booléen

■ Remarque

L'option 6 s'applique à un classeur (Application.ActiveWorkbook par exemple).

Chapitre 4

Structures de contrôle

Durée : 1 heure 55

Mots-clés

condition, choix, alternative, branchement conditionnel, compteur, itération, incrémenter, décrémenter, sortie, boucle, test

Objectifs

Maîtriser les structures de décision afin de tester des conditions puis effectuer des actions différentes selon le résultat obtenu. Maîtriser les instructions d'itération qui, associées aux instructions conditionnelles, permettent d'écrire du code Visual Basic pour la prise de décision et la répétition des actions. Vous retrouverez ces structures dans la suite du livre. Dans ce chapitre, nous nous limitons à l'emploi des boîtes de dialogue déjà rencontrées.

Prérequis

Pour valider les prérequis nécessaires, avant d'aborder le TP, répondez aux questions ci-après (certaines questions peuvent nécessiter plusieurs réponses) :

1. Les structures suivantes sont des structures de décision :
 - a. If ... Then ... Else ... End If
 - b. Do ... Loop
 - c. Select Case ... Case ... End Select
2. Résultat = IIf(7 / 2 > 3, IIf(2.8 * 3.3 < 11, "X", "Y"), "Z")
La variable Résultat contient la valeur :
 - a. X
 - b. Y
 - c. Z

3. Les instructions suivantes appartenant chacune à une structure de contrôle différente sont correctes :
 - a. Case If $N1 > N2$
 - b. Case A, B, C
 - c. Case 1 to 10
 - d. Case Nombre, Is > 50
4. Le mot-clé `ElseIf` :
 - a. peut apparaître de suite après une clause `Else`.
 - b. est facultatif.
 - c. peut être utilisé plusieurs fois dans un bloc `If`.
5. Répétition d'instructions tant qu'une condition a la valeur `True` :
 - a. `For Each ... Next`
 - b. `For ... Next`
 - c. `Do ... Loop`
 - d. `While ... Wend`
 - e. `With ... End With`
6. Utilisation d'un compteur pour exécuter des instructions un certain nombre de fois :
 - a. `For Each ... Next`
 - b. `For ... Next`
 - c. `Do ... Loop`
 - d. `While ... Wend`
 - e. `With ... End With`
7. Répéter un groupe d'instructions pour chaque élément d'un tableau ou d'une collection :
 - a. `For Each ... Next`
 - b. `For ... Next`
 - c. `Do ... Loop`
 - d. `While ... Wend`
 - e. `With ... End With`

8. Répéter un groupe d'instructions le nombre de fois indiqué :
 - a. For Each ... Next
 - b. For ... Next
 - c. Do ... Loop
 - d. While ... Wend
 - e. With ... End With
9. Exécuter une série d'instructions appliquées à un seul objet ou à un type défini par l'utilisateur :
 - a. For Each ... Next
 - b. For ... Next
 - c. Do ... Loop
 - d. While ... Wend
 - e. With ... End With

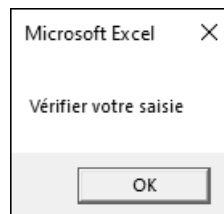
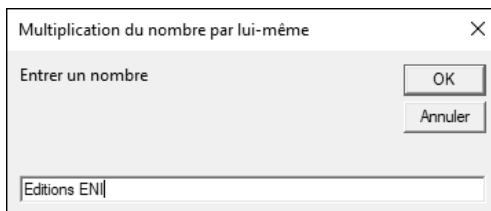
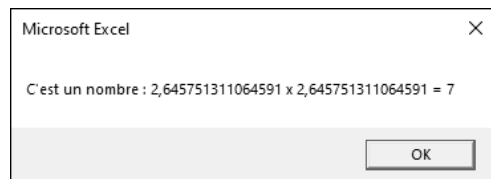
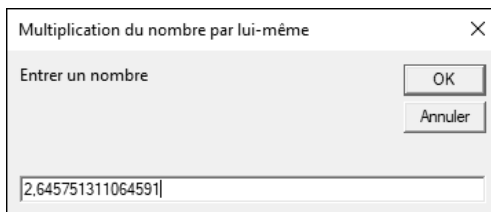
Corrigé p. 271

Énoncé 4.1 Vérifier que la saisie est numérique

Exercice 1

Durée estimative : 10 minutes

Créez la procédure **ContrôleSaisie** qui propose une boîte de saisie et vérifiez qu'il s'agit d'un nombre. Si c'est le cas, effectuez l'opération qui consiste à multiplier le nombre par lui-même, sinon affichez un message. Exemple :



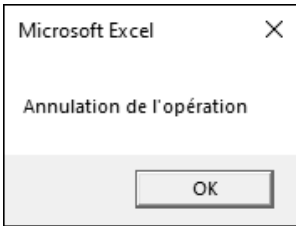
Indice

Utilisez la fonction `VBA.IsNumeric` pour tester la saisie.

Exercice 2

Durée estimative : 5 minutes

Complétez la procédure précédente qui vérifie que l'utilisateur a cliqué sur le bouton **OK**. Dans le cas contraire, affichez un message. Exemple :

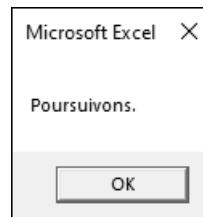
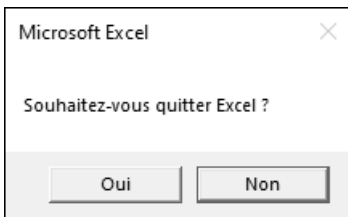
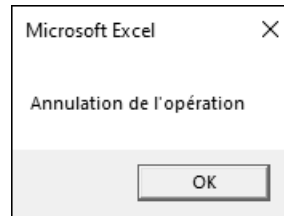
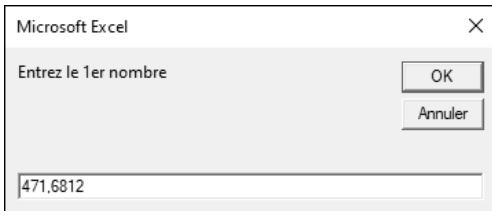


Corrigé p. 271

Énoncé 4.2 Dire au revoir

Durée estimative : 5 minutes

Complétez la procédure **AuRevoir**. Celle-ci propose une boîte de dialogue demandant si l'on souhaite ou non quitter Excel. Elle affiche le message "A bientôt!" si l'utilisateur clique sur le bouton **Oui**, et le message "Poursuivons" sinon. Exemple :



```
Sub AuRevoir()  
    Dim strMessage As String  
    Dim intStyle As Integer  
    Dim intChoix As Integer  
    strMessage = "Souhaitez-vous quitter Excel ?"  
    intStyle = vbYesNo + vbDefaultButton2  
    ... = MsgBox(strMessage, intStyle)  
    If ... = vbYes Then  
        MsgBox "A bientôt."  
    ...  
        MsgBox "Poursuivons."  
    ...  
End Sub
```

Indice

Voici un extrait de l'aide VBA en ligne. L'aide complète se trouve à l'adresse suivante : <https://docs.microsoft.com/fr-fr/office/vba/language/reference/user-interface-help/msgbox-function>

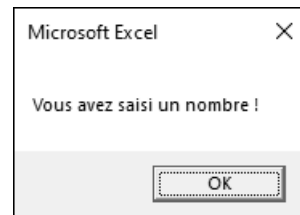
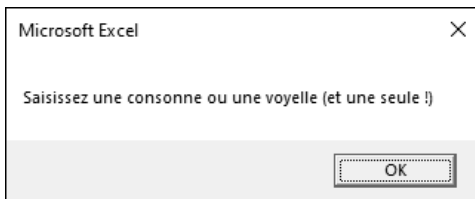
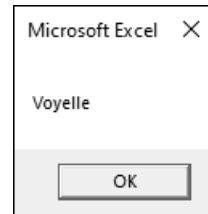
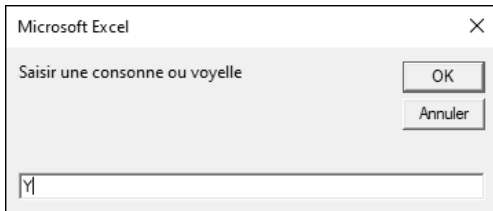
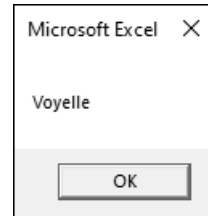
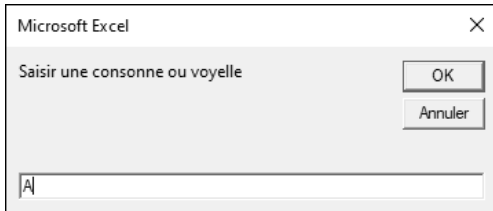
Constante	Valeur	Description
vbOKOnly	0	Bouton OK uniquement (par défaut)
vbOKCancel	1	Boutons OK et Annuler
vbAbortRetryIgnore	2	Bouton Abandonner, Réessayer et Ignorer
vbYesNoCancel	3	Boutons Oui, Non et Annuler .
vbYesNo	4	Boutons Oui et Non

Corrigé p. 272

Énoncé 4.3 Contrôler la saisie d'une consonne ou voyelle

Durée estimative : 10 minutes

Créez la procédure **ConsonneVoyelle** qui demande de saisir une voyelle ou une consonne. Affichez un message selon la saisie : "Voyelle", "Consonne" ou un message approprié en cas d'erreur. Exemple :



Indice

Pour tester la lettre en majuscule, on effectuera la transformation suivante :

```
varChoix = VBA.UCase(varChoix)
```

Corrigé p. 272

Énoncé 4.4 Afficher un message selon l'âge et le genre

Durée estimative : 15 minutes

Créez la procédure **VotreAge** qui demande le sexe et l'âge de l'utilisateur. Les messages affichés seront différents selon l'âge et selon qu'il s'agisse d'un homme ou d'une femme. Prenez également en considération l'abandon de la saisie et la saisie de lettres pour l'âge.

Pour les hommes :

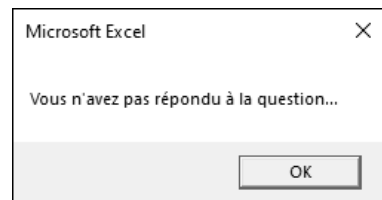
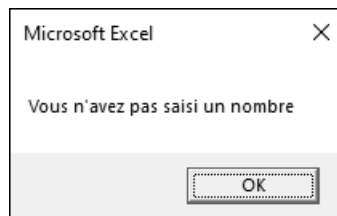
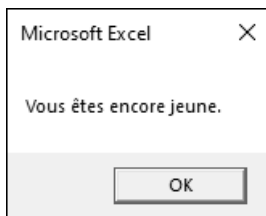
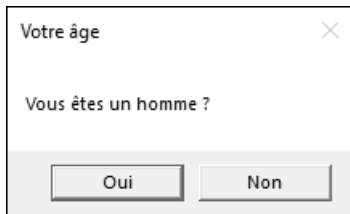
0 - 17 ans : "Gamin"

18 - 30 ans : "Vous êtes jeune"

31 - 50 ans : "Vous êtes encore jeune"

> 50 ans : "Vous commencez à vieillir"

Pour les femmes, un seul message : "On ne demande pas son âge à une femme". Exemple :



Indices

Déclaration des variables :

```
Dim strHomme As String * 1
Dim strFemme As String * 1
Dim intChoixSexe As Integer
Dim dblAge As Double
```

Pour afficher la première boîte de dialogue :

```
intChoixSexe = MsgBox("Vous êtes un homme ?", vbYesNo, "Votre âge")
```
