Memento VBA

Table des matières

Types

Voici quelques exemple s de types de variables disponibles en VBA :

1. Nombres entiers : Byte, Integer, Long, LongLong, LongPtr.
2. Nombres décimaux : Single, Double, Decimal, Currency.
3. Valeurs booléennes : Boolean.
4. Chaînes de caractères : String.
5. Dates : Date.

Déclarations des variables

Les variables et les constantes sont déclarées Private par défaut. Elles peuvent être déclarées Public ou Private de façon explicite.

Variables locales à une fonction ou à une procédure :

Exemples : Dim x As Integer, y As Double

Dim chaine As String

Variables globales à plusieurs procédures d'un même module :

Exemple : Dim resultat As Double (à placer en tête du code des procédures)

Variables globales dans un projet multimodule (applications avec formulaires…) :

Exemple : Public valeur As String (à placer en tête d'un module : Module1…)

Arguments d'une fonction ou d'une procédure :

Exemples : fct([ByVal ou ByRef] x As Integer,…) As Integer

proced([ByVal ou ByRef] y As Integer,…)

Affectation

Voici quelques exemples de syntaxes de l'instruction d'affectation :

x = 4, z = x+y, u = f(t)

Entrée d'information

L'instruction InputBox assure le tranfert d'information du clavier à la mémoire

centrale. Elle s'écrit par exemple de la façon suivante :

x=InputBox("donne la valeur de x : ")

Sortie d'information

L'instruction MsgBoxassure le transfert d'information de la mémoire centrale à l'écran.

Elle s'écrit par exemple de la façon suivante :

MsgBox "Le résultat vaut : " & res

Instruction conditionnelle

L'instruction conditionnelle permet de construire une alternative en testant si la condition est vraie ou fausse.

Syntaxes de instruction **If…Then…Else…End If** :

Forme standard	**Forme restreinte**
If Condition Then *Instruction 1a* *Instruction 1b* *(…..)* *Else* *Instruction 2a* *Instruction 2b* *(….)* *End if*	*If Condition Then* *Instruction 1a* *Instruction 1b* *(…..)* *End If*
Forme imbriquée syntaxe 1	**Forme imbriquée syntaxe 2**
If Condition1 Then *If Condition2 Then* *Instructions 2a* *Else* *Instructions 2b* *End if* *Else* *If Condition3 Then* *Instructions 3a* *Else* *Instructions 3b* *End if* *End if*	*If Condition1 Then* *Instructions1* *ElseIf Condition2 Then* *Instructions2* *ElseIf Condition3 Then* *Instructions3* *ElseIf (…)* *(…)* *[Else* *(…)]* *End If*

Syntaxe de l'instruction Select Case… :

Select Case Variable

Case Hypothèse_1

Instructions_1

Case Hypothèse_2

Instructions_2

(…)

Case Else

Instructions_n

End Select

Instruction itérative

Deux cas sont à distinguer selon que le nombre de répétitions est connu a priori ou bien qu'il dépend du contexte.

Le nombre de répétitions est connu a priori :

For var = val_début To val_fin [step p]

Instruction n°1

Instruction n°2

(…)

Next var

Le nombre de répétitions dépend du contexte :

Nous avons 2 types d'instructions

While Condition •

Instruction n°1 •

Instruction n°2

(…)

Wend

Do

Instruction n°1

Instruction n°2

(…)

Loop Until Condition

L'instruction **Do…Loop** peut être déclinée sous 4 formes :

Cond : Condition d'arrêt	**Cond : Condition de continuation**
Do *Instruction n°1* *(…)* *Loop Until Cond*	*Do* *Instruction n°1* *(…)* *Loop While Cond*
Do Until Cond *Instruction n°1* *(…)* *Loop*	*Do While Cond* *Instruction n°1* *(…)* *Loop*

Opérateurs

Il existe quatre familles d'opérateurs :

1. Les opérateurs numériques : +, –, *, /.

   ^ : élévation à la puissance.

   \ : division entière. Exemple : 23\3 donne 7.

   Mod : reste de la division entière. Exemple : 23 Mod 3 donne 2.

2. Les opérateurs de comparaison : <, <=, >, >=, =, <>.
3. Les opérateurs logiques : And, Or, Not, Xor, Eqv.
4. L'opérateur de concaténation : &.

Fonctions

Les fonctions effectuent des traitements informatiques et fournissent des résultats.

Syntaxe :

Function mafonction([ByVal ou ByRef] x As …, …) As …

    Dim … 'déclaration des variables locales

    (…)

    Instruction n°1

    Instruction n°2

    (…)

    mafonction = … 'valeur de retour

End Function

Exemples d'en-têtes de fonction :

Function fnum(…, …) As Double : fonction numérique.

Function fch(…, …) As String : fonction chaînes de caractères.

Function fbool(…, …) As Boolean : fonction booléenne.

Exemples d'appels de fonction : mettre des parenthèses

```
val_res = fnum(a,b)

ch_res = fch(ch1,x)

bool_res = fbool(x,5)
```

Procédures

Les fonctions effectuent des traitements informatiques et agissent directement sur l'environnement d'exécution, sans fournir de résultat.

Syntaxe :

```
Sub proced1() 'sans paramètres

ou Sub proced2(déclaration de paramètres ) 'avec paramètres

Dim … 'déclaration des variables locales

(…)

Instruction n°1

Instruction n°2

(…)

End Sub
```

Exemples d'en-têtes de procédure :

Sub calcul() : sans paramètres.

Sub etude(x As Integer, y As Integer) : avec deux paramètres.

Exemples d'appels de procédure : ne pas mettre de parenthèses

Sans paramètre : calcul

Avec paramètres : etude a, b

Les procédures et les fonctions sont déclarées Public par défaut. Elles peuvent être

déclarées Public ou Private de façon explicite.

Fonctions prédéfinies

Voici quelques exemples de fonctions prédéfinies du langage VBA :

Mathématiques : Abs(x), Sqr(x), Cos(x), Sin(x), Tan(x), Exp(x), Log(x), Int(x),

Fix(x), Round(x,nb)…

Chaînes de caractères : Len(ch), InStr(ch,ch2), Mid(ch,d,n), Mid(ch,d),

Left(ch,n), Right(ch,n), Replace(ch,ch1,ch2), Trim(ch), Ltrim(ch), Rtrim(ch),

Lcase(ch), Ucase(ch)…

Conversion des données : Asc(ch), CInt(ch), CDbl(ch), CLng(ch), Chr(n), CStr(n)…

Nombres aléatoires

Les nombres aléatoires sont calculés de la façon suivante :

Randomize : réinitialise le processus aléatoire.

Rnd() : donne un nombre aléatoire décimal tel que 0 <= Rnd() <1.

Int(b – a + 1) * Rnd() + a) : donne des nombres entiers aléatoires compris entre a et b, bornes comprises.

Exemple : Int(6 * Rnd() + 1) : donne des nombres entiers compris entre 1 et 6

(tirage d'un dé).

Les objets Excel

Les applications Excel mettent en jeu de nombreux objets qui sont définis selon un modèle. Les classeurs, les feuilles de calcul et les cellules peuvent être représentés par des variables objets. Il en est de même pour les objets de contrôle et les formulaires.

Voici une présentation synthétique de ces variables et de leurs propriétés.

Types, collections et variables objets

Le modèle objet d'Excel est structuré par plusieurs classes. Les classes les plus importantes sont définies par les types suivants :

• **Object** : la classe la plus générale.

• **Application** : l'application Excel elle-même.

• **Workbook** : la classe des classeurs.

• **Worksheet** : la classe des feuilles de calcul.

• **Range** : la classe des plages de cellules.

Le modèle objet comporte aussi plusieurs collections :

• **Workbooks** : l'ensemble des objets classeurs.

• **Worksheets** : l'ensemble des objets feuilles de calcul.

• **Charts** : l'ensemble des objets graphiques.

• **Sheets** : l'ensemble des objets feuilles de calcul et des objets graphiques.

Exemple de déclaration de variables objets :

```
Dim cls As Workbook, fe As Worksheet, plage As Range
Set cls = Workbooks("Exemple.xls")
 Set fe = cls.Worksheets("Stock")
Set plage = fe.Range("B1:C5")
```

ou bien :

```
Set plage = Workbooks("Exemple.xls").Worksheets("Stock").Range("B1:C5")
```

Les collections regroupent les objets du type correspondant :

- Workbooks("Exemple.xls")est un objet de la collection Workbooks, de type Workbook.
- Worksheets("Stock")est un objet de la collection Worksheets, de type Worksheet.
- Écriture particulière : Range("B1:C5") définit une collection de cellules, de type Range.
- Set… affecte à chaque variable l'adresse de l'objet correspondant.

Les déclarations peuvent être allégées s'il n'y a qu'un classeur et qu'une feuille de calcul dans l'application étudiée. Dans ce cas, on peut écrire : Set plage = Range("B1:C5").

Désignation des cellules

Les cellules peuvent être désignées sans déclaration préalable des variables objets ou bien avec une déclaration préalable des variables objets.

Sans déclaration des variables objets

- Range("A2") désigne la cellule A2 de la feuille active.
- Range("A2:B5") désigne la plage A2:B5 de la feuille active.
- Range("A2:B5").Cells(i, j)désigne la cellule de la ième ligne et de la jième colonne relative à la plage A2:B5.
- Range("A2:B5").Cells(1, 1) désigne la cellule A2 car Cells(i, j) commence à 1, 1.
- Range("A2:B5").Cells(3, 2) désigne la cellule B4.

Lorsqu'il n'y a qu'une seule feuille de calcul :

- Cells(i, j) désigne la cellule de la ième ligne et de la jième colonne de la feuille decalcul.
- Cells(3, 2) désigne la cellule B3 de la feuille active.
- Cells(i, 2) désigne la cellule de la ième ligne de la deuxième colonne de la feuille active.

Avec déclaration des variables objets

Premier exemple :

- Dim p as Range déclare la variable p comme une variable de type Range.
- Set p = Range("B2:D8") affecte à la variable p l'adresse de la plage B2:D8.
- p.Cells(i, j) désigne la cellule de la ième ligne et jième colonne relative à la plage B2:D8.
- p.Cells(1, 1) désigne la cellule B2 car Cells(i, j) commence à 1, 1.
- p.Cells(3, 2) désigne la cellule C4.

Deuxième exemple :

- Set p = Range("B1:B5") affecte à la variable p l'adresse de la plage B1:B5.
- p.Cells(i) désigne la cellule de la ième ligne relative à la plage B1:B5 (une seule colonne).

Troisième exemple :

- Dim c as Range déclare la variable c comme une variable de type Range.
- Set c= Range("C3") affecte à la variable c l'adresse de la cellule C3. •
- c.Offset(i, j) désigne la cellule obtenue par le déplacement i,j à partir de la cellule C3.
- c.Offset(0, 0) désigne la cellule C3 car Offset(i, j) commence à 0, 0.

- c.Offset(1, 0) désigne la cellule C4.
- c.Offset(2, 1) désigne la cellule D5.
- c.Offset(–1, 1) désigne la cellule D2.

Valeur d'une cellule

La propriété Value permet de définir la valeur d'une cellule :

- Range("A2").Value donne la valeur de la cellule A2.
- x=Range("A2").Value affecte à x la valeur de la cellule A2.
- p.Cells(i, j).Value donne la valeur d'une cellule relative à la plage p.
- c.Offset(i, j).Value donne la valeur d'une cellule définie à partir de la cellule c.

Lignes et colonnes d'une plage

Les plages de cellules sont structurées par des lignes et des colonnes :

- p.Rows désigne la collection des lignes de la plage p.
- p.Rows.Count donne le nombre de lignes de la plage p.
- p.Columns désigne la collection des colonnes de la plage p.
- p.Columns.Count donne le nombre de colonnes de la plage p.
- p.Count donne le nombre total de cellules de la plage p.

Plage et cellule sélectionnées par l'utilisateur

Toute plage et toute cellule sélectionnée par l'utilisateur au sein de la feuille de calcul peut être représentée par une variable objet de type Range :

- Set p = Selection : lorsqu'une plage a été sélectionnée, Selection attribue à pl'adresse de cette plage.
- Set c = ActiveCell : lorsqu'une cellule a été sélectionnée, ActiveCell affecte à c L'adresse de cette cellule.

Actions sur les cellules

Plusieurs actions sont possibles sur les cellules :

- p.ClearContents efface le contenu des cellules de la plage p.
- p.Clear efface le contenu et le formatage des cellules de la plage p.
- c.Interior.Pattern = xlPatternNone efface la couleur de la cellule c.

Couleurs dans les cellules

Deux modalités permettent de colorier des cellules :

- c.Interior.Color = RGB(x, y, z) met une couleur au sein de la cellule c quidépend des valeurs de x, y et z.
- c.Interior.ColorIndex = i met la couleur n° i au sein de la cellule c (la couleur fait partie d'un panel de 56 couleurs).

Objets de contrôle

Les objets de contrôle jouent le rôle d'interface entre l'utilisateur et les applications

Excel. Ils peuvent faire partie d'objets formulaires. Leurs caractéristiques principales

sont les suivantes :

- Types : Label, TextBox, ListBox, ComboBox, CommandButton, CheckBox, OptionButton,

ScrollBar…

- Propriétés : Name, Caption…
- Méthodes : selon les types.
- Événements : Click, DblClick…

Objets formulaires

Les formulaires sont des objets d'interfaces entre l'utilisateur et l'application. Leurs caractéristiques principales sont les suivantes :

- Types : UserForm
- Propriétés : Name, Caption…
- Méthodes : Show, Hide, Load, Unload…
- Événements : Initialize, Activate, Desactivate, Click, DblClick…