

VBA : Variables et conditions

Laure Tougne

Plan du cours

- Les variables
 - Les types de variables
 - Portée d'une variable
 - Récupérer le contenu d'une cellule dans une variable
 - Ecrire le contenu d'une variable dans une cellule
 - Les tableaux
- Les conditions
 - IF...
 - SELECT CASE...

LES VARIABLES

Variable

- Une variable est une donnée nommée dans laquelle on peut mettre une information : chiffre, date, texte, etc.
- Mieux vaut déclarer une variable même si VB autorise de travailler avec des variables sans les déclarer
- Il est possible de rendre les déclarations obligatoires (Outils/Options)

Types de variables

- Une douzaine de types de variables
- 4 ou 5 que nous utiliserons :
 - Chaîne de caractères : *String*
 - Dates : *Date*
 - Objets : *Object* (feuille de calcul, classeur, cellule, etc.)
 - Nombres : *Double*, *Long*, etc. (rem : les autres types nombres sont automatiquement convertis en cela par VB)
 - Booléens : *Boolean*
 - Fourre-tout : *Variant*

Déclaration d'une variable

- Plusieurs manières de déclarer une variable
 - **Dim Nom_De_La_Variable As Type_variable**
 - déclarer en début de procédure (connue dans la procédure)
 - initialisée par défaut
 - **STATIC Nom_De_La_Variable As Type_variable**
 - une variable déclarer en STATIC conserve sa valeur en mémoire, tant que le classeur Excel est ouvert
 - si on relance le code, la dernière valeur sera utilisée
 - **PUBLIC Nom_De_La_Variable As Type_variable**
 - au début d'un module, la variable sera visible par toutes les procédures du projet
 - la valeur est conservée tant qu'Excel reste ouvert

Exemple 1

```
Sub Variable_Static()
Static Num_Ligne As Long
Num_Ligne = Num_Ligne + 1
Cells(Num_Ligne, 1) = Num_Ligne
End Sub
```

	A	B	C	D	E
1	1				
2	2				
3	3				
4	4				
5	5				
6	6				
7					
8					
9					
10					
11					Exécution
12					

	A	B	C	D	E
1	1				
2	2				
3	3				
4	4				
5	5				
6	6				
7	7				
8					
9					
10					
11					Exécution
12					

	A	B	C	D	E
1	1				
2	2				
3	3				
4	4				
5	5				
6	6				
7	7				
8	8				
9					
10					
11					Exécution
12					

Exemple 2 (1)

- Ecrire un programme qui affiche un numéro de ligne, l'additionne avec lui-même et multiplie ce numéro par lui-même

```
Sub Calcul()
    Cells(1, 1) = 1
    Cells(1, 2) = 1 + 1
    Cells(1, 3) = 1 * 1
End Sub
```

	A	B	C	D	E	F
1	1		2	1		
2						
3						
4						
5						
6						

The screenshot shows a Microsoft Excel spreadsheet. Column A contains the numbers 1 through 6. Column B contains the values 1, 2, and 1 respectively. Column C is empty. Column D is empty. Column E is empty. Column F contains the word "Calcul". The code in the question is a VBA subroutine named "Calcul" that performs the operations described in the question.

Exemple 2 (2)

- La même chose avec une variable

```
Sub Calcul_2()
    Dim Valeur As Long

    Valeur = 1
    Cells(1, 1) = Valeur
    Cells(1, 2) = Valeur + Valeur
    Cells(1, 3) = Valeur * Valeur
End Sub
```

	A	B	C	D	E
1	1	2	1	Calcul_2	
2					
3					
4					
5					
6					

Exemple 2 (3)

- Changeons la valeur initiale de la variable

```
Sub Calcul_2()
    Dim Valeur As Long

    Valeur = 2
    Cells(1, 1) = Valeur
    Cells(1, 2) = Valeur + Valeur
    Cells(1, 3) = Valeur * Valeur
End Sub
```

	A	B	C	D	E
1	2	4	4		
2					
3					
4					
5					
6					
7					
8					

Exemple 2 (4)

- Utilisons Valeur comme numéro de ligne

```
Sub Calcul_3()
    Static Valeur As Long

    Valeur = Valeur + 1
    Cells(Valeur, 1) = Valeur
    Cells(Valeur, 2) = Valeur + Valeur
    Cells(Valeur, 3) = Valeur * Valeur
End Sub
```

	A	B	C	D	E	F
1	1	2	1			
2	2	4	4			
3	3	6	9			
4	4	8	16			
5						
6						
7						
8						

Exemple avec une fonction

- Ecrivons une fonction VBA qui calcule l'âge d'une personne en fonction de sa date de naissance

```
Function Calcule_Age(dateDeNaissance As Date)
    Dim Age As Double
    Age = Abs(DateDiff("YYYY", dateDeNaissance, Date))
    Calcule_Age = Age & " ans"
End Function
```

	A	B	C	D
1				
2		22/10/1995		21 ans
3		20/01/1999		17 ans
4				
5				

Les tableaux

- Permet d'affecter plusieurs valeurs à une seule variable
- Déclaration

Dim Tableau(30) As Double

- Le premier élément du tableau est à la position 0
- Ce tableau peut contenir 31 données

Utilisation des tableaux avec Excel

- Afficher en A1 le premier élément du tableau

Cells(1,1)=Tableau(0)

- Afficher le dernier élément du tableau en A2

Cells(1,2)=Tableau(30)

- Attention

Cells(1,3)=Tableau(31)

Dépassemement de capacité !

Chargement d'un tableau

- 2 méthodes
 - Au moyen d'une **boucle** (on reverra cela la prochaine fois ;-))
 - L'instruction **Array**

Array(donnée1, donnée2,donnée3,...)

→ avec cette méthode, le type du tableau est obligatoirement Variant

Exemple

```
Sub Chargement_Tableau()
    Dim Tableau As Variant

    Tableau = Array("Lundi", "Mardi", "Mercredi", "Jeudi", "Vendredi", "Samedi", "Dimanche")

    Cells(1, 1) = Tableau(0)
    Cells(2, 1) = Tableau(1)
    Cells(3, 1) = Tableau(2)
    Cells(4, 1) = Tableau(3)
    Cells(5, 1) = Tableau(4)
    Cells(6, 1) = Tableau(5)
    Cells(7, 1) = Tableau(6)

End Sub
```

	A	B	C	D
1	Lundi			
2	Mardi			
3	Mercredi			
4	Jeudi			
5	Vendredi			
6	Samedi			
7	Dimanche			
8				

Redimensionnement d'un tableau

- Il est possible de déclarer un tableau sans dimension
Ex : Dim Tableau() As String
- Et ensuite de le redimensionner au cours du programme
Ex : ReDim Tableau(10)

Exemple

- On veut récupérer dans un tableau les éléments de la colonne A

```
Sub Chargement2_Tableau()
    Dim Nb_Lignes As Long
    Dim Tableau() As String

    Nb_Lignes = Cells(1, 1).CurrentRegion.Rows.Count
    Cells(1, 6) = Nb_Lignes
    ReDim Tableau(Nb_Lignes - 1)
End Sub
```

	A	B	C	D	E	F
1	Toto					6
2	Titi					
3	Tutu					
4	Tata					
5	TonTon					
6	Tete					
7						
8						
9						

Remarques

- On peut spécifier qu'un tableau commence à la position 1
Option Base 1 'A mettre en début de module

- Le redimensionnement d'un tableau entraîne son initialisation. Pour préserver les données :
ReDim Preserve Tableau (Nb_Lignes-1)

Tableaux à plusieurs dimensions

- Déclaration

Dim Nom_Tableau (x,y,z,...) As Type

- Exemple

Dim Tableau(3,4) As String

→ Tableau de 3 lignes et 4 colonnes

Exemple

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	0	1	1	1	1	0									
2	1	0	0	0	0	1									
3	1	0	0	0	0	1									
4	1	0	0	0	0	1									
5	1	0	0	0	0	1									
6	0	1	1	1	1	0									
7															
8															

Chargement !

Ce sera plus facile avec
Des boucles !

```
Sub Chargement3_Tableau()
    Dim Tableau(6, 5) As Byte

    Tableau(0, 0) = Cells(1, 1)
    Tableau(0, 1) = Cells(1, 2)
    Tableau(0, 2) = Cells(1, 3)
    Tableau(0, 3) = Cells(1, 4)
    Tableau(0, 4) = Cells(1, 5)
    Tableau(1, 0) = Cells(2, 1)
    Tableau(1, 1) = Cells(2, 2)
    Tableau(1, 2) = Cells(2, 3)
    Tableau(1, 3) = Cells(2, 4)
    Tableau(1, 4) = Cells(2, 5)
    Tableau(2, 0) = Cells(3, 1)
    Tableau(2, 1) = Cells(3, 2)
    Tableau(2, 2) = Cells(3, 3)
    Tableau(2, 3) = Cells(3, 4)
    Tableau(2, 4) = Cells(3, 5)
    Tableau(3, 0) = Cells(4, 1)
    Tableau(3, 1) = Cells(4, 2)
    Tableau(3, 2) = Cells(4, 3)
    Tableau(3, 3) = Cells(4, 4)
    Tableau(3, 4) = Cells(4, 5)
    Tableau(4, 0) = Cells(5, 1)
    Tableau(4, 1) = Cells(5, 2)
    Tableau(4, 2) = Cells(5, 3)
    Tableau(4, 3) = Cells(5, 4)
    Tableau(4, 4) = Cells(5, 5)
    Tableau(5, 0) = Cells(6, 1)
    Tableau(5, 1) = Cells(6, 2)
    Tableau(5, 2) = Cells(6, 3)
    Tableau(5, 3) = Cells(6, 4)
    Tableau(5, 4) = Cells(6, 5)

End Sub
```

LES CONDITIONS

L'instruction If

- **Syntaxe**

If condition Then

[Instructions si Vrai]

[ELSE]

[Instructions si Faux]

End If

- Vous pouvez tester, par exemple :

- La valeur des numéros de ligne

- La présence ou non de données dans une cellule

- L'écart de valeur entre deux cellules

- La présence d'une feuille dans un classeur

- ...

Exemple 1

	A	B	C
1	Equipe	Points	Résultat
2	Equipe 1	34	0
3	Equipe 2	31	1
4	Equipe 3	28	1
5	Equipe 4	20	0
6	Equipe 5	25	0
7	Equipe 6	24	1
8	Equipe 7	30	1
9	Equipe 8	22	1
10	Equipe 9	27	0
11	Equipe 10	25	0
12			

- Quand un match est gagné, l'équipe marque 3 points, sinon l'équipe marque 1 point.
- Modifier la colonne B : ajouter 3 quand le résultat est 1 et, 1 sinon.

Exemple 1 – Code (1)

```
Sub Resultat_Match1()
    If Cells(2, 3) = 1 Then
        Cells(2, 2) = Cells(2, 2) + 3
    Else
        Cells(2, 2) = Cells(2, 2) + 1
    End If
End Sub
```

	A	B	C
1	Equipe	Points	Résultat
2	Equipe 1	34	0
3	Equipe 2	31	1
4	Equipe 3	28	1
5	Equipe 4	20	0
6	Equipe 5	25	0
7	Equipe 6	24	1
8	Equipe 7	30	1
9	Equipe 8	22	1
10	Equipe 9	27	0
11	Equipe 10	25	0
12			



	A	B	C
1	Equipe	Points	Résultat
2	Equipe 1	35	0
3	Equipe 2	31	1
4	Equipe 3	28	1
5	Equipe 4	20	0
6	Equipe 5	25	0
7	Equipe 6	24	1
8	Equipe 7	30	1
9	Equipe 8	22	1
10	Equipe 9	27	0
11	Equipe 10	25	0
12			

Exemple 1 – Code (2)

```
Sub Resultat_Match2()
Dim Num_Ligne As Long
    Num_Ligne = 2
    If Cells(Num_Ligne, 3) = 1 Then
        Cells(Num_Ligne, 2) = Cells(Num_Ligne, 2) + 3
    Else
        Cells(Num_Ligne, 2) = Cells(Num_Ligne, 2) + 1
    End If
End Sub
```

Le programme peut être lancé avec différentes valeur de Num_Ligne

→ Faire une boucle... prochaine séance

Exemple 2

	A	B	C
1	Equipe	Points	Résultat
2	Equipe 1	36	0
3	Equipe 2	31	N
4	Equipe 3	28	1
5	Equipe 4	20	N
6	Equipe 5	25	1
7	Equipe 6	24	0
8	Equipe 7	30	1
9	Equipe 8	22	0
10	Equipe 9	27	0
11	Equipe 10	25	1
12			

Un match gagné apporte 3 points, un match nul 2 points et un match perdu 1 point.

Exemple 2 – Code

```
Sub Resultat_Match3()
Dim Num_Ligne As Long
Num_Ligne = 2
If Cells(Num_Ligne, 3) = 1 Then
    Cells(Num_Ligne, 2) = Cells(Num_Ligne, 2) + 3
Else
    If Cells(Num_Ligne, 3) = 0 Then
        Cells(Num_Ligne, 2) = Cells(Num_Ligne, 2) + 1
    Else
        Cells(Num_Ligne, 2) = Cells(Num_Ligne, 2) + 2
    End If
End If
End Sub
```

L'instruction Select Case

- **Syntaxe**

Select Case Expression

 Case Valeur 1

 Instructions

 Case Valeur 2

 Instructions

 Case Valeur..

 Instructions

 Case Else

 Instructions

End Select

Exemple

- On souhaite afficher le nombre de voix obtenues pour 3 candidats. Si le nom du candidat figure en A2, afficher le résultat correspondant en B2.

	A	B	C	D	E
1	Candidat	Nombre de voix			
2	Candidat B		344		
3					
4					Nombre de voix
5					
6					
-					

Exemple (code)

```
Sub Nombre_de_voix()
    Select Case Cells(2, 1)
        Case "Candidat A"
            Cells(2, 2) = 456
        Case "Candidat B"
            Cells(2, 2) = 344
        Case "Candidat C"
            Cells(2, 2) = 675
    End Select
End Sub
```

	A	B
1	Candidat	Nombre de voix
2	Candidat A	456
3		
4		

	A	B
1	Candidat	Nombre de voix
2	Candidat B	344
3		
4		

	A	B	C
1	Candidat	Nombre de voix	
2	Candidat C		675
3			
4			

Test sur plusieurs valeurs non contigües

- Exemple : afficher « nombre premier » en B1 en fonction de la valeur contenue en A1

```
Sub Est_Premier()
    Select Case Cells(1, 1)
        Case 1, 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37
            Cells(1, 2) = "Nombre premier"
        Case Else
            Cells(1, 2) = "Ce n'est pas un nombre premier"
    End Select
End Sub
```

	A	B	C	D
1	3	Nombre premier		
2				
3				
4				
5				
6				
7				
8				
9				
10			Premier ?	
11				

	A	B	C	D
1	6	Ce n'est pas un nombre premier		
2				
3				
4				
5				
6				
7				
8				
9				
10			Premier ?	
11				

Test sur des valeurs contigües

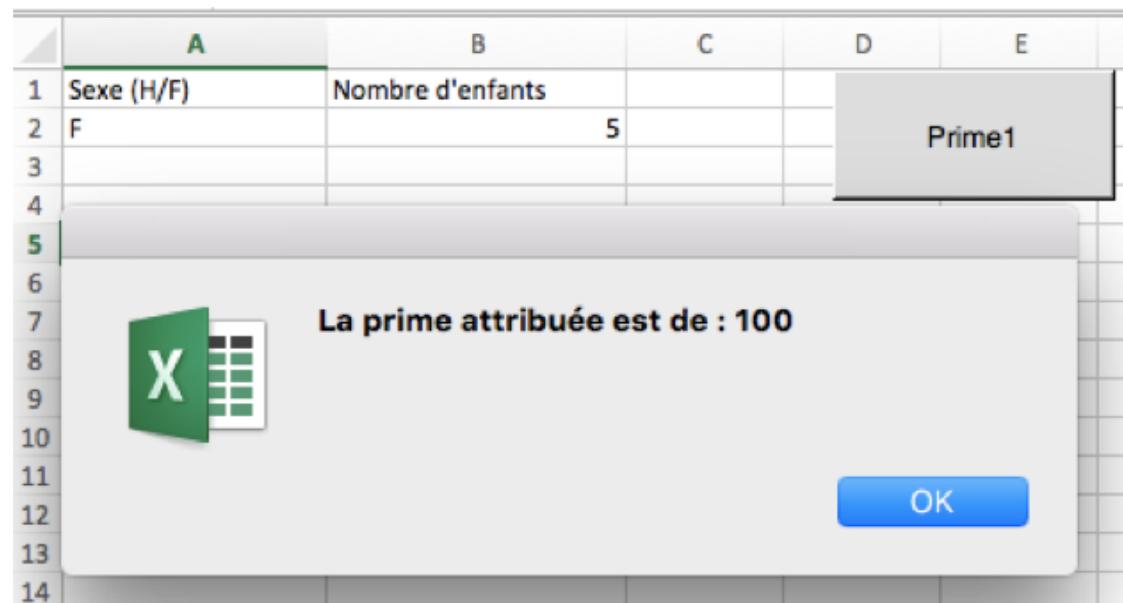
- Exemple : retourner en B1 la mention en fonction de la note contenue en A1

```
Sub Resultat_Examen()
    Select Case Cells(1, 1)
        Case 0 To 9.99
            Cells(1, 2) = "Non reçu"
        Case 10 To 11.99
            Cells(1, 2) = "Reçu"
        Case 12 To 13.99
            Cells(1, 2) = "Reçu - Mention Assez Bien"
        Case 14 To 15.99
            Cells(1, 2) = "Reçu - Mention Bien"
        Case Else
            Cells(1, 2) = "Reçu - Mention Très Bien"
    End Select
End Sub
```

	A	B	C
1	14,5	Reçu - Mention Bien	
2			
3			
4			
5			Résultat
6			
7			
8			
9			

Exercice

- Ecrire un programme qui permet de calculer une prime en fonction des critères suivants :
 - Si la personne est un homme (H) :
$$\text{prime} = 100 + 2 * \text{nombre d'enfants}$$
 - Si la personne est une femme (F) :
$$\text{prime} = 50 + 10 * \text{nombre d'enfants}$$



Code

```
Sub prime1()
Dim sexe As String
Dim nb_enfants As Integer
Dim prime As Integer

sexe = Cells(2, 1)

nb_enfants = Cells(2, 2)

If (sexe = "H" Or sexe = "F") And (nb_enfants >= 0 And nb_enfants <= 10) Then
    If sexe = "H" Then
        prime = 100 + 2 * nb_enfants
    Else
        prime = 50 + 10 * nb_enfants
    End If
    MsgBox ("La prime attribuée est de : " & prime)
Else
    MsgBox ("Votre saisie n'est pas bonne")
End If

End Sub
```

Exercice - suite

- On modifie les critères d'attribution de la prime :

	Fumeur	Non Fumeur
Homme	$100 + 2 * \text{nombre d'enfants}$	100 - 10 par enfant
Femme	50 par enfant	20

The screenshot shows an Excel spreadsheet with data in rows 1 to 15 and columns A to F. Row 1 contains headers: 'Sexe (H/F)' in A1, 'Nombre d'enfants' in B1, and 'Fumeur (O/N)' in C1. Row 2 contains values: 'F' in A2, '2' in B2, and 'O' in C2. A callout box labeled 'Prime 2' points to cell D2, which contains the value '70'. A message box at the bottom left displays the text 'La prime attribuée est de : 70' above an Excel icon. A blue 'OK' button is at the bottom right of the message box.

Code

```
Sub prime2()
    Dim sexe As String
    Dim nb_enfants As Integer
    Dim fumeur As String
    Dim prime As Integer

    sexe = Cells(2, 1)
    nb_enfants = Cells(2, 2)
    fumeur = Cells(2, 3)

    If (sexe = "H" Or sexe = "F") And (nb_enfants >= 0 And nb_enfants <= 10) And (fumeur = "O" Or fumeur = "N") Then
        If sexe = "H" Then
            If fumeur = "O" Then
                prime = 100 + 2 * nb_enfants
            Else
                prime = 100 - 10 * nb_enfants
            End If
        Else
            If fumeur = "O" Then
                prime = 50 * nb_enfants
            Else
                prime = 20
            End If
        End If
        MsgBox ("La prime attribuée est de : " & prime)
    Else
        MsgBox ("Votre saisie n'est pas bonne")
    End If
End Sub|
```

Exercice - suite

- Dernier changement pour les critères :

	Fumeur	Non Fumeur
Homme	$100 + 2 * \text{nombre d'enfants}$	$100 - 10 \text{ par enfant}$
Femme	Le maximum entre 100 et 50 par enfant	20

The screenshot shows a portion of an Excel spreadsheet. Row 1 contains column headers: A (Sexe (H/F)), B (Nombre d'enfants), and C (Fumeur (O/N)). Row 2 contains data: F in A, 1 in B, and O in C. A formula in cell D1 is: =IF(A1="F", 100+2*B1, 100-10*B1). A callout box labeled "Prime 3" points to the formula in cell D1. A message box at the bottom says "La prime attribuée est de : 100". An Excel icon is in the bottom-left corner.

The screenshot shows a portion of an Excel spreadsheet. Row 1 contains column headers: A (Sexe (H/F)), B (Nombre d'enfants), and C (Fumeur (O/N)). Row 2 contains data: F in A, 3 in B, and O in C. A formula in cell D1 is: =IF(A1="F", 100+2*B1, 100-10*B1). A callout box labeled "Prime 3" points to the formula in cell D1. A message box at the bottom says "La prime attribuée est de : 150". An Excel icon is in the bottom-left corner.

Code

```
Sub prime3()
Dim sexe As String
Dim nb_enfants As Integer
Dim fumeur As String
Dim prime As Integer

sexe = Cells(2, 1)
nb_enfants = Cells(2, 2)
fumeur = Cells(2, 3)

If (sexe = "H" Or sexe = "F") And (nb_enfants >= 0 And nb_enfants <= 10) And (fumeur = "O" Or fumeur = "N") Then
    If sexe = "H" Then
        If fumeur = "O" Then
            prime = 100 + 2 * nb_enfants
        Else
            prime = 100 - 10 * nb_enfants
        End If
    Else
        If fumeur = "O" Then
            prime = WorksheetFunction.Max(100, 50 * nb_enfants)
        Else
            prime = 20
        End If
    End If
    MsgBox ("La prime attribuée est de : " & prime)
Else
    MsgBox ("Votre saisie n'est pas bonne")
End If

End Sub
```