Programmation en LibreOffice Basic

Samuel TOUBON

séminaire Ensai



1 / 59

Introduction

- LibreOffice est un fork de OpenOffice.org
- Il hérite de son langage de macros : OpenOffice.org Basic qu'il renomme LibreOffice Basic (LOB).
- Ces langages font partie de la famille Basic, au même titre que VBA, et sont donc très proches.
- LibreOffice, contrairement à Microsoft Office, est ouvert à d'autres langages d'automatisation des tâches : Python, BeanShell, JavaScript, Java, C++.
- Le plus simple reste Basic, il est de plus mieux intégré, avec un éditeur plus sympathique.



2/59

Introduction

LibreOffice Basic peut être divisé en quatre composants :

- Le langage LibreOffice Basic.
- La bibliothèque d'exécution qui fournit des fonctions standard ne faisant pas directement référence à LibreOffice.
- L'API LibreOffice qui permet de manipuler les documents.
- L'éditeur de boîtes de dialogues.

Les deux premiers éléments sont compatibles avec VBA, à quelques détails près.



3/59

Sommaire

1 Le langage LibreOffice Basic

- Généralités
- Variables
- Portée des variables
- Tableaux et matrices
- Opérateurs
- Structures conditionnelles
- Boucles
- Procédures et fonctions

2 Bibliothèque d'exécution de LibreOffice Basic

- Fonctions de conversion
- Manipulations de chaînes
- Manipulations de dates et heures
- Boîtes de message et zones de saisie

3 Manipulation des classeurs

- Classeurs
- Feuilles de calcul
- Lignes et colonnes
- Cellules
- Plages de cellules



4 / 59

Le langage LibreOffice Basic

Idée générale par rapport à Java :

- Seuls des éléments de syntaxe changent, mais il n'y a aucun concept nouveau.
- Dans ce cours, on manipulera des objets issus de l'API, mais on ne définira pas ses propres classes.

LOB:

- est un langage interprété,
- ne peut être utilisé en dehors de LibreOffice.



5 / 59

Lignes de programme

- Cas général : une instruction = une ligne
- Pas de;
- Possibilité d'utiliser le pour écrire une instruction sur plusieurs lignes
- Possibilité d'utiliser le : pour écrire plusieurs instructions par ligne

Commentaires

- Introduits par le symbole ?
- Ou le mot clé Rem



6 / 59

Noms des variables

Noms de variables

- LOB insensible à la casse
- Règles habituelles : pas de caractères spéciaux, le premier caractère doit être une lettre...

Notation hongroise

- Souvent rencontrée dans les langages Basic
- Utilisation non obligatoire
- On rencontrera dans les exemples o (Object), s (String)



Variables

Comportement par défaut

■ Déclaration implicite des variables

1 a = b + c

Combien de variables déclarées? Que valent-elles?



8/59

Le problème :

```
1 Sub Main
2 Dim cellule as Integer
3 cellule = 0
4 celule = cellule + 1
5 MsgBox cellule
6 End Sub
```

Qu'est-ce qui est affiché?



La solution

■ Imposer une déclaration explicite des variables avec Option Explicit.

```
1 Option Explicit
2
3 Sub Main
4 Rem TODO
5 End Sub
```

Option Explicit doit se trouver sur la première ligne de chaque module.



Variables – chaînes de caractères



Variables – numériques

Variables

5 types des bases pour les numériques :

- Integer pour les entiers de 32 768 à 32 767
- Long pour les entiers de 2 147 483 648 à 2 147 483 647
- Single pour les réels positifs ou négatifs 1,401298 \times 10⁻⁴⁵ à 3,402823 \times 10³⁸
- Double pour les réels positifs ou négatifs de 4,94065645841247x10⁻³²⁴ à 1,79769313486232x10³⁰⁸
- Currency pour les valeurs monétaires entre -922 337 203 685 477,5808 et +922 337 203 685 477,5807

Il est possible d'utiliser la notation exponentielle : A = 1.43E2



Variables – autres

Booléens

Il existe un type Boolean qui prend True Ou False.

Si on stocke un entier dans une variable booléenne, celle-ci vaut False si l'entier vaut 0, True dans les autres cas.

Dates et heures

On peut déclarer une variable de type **Date**. Les comparaisons et les opérations arithmétiques entre dates sont alors possibles.

Constantes

On peut déclarer des constantes grâce à la syntaxe suivante :

Const B As Double = 10.

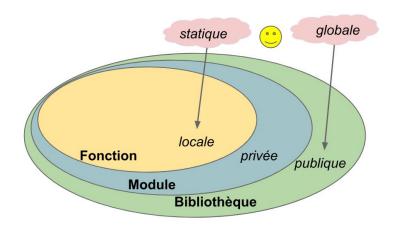


Variables – type Variant

```
1 Dim MyVar As Variant
2 MyVar = "Hello World"
3 MyVar = 1
4 MyVar = 1.0
5 MyVar = True
```

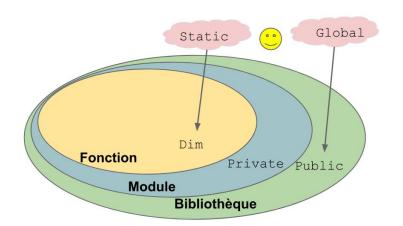


Portée des variables *





Portée des variables *





Tableaux

Dimensions

Dim MyInteger (3) As Integer déclare un tableau de 4 entiers indexés de 0 à 3.

Index

On peut modifier le champ des index à la déclaration :

Dim MyInteger (5 To 10) As Integer. Des bornes négatives sont possibles.

Indice de départ

Si on préfère compter à partir de 1, on peut utiliser Option Base 1. La commande Dim MyInteger(3) As Integer initialise toujours un tableau de 4 entiers dans ce cas, indicés de 1 à 4.

VBA

En VBA, lorsqu'on modifie l'indice de départ, le nombre d'éléments créés n'est pas conservé. Dim MyInteger(3) As Integer initialise un tableau de taille 3. Pour que LibreOffice se comporte comme VBA, on peut utiliser Option Compatible.

le la statistique et de l'analyse de l'information

Matrices

Déclaration

On peut déclarer une matrice de dimension 2 ainsi : Dim MyIntArray(5, 5) As Integer. Celle-ci possède 36 cellules.

Dimensions

Le nombre de dimensions d'une matrice n'est limitée que par la mémoire de la machine.



Redimensionnement

Redimensionnement

On peut redimensionner dynamiquement un tableau ou une matrice grâce à la commande ReDim MyArray(10) As Integer.

Préservation des données

Si l'on souhaite redimensionner un tableau ou une matrice sans écraser les données contenues, il faut utiliser l'option Preserve :

ReDim Preserve MyArray(20) As Integer.

En VBA, seule la limite supérieure de la dernière dimension d'une matrice peut être modifiée en préservant les données.



Fonctions LBound et UBound

Elles permettent de récupérer les indices.

```
1 Dim MyArray(10) As Integer
2 '... des instructions redimensionnent le tableau
3 MsgBox(LBound(MyArray)) ' affiche la borne basse (0)
4 MsgBox(UBound(MyArray)) ' affiche la borne haute (10)
```

En multi-dimensionnel:



Lecture et écriture

```
1 Dim MyArray(10) As Integer
2 MyArray(0) = 57
3 MsgBox(MyArray(0))
```

En multi-dimensionnel:

```
1 Dim MyArray(10, 13 To 28) As Integer
2 MyArray(0,13) = 57
3 MsgBox(MyArray(0,13))
```



Arithmétiques

- +: nombres, dates, chaînes
- : idem
- *: nombres
- /: nombres
- î : nombres (élévation à la puissance)
- Mod : nombres (opération modulo)

Logiques

- And : et
- Or: ou
- Xor : ou exclusif
- Not : négation
- Eqv : les deux éléments ont la valeur True ou False
- Imp: si la première expression est vraie, alors la seconde l'est également

a statistique e l'analyse 'information

22 / 59

Comparaison

- = : nombres, dates, chaînes
- <> : idem
- < : idem</p>
- <= : idem</p>
- > : idem
- >= : idem

VBA

VBA introduit l'opérateur de comparaison Like qui n'est pas disponible en LOB.



If... Then... Else

```
1 If A = 0 Then
2   B = 0
3 ElseIf A < 3 Then
4   B = 1
5 Else
6   B = 2
7 End If</pre>
```



Select... Case

```
1 Select Case DayOfWeek
2
    Case 1:
      NameOfWeekday = "Sunday"
3
    Case 2:
4
      NameOfWeekday = "Monday"
5
    Case 3:
6
7
      NameOfWeekday = "Tuesday"
    Case 4:
Я
      NameOfWeekday = "Wednesday"
9
    Case 5:
10
      NameOfWeekday = "Thursday"
11
    Case 6:
12
13
      NameOfWeekday = "Friday"
    Case 7:
14
      NameOfWeekday = "Saturday"
15
16 End Select
```



Samuel Toubon LibreOffice Basic séminaire Ensai 25 / 59

Select... Case

```
1 Select Case Var
2 Case 1 To 5
3 '... Var is between the numbers 1 and 5
4 Case 6, 7, 8
5 '... Var is 6, 7 or 8
6 Case > 8
7 '... Var is greater than 8
8 Case Else
9 '... all other instances
10 End Select
```



For... Next

```
1 Dim I
2 For I = 1 To 10
3 ' ... Inner part of loop
4 Next I
```

```
1 Dim I
2 For I = 1 To 10 Step 0.5
3 ' ... Inner part of loop
4 Next I
```

```
1 Dim I
2 For I = 10 To 1 Step -1
3 ' ... Inner part of loop
4 Next I
```



For Each

Boucles

```
1 Const d1 = 2
2 Const d2 = 3
3 Const d3 = 2
4 Dim a(d1, d2, d3)
5 For Each i In a()
6 ' ... Inner part of loop
7 Next i
```



Do... Loop

Quelle ménagerie!

En LOB, il existe cinq façons d'écrire une boucle While. En voici deux.



Procédures et fonctions

Procédures

```
1 Sub Test
```

- 2 ' ... here is the actual code of the procedure
- 3 End Sub



Fonctions

```
1 Function Test ' cette fonction retourne un Variant
2  Test = 12
3  ' ...
4  Test = 123 ' valeur retournee
5  End Function
```

```
1 Function Test As Integer
```

- 2 ' ... here is the actual code of the function
- a End Function



Paramètres facultatifs (=optionnels)

```
Sub Test(A As Integer, Optional B As Integer)
    Dim B_Local As Integer
2
    ' Check whether B parameter is actually present
3
    If Not IsMissing (B) Then
     B_Local = B , B parameter present
5
    Else
6
     B_Local = 0 'B parameter missing -> default value 0
7
    End If
8
    ' ... Start the actual function
10 End Sub
```

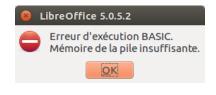
Le mot-clé ParamArray proposé par VBA n'est pas pris en charge en LOB.



Récursivité

Récursivité

- L'appel récursif des fonctions est possible en LOB.
- La taille de la pile dépend du système d'exploitation et de la machine. À l'Ensai, quelque part entre 1 500 et 2 000, avec un plantage en cas de dépassement ¹.
- Sur un PC avec Windows 7, 8 Go de RAM, LibreOffice 5.2, quelque part entre 5 000 et 6 000, avec un message d'erreur poli.
- Sur un PC avec Ubuntu 15.10, 2 Go de RAM, LibreOffice 5.0, quelque part entre 9 000 et 9 500, avec un message d'erreur poli.





33 / 59

C'était vrai en juin 2017 avec LibreOffice 4, à retester maintenant.

Récursivité

```
Sub Main
     MsgBox(Test(9500)) 'entre 9 000 et 9 500
2
 End Sub
4
 Function Test(A As Integer)
     If A>O Then
6
        Test = 1 + Test(A-1)
7
     Else
8
        Test = 0
9
     End IF
10
  End Function
```



Introduction Le langage LibreOffice Basic Bibliothèque d'exécution de LibreOffice Basic Manipulation des classeurs Conclusion

Sommaire

1 Le langage LibreOffice Basic

- Généralités
- Variables
- Portée des variables
- Tableaux et matrices
- Opérateurs
- Structures conditionnelles
- Boucles
- Procédures et fonctions

2 Bibliothèque d'exécution de LibreOffice Basic

- Fonctions de conversion
- Manipulations de chaînes
- Manipulations de dates et heures
- Boîtes de message et zones de saisie

3 Manipulation des classeurs

- Classeurs
- Feuilles de calcul
- Lignes et colonnes
- Cellules
- Plages de cellules



Conversion implicite

```
Dim A As String
Dim B As Integer
Dim C As Integer
A = B + C
```

Que vaut A?



Conversion explicite

- CStr(Var) : convertit tout type de données en chaîne de caractères.
- CInt(Var) : convertit tout type de données en valeur entière.
- CLng(Var) : convertit tout type de données en valeur longue.
- CSng(Var) : convertit tout type de données en valeur simple.
- CDbl(Var) : convertit tout type de données en valeur double.
- CBool (Var) : convertit tout type de données en valeur booléenne.
- CDate(Var) : convertit tout type de données en date.



Samuel TOUBON LibreOffice Basic séminaire Ensai 37 / 59

Fonctions de test

- IsNumeric(Value): détermine si une valeur est un nombre.
- IsDate(Value): détermine si une valeur est une date.
- IsArray(Value) : détermine si une valeur une liste.

Booléens

Il n'existe pas de fonction similaire pour les booléens, mais vous pouvez la construire vous-mêmes.



Samuel TOUBON LibreOffice Basic séminaire Ensai 38 / 59

Cas limite

Conversion impossible

- Si une chaîne contenant une valeur non numérique est assignée à un nombre, LOB ne génère pas d'erreur, mais transmet la valeur 0 à la variable.
- Ceci est encore vrai si l'on utilise la fonction de conversion explicite.

VBA aurait généré une erreur.



Samuel TOUBON LibreOffice Basic séminaire Ensai 39 / 59

Introduction Le langage LibreOffice Basic

Accès à une partie de la chaîne

```
1 Dim MyString As String
2 Dim MyResult As String
3 Dim MyLen As Integer
4
5 MyString = "This is a small test"
6 MyResult = Left(MyString,5)
                                    ' Provides the string "This "
7 MyResult = Right(MyString, 5)
                                    ' Provides the string " test"
8 MyResult = Mid(MyString, 8, 5)
                                    ' Provides the string " a sm"
9 MyLen = Len(MyString)
                                    ' Provides the value 21
```



Samuel TOURON LibreOffice Basic séminaire Ensai 40 / 59

Assignation des dates

Formalisme conseillé pour les dates (année, mois, jour) :

```
Dim MyVar As Date
MyVar = DateSerial (2001, 1, 24)
```

Pour les temps (heure, minute, seconde):

```
Dim MyVar As Date
MvVar = TimeSerial(11, 23, 45)
```

Autre formalisme possible :

```
Dim MyDate As Date
MyDate = "24.1.2002"
```



Samuel TOUBON LibreOffice Basic séminaire Ensai 41 / 59

- Day (MyDate) : retourne le jour du mois de MyDate.
- Month(MyDate): retourne le mois de MyDate.
- Year (MyDate) : retourne l'année de MyDate.
- Weekday (MyDate) : retourne le numéro du jour de la semaine de MyDate.
- Hour (MyTime) : retourne les heures de MyTime.
- Minute(MyTime): retourne les minutes de MyTime.
- Second (MyTime) : retourne les secondes de MyTime.

Heure et date du système :

- Date(): retourne la date actuelle.
- Time(): retourne l'heure actuelle.
- Now() : retourne le point présent dans le temps (la date et l'heure combinées dans une seule valeur).



42 / 59

Samuel TOUBON LibreOffice Basic séminaire Ensai

Formatage des dates

```
Sub Main
     Dim myDate as Date
2
     myDate = "01/06/98"
3
     TestStr = Format(myDate, "mm-dd-yyyy") , 01-06-1998
4
     MsgBox TestStr
 End Sub
```



Samuel TOUBON LibreOffice Basic séminaire Ensai 43 / 59

Affichage des messages

```
1 MsgBox "This is a piece of information!"
2 MsgBox "This is a piece of information!", MB_OK

1 MsgBox "Do you want to continue?", 292
2 MsgBox "Do you want to continue?", _
3 MB_YESNO + MB_DEFBUTTON2 + MB_ICONQUESTION

1 MsgBox "Do you want to continue?", 292, "Box Title"
```



Samuel TOUBON LibreOffice Basic séminaire Ensai 44 / 59

Affichage des messages

- 0, MB_OK : bouton OK;
- 1, MB OKCANCEL : boutons OK et Annuler;
- 2, MB ABORTRETRYIGNORE : boutons Abandonner, Réessayer et Ignorer ;
- 3, MB YESNOCANCEL: boutons Oui, Non et Annuler;
- 4, MB YESNO : boutons Oui et Non ;
- 5, MB_RETRYCANCEL : boutons Réessayer et Annuler.
- 0, MB DEFBUTTON1 : le premier bouton est sélectionné par défaut ;
- 256, MB_DEFBUTTON2 : le deuxième bouton est sélectionné par défaut ;
- 512, MB_DEFBUTTON3 : le troisième bouton est sélectionné par défaut.
- 16, MB_ICONSTOP : signe stop;
- 32, MB_ICONQUESTION : point d'interrogation ;
- 48, MB ICONEXCLAMATION : point d'exclamation ;
- 64, MB_ICONINFORMATION : icône Astuce.



Samuel TOUBON LibreOffice Basic séminaire Ensai 45 / 59

Sommaire

1 Le langage LibreOffice Basic

- Généralités
- Variables
- Portée des variables
- Tableaux et matrices
- Opérateurs
- Structures conditionnelles
- Boucles
- Procédures et fonctions

2 Bibliothèque d'exécution de LibreOffice Basic

- Fonctions de conversion
- Manipulations de chaînes
- Manipulations de dates et heures
- Boîtes de message et zones de saisie

3 Manipulation des classeurs

- Classeurs
- Feuilles de calcul
- Lignes et colonnes
- Cellules
- Plages de cellules



Samuel TOUBON LibreOffice Basic séminaire Ensai 46 / 59

Avertissement

- Cette partie n'est en aucun cas exhaustive. Elle vous présente quelques fonctions et quelques modèles pour que vous puissiez commencer rapidement.
- Vous pourrez vous référer à la documentation (cf. bibliographie), votre moteur de recherche favori, stackoverflow.com et toute source pertinente pour obtenir des informations plus complètes.
- Lorsque c'est nécessaire, le TP vous soufflera les parties de documentation les plus pertinentes.



Samuel TOUBON LibreOffice Basic séminaire Ensai 47 / 59

Accès au classeur

Classeur contenant la macro :

```
Dim oDoc As Object
2 oDoc = ThisComponent
```

Le composant actif (ce peut être un classeur ou non) :

```
Dim oDoc As Object
2 oDoc = CurrentComponent
```



Samuel TOUBON LibreOffice Basic séminaire Ensai 48 / 59

Accès à la feuille de calcul

```
Par son numéro :
```

Introduction Le langage LibreOffice Basic

```
Dim Doc As Object
 Dim Sheet As Object
3
4 Doc = ThisComponent
 Sheet = Doc.Sheets (0)
```

Par son nom:

```
Dim Doc As Object
 Dim Sheet As Object
3
 Doc = ThisComponent
 Sheet = Doc.Sheets.getByName("Sheet 1")
```



Samuel TOUBON LibreOffice Basic séminaire Ensai 49 / 59

Accès à la feuille de calcul

Feuille active :

Introduction Le langage LibreOffice Basic

ThisComponent.getCurrentController.getActiveSheet

Parcours des feuilles :

Samuel TOUBON LibreOffice Basic séminaire Ensai

50 / 59

Création d'une feuille de calcul

```
Dim Doc As Object
 Dim Sheet As Object
3
4 Doc = StarDesktop.CurrentComponent
 Sheet = Doc. Sheets(0)
6
  If Doc.Sheets.hasByName("MySheet") Then
     Sheet = Doc.Sheets.getByName("MySheet")
8
 Else
     Sheet = Doc.createInstance("com.sun.star.sheet.Spreadsheet")
10
     Doc. Sheets.insertByName("MySheet", Sheet)
11
  End If
```



Samuel TOUBON LibreOffice Basic séminaire Ensai 51 / 59

Création et suppression d'une feuille de calcul

On peut aussi utiliser insertNewByName dont le deuxième argument est la position :

```
Doc. Sheets.insertNewByName("MySheet", 2)
```

La suppression peut se faire ainsi :

```
Doc.Sheets.removeByName("MySheet")
```



Samuel TOUBON LibreOffice Basic séminaire Ensai 52 / 59

Sélection de lignes ou colonnes

```
1 Dim Doc As Object
2 Dim Sheet As Object
3 Dim FirstRow As Object
4 Dim FirstCol As Object
5
6 Doc = StarDesktop.CurrentComponent
7 Sheet = Doc.Sheets(0)
8
9 FirstCol = Sheet.Columns(0)
0 FirstRow = Sheet.Rows(0)
```



Samuel TOUBON LibreOffice Basic séminaire Ensai 53 / 59

Insertion de lignes ou colonnes

```
1 Dim Doc As Object
2 Dim Sheet As Object
 Dim NewColumn As Object
4
5 Doc = StarDesktop.CurrentComponent
Sheet = Doc.Sheets(0)
7
 Sheet.Columns.insertByIndex(3, 1) 'insert 1 colonne en 4e pos.
 Sheet.Columns.removeByIndex(5, 1) 'efface la colonne 6
```



Samuel TOUBON LibreOffice Basic séminaire Ensai 54 / 59

Sélection de cellule

Cellules

```
1 Dim Doc As Object
2 Dim Sheet As Object
3 Dim Cell As Object
4
5 Doc = StarDesktop.CurrentComponent
6 Sheet = Doc.Sheets(0)
7
8 Cell = Sheet.getCellByPosition(0, 0)
9 Cell.String = "Test"
```

Ou bien:

```
1 Cell = Sheet.getCellRangeByName("A1")
```



Samuel TOUBON LibreOffice Basic séminaire Ensai 55 / 59

Contenu d'une cellule

```
1 Cell = Sheet.getCellByPosition(0, 0)
2 Cell.Value = 100
3
4 Cell = Sheet.getCellByPosition(0, 1)
5 Cell.String = "Test"
6
7 Cell = Sheet.getCellByPosition(0, 2)
8 Cell.Formula = "=A1"
```



Samuel TOUBON LibreOffice Basic séminaire Ensai 56 / 59

Sélection de plage de cellules

```
oRange = oSheet.getCellRangeByName( "B2:C3" )
```

Ou bien:

```
oRange = oSheet.getCellRangeByPosition(1,1,2,2)
```

L'ordre est left, top, right, bottom.



Samuel TOUBON LibreOffice Basic séminaire Ensai 57 / 59 Introduction Le langage LibreOffice Basic

```
1 Sheet = Doc.Sheets.getByName("Sheet 1")
2 CellRange = Sheet.getCellRangeByName("A1:C3")
3 MsgBox CellRange.computeFunction(com.sun.star.
4 sheet.GeneralFunction.AVERAGE)
```

- SUM : somme de toutes les valeurs numériques.
- COUNT : nombre total de valeurs (y compris les valeurs non numériques).
- COUNTNUMS : nombre total de valeurs numériques.
- AVERAGE : moyenne de toutes les valeurs numériques.
- MAX : valeur numérique la plus élevée.
- MIN : valeur numérique la plus petite.
- PRODUCT : produit de toutes les valeurs numériques.
- STDEV : écart-type.
- VAR : variance.
- STDEVP : écart-type calculé sur la base de la population totale.
- VARP : variance calculée sur la base de la population totale.



École nationale de la statistique et de l'analyse de l'informatio

58 / 59

Nous verrons au TP4 comment utiliser n'importe quelle fonction de LibreOffice pour faire des calculs sur les plages, et pas seulement les précédentes.

Bibliographie

- Guide de programmation de OpenOffice.org BASIC: https://wiki.openoffice.org/wiki/FR/Documentation/BASIC_Guide
- Le même en anglais (plus complet, moins d'erreurs): https://wiki.openoffice.org/wiki/Documentation/BASIC_Guide
- Mémento sur la manipulation des feuilles de calcul : https://wiki.openoffice.org/wiki/Spreadsheet_common



Samuel TOUBON LibreOffice Basic séminaire Ensai 59 / 59