Programmation Excel/VBA

VBA
CONCEPTS DE BASE

MASTER SES

UNIVERSITÉ NANTERRE LA DÉFENSE
SANA BEN HAMIDA

Plan

1. Les variables et les constantes
2. Les boites de dialogues
3. Expressions arithmétiques et opérateurs
4. Concordance de types
5. Les procédures et fonctions
6. Les structures de contrôle : Instructions conditionnelles
7. Les structures de contrôle : Instructions répétitives

Les variables

3

- Le programmeur utilise la mémoire au travers de la notion de variable
- Une variable est un « mot mémoire » auquel le programmeur donne un **nom** et un **type**
  - nom ➔ désigne l'emplacement et/ou la valeur qu'il contient
  - type ➔ indique à la machine quel codage employer pour les valeurs qui seront rangées dans la variable

Les variables: les types

4

Type	Nb Octets	Domaine
Byte	1	0..255
Boolean	2	True / False
Integer	2	-32768 .. 32767
Long	4	-2147483648.. 2147483647
Single	4	-3,402823E38..-1,401298E-45 ; 1,401298E-45...3,402823E38
Double	8	4,94065645841247E-324 .. 1,79769313486232E308 (>0 et <0)
Date	8	1/1/100 à 0:0:0 .. 31/12/9999 à 23:59:59
String	1 par caract	longueur indéfinie(<2000000) ou fixée (<65536)
Variant		prend selon les divers affectations les valeurs précédentes ainsi que Empty, Error ou Null
...	...	...

Les variables: déclaration

5

- Dim x As Integer
  - ➔ Dans l'emplacement appelé x, on pourra ranger un entier
- Dim x
  - ➔ x variant
- Dim nombre1,nombre2 As Integer
  - ➔ ***nombre1 n'est pas déclaré comme Integer mais comme Variant .***

Les constantes

6

La constante est comme une variable mais sa valeur ne peut être modifiée après sa déclaration;

Deux types de constantes:

1. Constantes utilisateur **typées**

Const Pi as Single = 3.14

2. Constantes utilisateur **non typées**

Const Pi = 3.14

➔prend plus de place en mémoire que la constante précédente

Les affectations

7

Syntaxe : Nomvar = expression

- « Nomvar » est un nom de variable déclarée
- « Expression » décrit un calcul ou appelle une fonction à partir d'opérandes et d'arguments

Effet :

- L'ordinateur calcule la valeur de l'expression, puis range cette valeur dans l'espace mémoire associé à la variable

Exemple expression arithmétique

8

```
Sub exemple2()
Dim a As Integer
Dim b As Integer
a=15     'met la valeur entière 15 dans a.
b=a+2     'met la valeur entière 17 dans b
b=b*2 +a+10
 'met la valeur entière 59 dans la variable b
End Sub
```

Les boîtes à messages

9

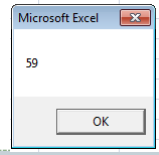
- **MsgBox** ➔ permet à une macro d'afficher (à l'utilisateur) une information sous forme d'une boite de dialogue

Syntaxe :
MsgBox(«message à afficher»)

- Exemple

MsgBox(b) 'exemple précédent

Effet:



Les boîtes à entrées

10

- **InputBox** ➔ permet à l'utilisateur de fournir des données à une macro

Syntaxe :
réponse= InputBox(«message à afficher»)

- Une boite de dialogue est affichée avec le message dans la zone grise.
- L'utilisateur entre une donnée dans la zone blanche
- La valeur rentrée par l'utilisateur est le résultat de la fonction (on dit que la fonction retourne cette valeur)
- Cette valeur est rangée dans la variable réponse

Les boîtes à entrées: Exemple

11

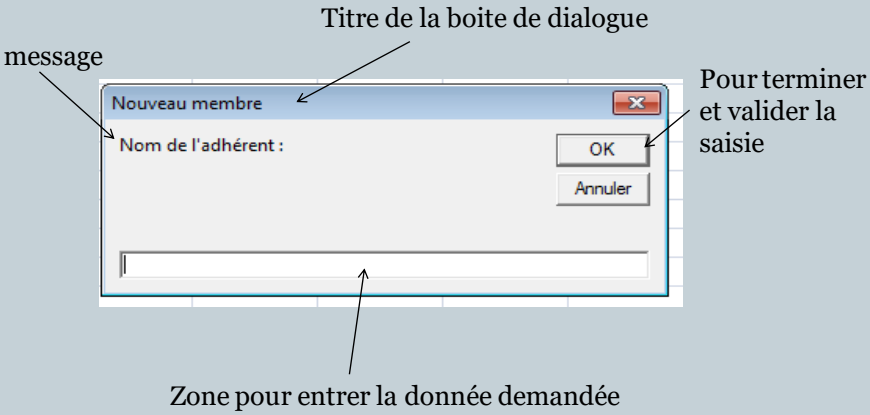
```
Sub UtilisationDeInputboxFonction()

    Dim Inscription As String
    Inscription = InputBox("Nom de l'adhérent :", "Nouveau
membre")
    ' Si l'utilisateur n'a saisi aucune donnée et qu'il clique sur
OK, ou si l'utilisateur clique sur Annuler, on quitte la procédure.

 If Inscription = "" Then
        MsgBox "Aucune donnée n'a été saisie"
        Exit Sub
    ' Sinon la donnée saisie est affichée
    Else
        MsgBox (Inscription)
    End If
End Sub
```

Les boîtes à entrées: Exemple

12



Concordance de type

13

- VBA peut faire des conversions de type.
- Exemple: **nomvariable=expression**

→ Si la variable est de type *integer*, l'expression peut-être *byte* ou *Integer*

→ Si la variable est de type *double*, l'expression peut-être *byte, Integer* ou *double*

→ Si la variable est de type *String*, toutes les valeurs de l'expression sont transformées en chaînes de caractères.

Concordance de type

14

- Effets de bord → conversion inattendues

Exemple:

- Si la variable est de type entier et l'expression de type double, il calcule la partie entière ou l'arrondi:

```
Dim x As Integer
x=3.5 'x vaut 4 (arrondi)
```

Les opérateurs binaires sous VBA

15

4 types d'opérateurs binaires:

1. **Opérateurs arithmétiques** (+,-,*,/, Mod)
2. **Opérateurs de comparaison** (<,>,=,<=,>=,<>)
3. **Opérateurs logiques** (And, Or) (+ opérateur unaire not)
4. **Opérateurs sur les textes** (&, +)

Syntaxe:
expression1 **opérateur** expression2

Liste des opérateurs arithmétiques

16

Opérateur	Définition	Exemple
^	permet d'élever un nombre à une puissance	2^3 *'vaut 8*
*	multiplie 2 nombres	2*3 *'vaut 6*
/	divise 2 nombres	20/3 *'vaut 6.6666667*
\	divise 2 nombres en renvoyant un entier	20\3 *'vaut 6*
Mod	(modulo) renvoie un entier qui est le reste de la division de 2 nombres	20 Mod 3 *'vaut 2*
-	soustrait 2 nombres ou inverse le signe d'un nombre	20-3 *'vaut 17*
+	ajoute 2 nombre	20+3 *'vaut 23*

Comparaison de chaines : exemples

17

Les opérateurs numériques vous permettent de comparer des valeurs **String** en fonction de leur ordre de tri

"73" < "9" ' True ➔ le premier caractère de la première chaîne est trié avant le premier caractère de la deuxième chaîne

---

"aaa" > "aa" ' True ➔ comparaison textuelle

Opérateurs logiques: exemples

18

a = (23 > 14) And (11 > 8) ‘true
b = (14 > 23) And (11 > 8) ‘false

c = (23 > 14) Or (8 > 11) ‘true
d = (23 > 67) Or (8 > 11) ‘false

h = Not (23 > 14) ‘false
i = Not (23 > 67) ‘true

Opérateurs de concaténation (sur le texte)

19

```
Dim x As String = "Con" & "caten" & "ation"
Dim y As String = "Con" + "caten" + "ation“

' Les deux variables x et y contiennent le mot
   "Concatenation".
```

& n'est défini que pour des opérandes **String**

Les procédures et fonctions

20

Procédures et fonctions

21

- Trois types:
  1. **Fonction**: Peut prendre des arguments et doit retourner une seule valeur;
  2. **Procédure** : Peut prendre des arguments mais ne retourne pas de valeur;
  3. **Macro**: ne prend pas d'argument et ne retourne pas de valeur;

Seules les macros peuvent s'exécuter

Déclaration de procédures

22

Sub <ma_proc>(<arguments> As <type arg>)
<déclaration des variables locales et statiques>
<Corps de procédures>
End sub

As <type arg.> est facultatif (Variant par défaut)

Déclaration de procédures

23

```
Private Sub test_proc(a As String)
MsgBox (a)
End Sub
'procédure qui affiche son argument

Sub cube(Dim a As Integer)
Dim b As Integer
b = a ^ 3
MsgBox b
End Sub
'procédure qui calcule et affiche le cube de son argument
```

Appel de procédures

24

On peut appeler une procédure de 3 façons différentes:

- **Call <ma_proc> (<mes_val>)**
- <ma_proc> <mes_val>
- <ma_proc> <nom_arg> **:= <mes_val>**

exemple: **MsgBox "<Message>" est considéré comme une procédure**

Appel de procédures

25

Exemple:

```
Sub Exemple_appel_proc()
Dim d As Integer ' variable locale
test_proc  "abcd"
Call test_proc("def")
test_proc a:="ghi“

End sub
```

Déclaration de fonctions

26

Function <ma_fonc> (<arguments> As <type arg>) As <Type du résultat>

<déclaration des variables locales et statiques>

<Corps de fonction>

<ma_fonc> = **<résultat>**

End Function

Retour de la fonction

As <type arg> et **As <Type du résultat>** sont facultatifs (Variant par défaut)

Déclaration de fonctions

27

```
Function dble(x As Integer) As Integer
dble = x * 2
End Function
'fonction qui calcule le double de son argument et le retourne
Function cube(a As Integer) As Long
cube= a ^ 3
End Function
'fonction qui calcule le cube de son argument et le retourne
```

Appel de fonctions

28

classique :

- **<ma_var> = <mafonc>(<mes_arg>)**

ou pour une fonction qui produit des effets de bords dont on ne voudrai pas récupérer le résultat

- **<mafonc>(mes_arg>)**

Appel de fonctions

29

- exemple classique : MsgBox est une fonction qui renvoie le numéro du bouton sur lequel l'utilisateur a appuyé (il peut y en avoir de 1 à 3) , s'il y a qu'un bouton il est inutile de mémoriser le résultat:
- **MsgBox (" <Message> ")**
- ou avec 3 boutons:
- <var_reponse> = **MsgBox (" < Message> ",VbAbortRetryIgnore)**

Les fonctions élémentaires de VBA

30

- VBA comporte un très grand nombre de fonctions prédéfinies. La syntaxe de l'appel:

Nomdefonction(paramètre1,paramètre2,..etc)

- Les paramètres sont des expressions de type indiqué dans le mode d'emploi de la fonction (ou compatible)
- Le type du résultat est également indiqué dans le mode d'emploi de la fonction.
- Attention le séparateur entre deux paramètres est la virgule!

Les fonctions élémentaires de VBA

31

- **Int(nombre**):un seul paramètre de type double, calcule la partie entière inférieure
- **Rnd**():sans paramètres, tire un nombre au hasard dans [0,1[avec une loi uniforme.
- **sqr(nombre**):un seul paramètre de type double, calcule la racine carrée.
- Autres fonctions mathématiques:
  log(logarithme népérien), **abs**(valeur absolue)

Les fonctions élémentaires de VBA

32

- Un programme qui propose une multiplication et vérifie que la réponse de l'utilisateur est correcte :

```
Sub multiplication()
Dim x As Integer
Dim y As Integer
Dim res As Integer
Dim gagne As Boolean
x=Int(Rnd()*9)+1'Calcul d'un nombre aléatoire entier entre 1 et 10
y=Int(Rnd()*9)+1
res=InputBox("Combien font " & x & " fois " & y & "?")
gagne=(res=x*y)
MsgBox("votre calcul est "& gagne)
End Sub
```

Structures de contrôle

33

ALTERNATIVES ET RÉPÉTITIVES

Structures de contrôles

34

Deux types de structure :

- Structures de contrôle ***alternatives*** **(**choix entre plusieurs séquences d'instructions)
- Structures de contrôle ***répétitives*** (possibilité de répéter sous conditions, une séquence d'instruction)

Instructions conditionnelles

35

Syntaxe 1:

If <cond> **Then** <instr1>**:** <instr2 >**: … :**<instr n>

End If

Syntaxe 2:

If <cond> **Then**

<instr1>

<instr2 >

…

<instr n>

End If

Instructions conditionnelles

36

Syntaxe 3:

If <cond> **Then**

<instr alors 1>

…

<instr alors n>

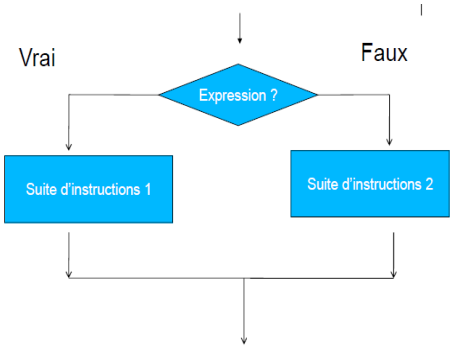
Else

<instr sinon 1>

…

<instr sinon p>

End If



Instructions conditionnelles

37

Syntaxe 4:

If <cond 1> **Then**
<instr alors1>
...
<instr alorsn>
Elseif <cond 2> **Then**
<suite instr>
...
Elseif <cond 3> **Then**
<suite instr>
...
Else
<instr sinon1>
...
<instr sinon p>
End If

Le dernier est un « Else »

Instructions conditionnelles

38

Exemple1:

```
Sub moyen ()
Dim note1 as double
Dim note2 as double
Dim moy as double
note1=InputBox("entrez votre première note")
note2=InputBox("entrez votre seconde note")
moy=(note1+note2)/2
If moy<10 Then
MsgBox("eliminé")
Else
MsgBox("Reçu")
End If
End Sub
```

Instructions conditionnelles

39

Exemple2:

```
Sub ChoixOpération()
Dim x As Double, y As Double, r As Double, choix As String
x = InputBox("donner la valeur de x")
y = InputBox("donner la valeur de y")
choix = InputBox("Quelle opération vous souhaitez appliquer?" & vbCr & _
"1 pour addition" & vbCr & "2 pour soustraction" & vbCr & _
"3 pour division" & vbCr & "4 pour multiplication")
If choix = 1 Then
MsgBox ("l'addition de " & x & " et " & y & " vaut " & x + y)
ElseIf choix = 2 Then
MsgBox ("la soustraction de " & x & " et " & y & " vaut " & x - y)
ElseIf choix = 3 Then
MsgBox ("la division de " & x & " par " & y & " vaut " & x / y)
ElseIf choix = 4 Then
MsgBox ("la multiplication de " & x & " par " & y & " vaut " & x * 4)
Else
MsgBox ("Ce choix n'est pas dans la liste")
End If
End Sub
```

Le cas

40

Définition :

Choix d'une séquence d'instruction parmi plusieurs selon la valeur d'une expression

Remarque :

Dans les exemples suivants <Expr>, <valeur 1>, … ,<valeur n> sont des expressions de même type

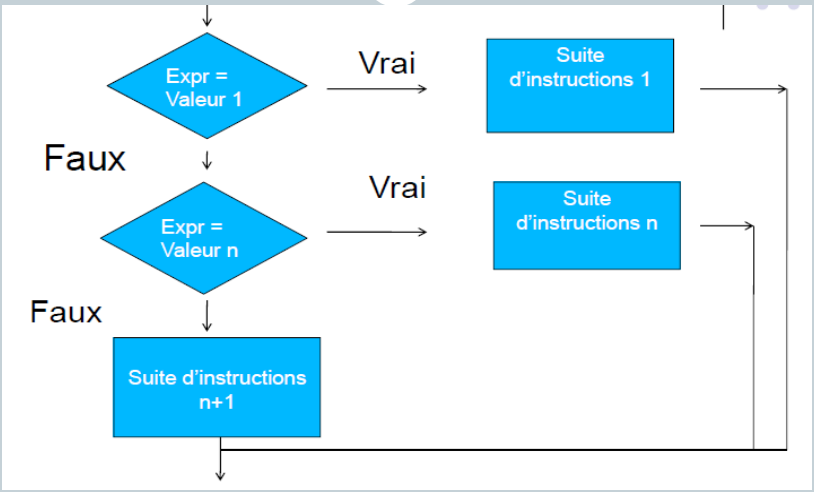
Le cas

41

Select Case expr
Case valeur 1
Suite instructions 1
Case valeur 2
Suite instructions 2
.
.
.
Case valeur n
Suite instructions n
Case Else *´(facultatif)*
Suite instructions n+1
End Select

Le cas

42



Le cas

43

```
Sub departement()
Dim Deptnum As Integer
Dim Deptnom As String
Deptnum = InputBox(" entrez un
    numéro _
de département ")
Select Case Deptnum
Deptnom= "Hauts de Seine"
Case 93
Deptnom= "Seine Saint Denis"
Case 94
Deptnom= "Val de Marne"
Case 95
Deptnom= "Val d'oise"
Case 75
Deptnom="Paris"
Case 91
Deptnom=" Essonne "
 Case 77
 Deptnom= "Seine et Marne"
 Case 78
 Deptnom="Yvelines"
 Case 92
 Deptnom= "Val d'oise"
 Case Else
 Deptnom= "ce n'est pas en ile
    de France"
 End Select
 MsgBox(Deptnom)
 End Sub
```

Les boucles

44

Il existe 4 syntaxes possibles

Do ... loop while **Do while ... loop**	Exécute une suite d'instructions en boucle **tant que** la condition est vraie (True)
Do ... loop until **Do until ... loop**	Exécute une suite d'instructions en boucle **jusqu'à ce que** la condition soit vraie (True)
For ..Step ..Next	Exécute une suite d'instructions en boucle autant de fois que spécifié
Foreach In ...Next	Parcourt tous les éléments d'une collection

Boucle *Do while…loop*

45

- L'instruction **Do while loop** permet de réaliser une boucle conditionnelle.

Syntaxte:

Do while <condition>

<instructions>

Loop

Attention, la condition est évaluée au début des traitements

Boucle *Do while…loop*

46

Exemple:

- Calcul du produit factoriel d'un entier saisi.

```
sub Testboucle1()
   Dim Reponse As Integer, n As Integer, cpt As Integer
   Dim p as Long
   Reponse=InputBox(« Entrer un entier entre 1 et 20 »)
p=1
cpt=2
Do While cpt <= Reponse
p=p*cpt
cpt=cpt+1
Loop
MsgBox("Le produit factoriel de " & Reponse & " est "  & p)
End Sub
```

Boucle *Do … loop until*

47

- L'instruction ***Do … loop until*** est similaire *à Do while loop*, cependant les instructions sont d'abord exécutées, à la suite de quoi la condition est évaluée.

Syntaxte:

Do

<instructions>

Loop until <condition>

Boucle *Do … loop until*

48

- Exemple:

Demander à l'utilisateur d'entrer un chiffre entre 1 et 10 (la boucle permet d'effectuer un contrôle sur la valeur saisie)

```
sub Testboucle2()
  Dim Reponse As Integer
  Do
    Reponse = InputBox("Entrez un chiffre entre 1 et 10")
  Loop Until (Reponse >=1 and Reponse <=10)
End Sub
```

Loop While (Reponse <1 or Reponse > 10)

Boucle *For Step Next*

49

- L'instruction ***For Step Next*** est utilisée pour répéter une action selon un nombre d'itérations déterminé.

Syntaxte:

For <variable>= valeur1 **to** valeur2 **step** <pas>

<instructions>

Next<variable>

<pas> sert à indiquer le pas d'itérations (facultatif, valeur par défaut 1)

Boucle *For Step Next*

50

Exemple:

- Afficher les nombres paires entre 1 et 20

```
sub TestBoucle3()
Dim nombre As Integer
For nombre = 0 to 20 Step 2
MsgBox nombre
Next nombre
End Sub
```