VBA / Excel

M2 IF Apprentissage

Julien Saunier
saunier@inrets.fr
http://www.lamsade.dauphine.fr/~saunier/m2if/

Sommaire

- VB? VB.NET? VBA? Excel?
- Un language procédural...
- ... qui utilise des objets!
- Outils Mathématiques

Historique Visual Basic

- Sortie de VB1 en 1991
  - Language interprété
- VB5 (1997)
  - Possibilité de créer des exécutables
- VB6 (1998)
  - Fin de cycle
- VB.NET

Visual Basic

- Faiblesses:
  - Performance
  - Gestion des erreurs
  - Simplicité
  - Procédural (pas d'héritages)
- Avantages:
  - Simplicité
  - Lien avec les applications Office

Le cas VB.NET

- Introduit en 2003 dans le cadre du Framework .NET (avec C# et ASP.NET)
- Language entièrement orienté objet
- Pas de rétro-compatibilité
- Pas de lien avec Office

Le cas VB.NET

- Classic VB example:

```
Private Sub Command1_Click()
    MsgBox "Hello, World"
End Sub
```

- A VB.NET example:

```
Private Sub Button1_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button1.Click
    MessageBox.Show("Hello, World")
End Sub
```

VBA:
Visual Basic for Applications

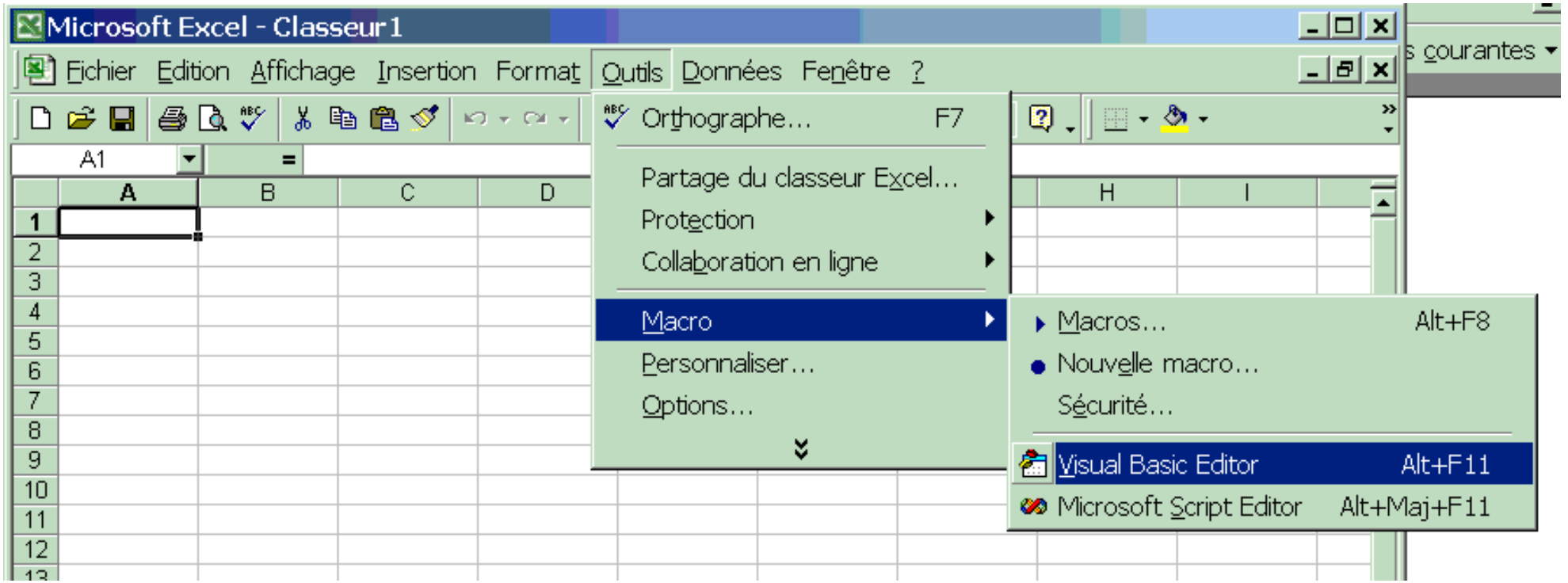
- Disponible dans toutes les applications Office
- Compatible avec VB (jusqu'à la version 6)
- Mêmes avantages et faiblesses que VB!
- Futur Incertain:
  - arrêt du développement et du support général
  - VSTA (Visual Studio Tools for Applications)?
- StarBasic, disponible avec la suite bureautique (libre) OpenOffice.org: Compatibilité non assurée

VBA (suite)

- Macro = Programme
- Interpreté
- Code éditable de deux façons:
  - Enregistrer les actions
  - Créer / Modifier du code dans Visual Basic Editor (VBE)
- VBA n'est pas Excel!
  - > Des fonctions de même nom peuvent ne pas avoir le même effet.

Utilisation pratique de VB dans Excel

- Menu Outils -> Macro -> Visual Basic Editor



Un language Procédural...

Exemple

VBA

```
Public Sub afficher_texte()
    Dim i As Integer
    For i=1 To 2
        MsgBox "Hello World !"
    Next i
End Sub
```

C/C++

```
int main(int argc, char * argv[]){
    int i = 0;
    for(i=0; i<2; i++){
        printf("Hello World !\n");
    }
}
```

VBA

Les sous-programmes (procédures et fonctions) sont regroupés en « modules »

Pas de programme principal

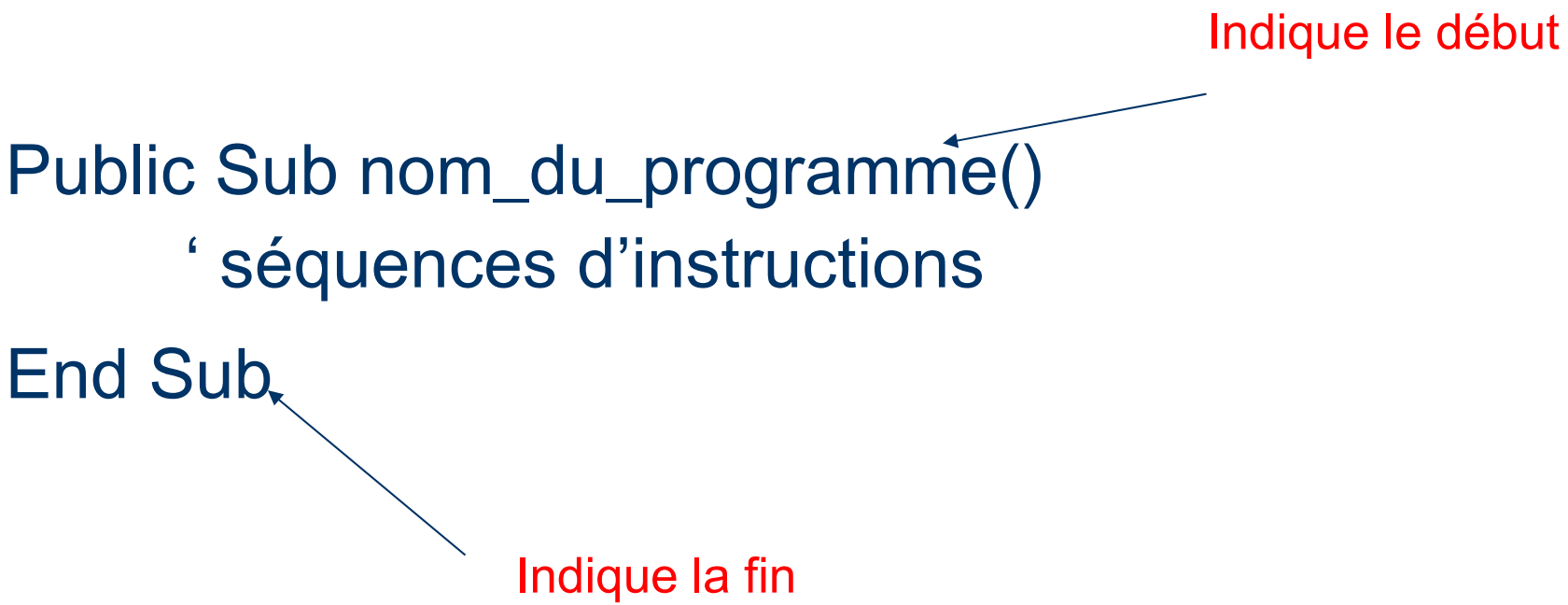
Les sous-programmes sont « lancés »

- **Par un autre sous-programme**
- **Parce qu'il se passe quelque chose dans l'application**

Une programmation dite « événementielle »

Syntaxe de base

- Il faut toujours indiquer où se trouve le début et la fin du programme que l'on écrit.



Variables

- En Visual Basic, on considère env. 12 types de variables. Les plus utilisées:
  - String pour stocker des chaînes de caractères
  - Integer pour stocker des valeurs entières
  - Double pour stocker des valeurs décimales
  - Long pour stocker des grandes valeurs entières
  - Boolean pour stocker soit un 0 soit un 1 (un bit)

Variables

- En VB, il n'est pas obligatoire de déclarer les variables, mais vivement conseillé
- Syntaxe de déclaration:

```
Public Sub prog_qui_fait_rien()
        Dim age As Integer
        Dim nom As String
        Dim revenu As Long
End Sub
```

Danger

Le type Variant

La déclaration automatique

Une garantie : commencer le module par

Option Explicit

-> Obligation de déclarer les variables

Opérateurs et variables

- Nombreux opérateurs: + * / - & Mod \ ^
- Les opérateurs font des opérations avec (sur) des variables

```
Public Sub mon_prog ()
Dim a As Integer, b As Integer, c As Integer, d As Integer
a = 5
b = 7
c = a + b
d = a * b
End Sub
```

Tableaux

- Deux déclarations possibles:

1. Dim tabl1(0 To 12) As Integer
   - On prévoit 13 emplacements pour des entiers
   - tabl1(0) = 14
   - tabl1(2) = 17
2. Dim tabl2 As Variant
   - Souvent lorsqu'on utilise la fonction Array
   - tabl2 = Array("Janvier", "Février", "Mars")

Tableaux

- Structure pour afficher le contenu:

```
Dim mois As Variant, m As Variant
mois = Array("Janvier", "Mars", "Août", "Décembre")

For Each m In mois
        MsgBox m
Next m
```

- Ou

```
Dim mois As Variant, i As Integer
mois = Array("Janvier", "Mars", "Août", "Décembre")

For i = 0 To 3
        MsgBox mois(i)
Next i
```

Tableaux

- Fonctions sur les tableaux:
  - LBound: plus petit index du tableau
  - UBound: plus grand index
  - Array(…): retourne un tabeau (doit être affecté à un Variant)
  - Erase efface le tableau de la mémoire

Instructions conditionnelles

- Syntaxe en VB:

```
Dim a As Integer, b As Integer
a = 5
If a < 10 Then
        b = 1
Else
        b = 2
End If
```

Boucles - For

- Syntaxe:

```
Dim a As Integer

For a = val1 To val2


Next a
```

Boucles - While

- Différence avec For: on ne connaît pas toujours la fin de la boucle à l'avance

```
Dim a As Integer
a = val1
While  a < val2
        …
        a = a + 1
Wend
```

Entrée Sorties

- Pas d'execution dans un terminal
  - > Pas d'entrées sorties équivalentes aux languages classiques
- Deux Possibilités principales:
  - Lecture et écriture dans des cellules d'Excel *(Partie Excel)*
  - Utilisation de boites « pop-ups »

Ecriture: Msgbox

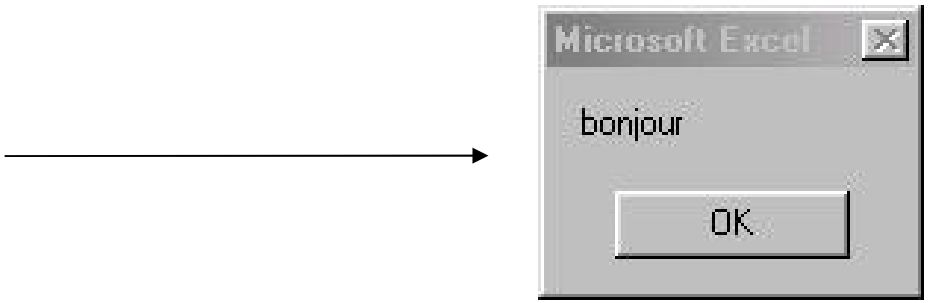
Permet d'afficher un message dans une boîte de dialogue

Syntaxe :

msgbox("*votre message*")

Exemple :

msgbox("bonjour")



Lecture: Inputbox

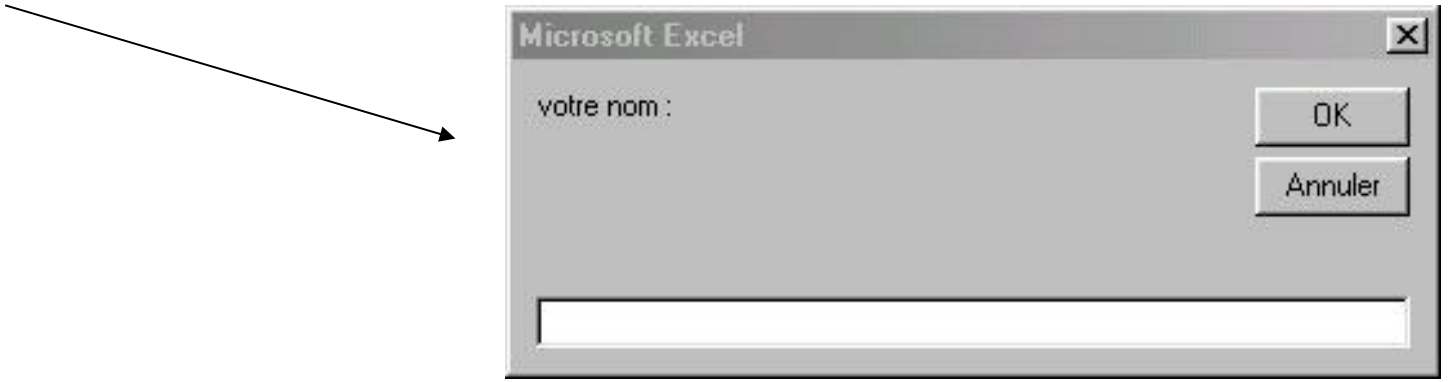
Permet de demander une information à l'aide d'une boîte de dialogue

Syntaxe :

variable = Inputbox("*votre message*")

Exemple :

Nom = InputBox("votre nom :")



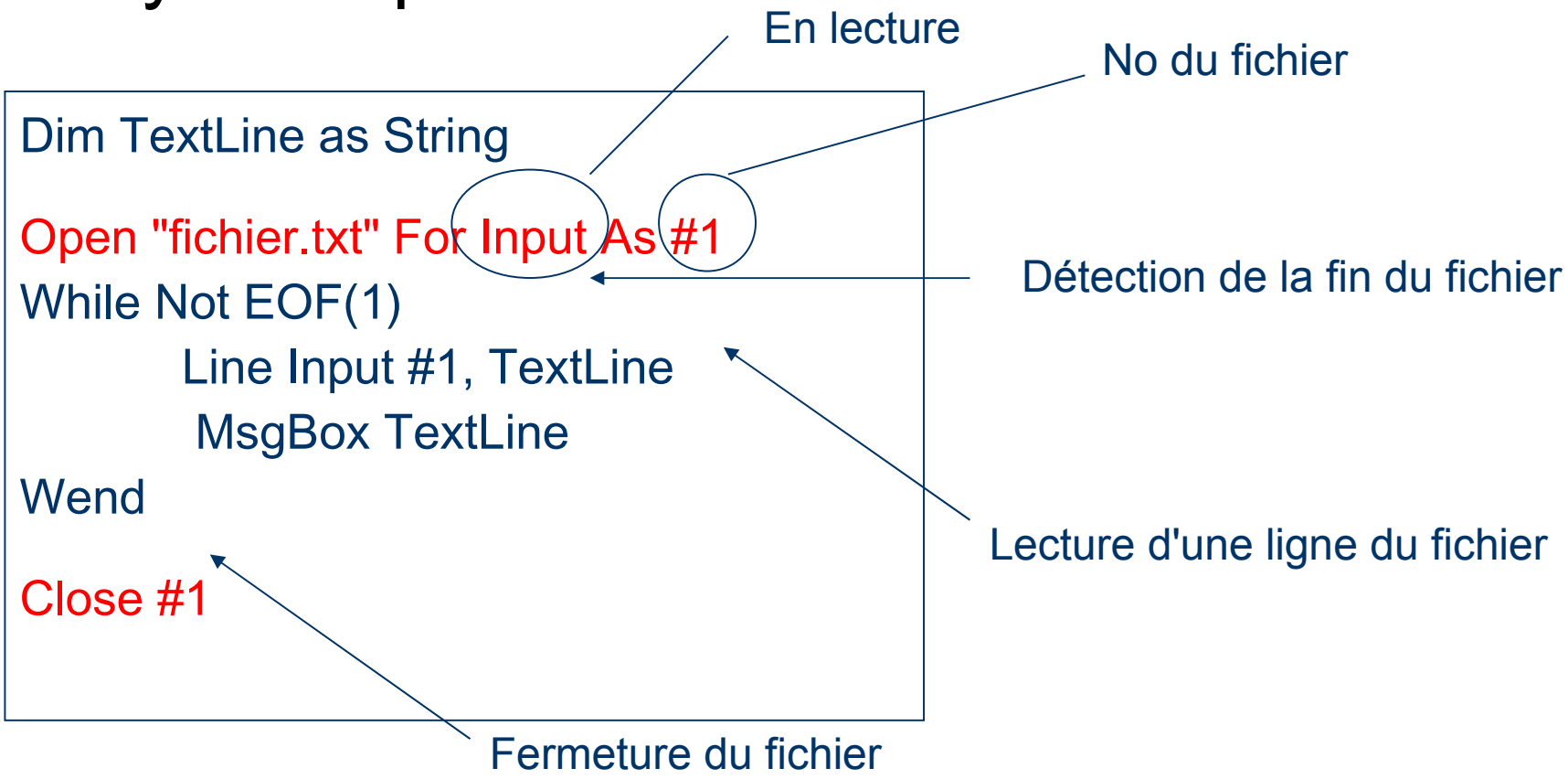
Accès au système de fichiers (1)

- Une façon de « communiquer » avec d'autres languages!

- En Visual Basic, on peut:
  - Créer des fichiers
  - Compléter des fichiers
  - Effacer des fichiers
  - Lire des fichiers
  - Créer et supprimer des répertoires

Accès au système de fichiers (2)

- Syntaxe pour la lecture d'un fichier:



Accès au système de fichiers (3)

- Syntaxe pour l'écriture d'un fichier:

```
Public Sub ecrirefichier()

    Dim liste As Integer
    liste = 0
    Open "cible.txt" For Output As #1
    Do While liste < 100
        liste = liste + 1
        Print #1, liste
    Loop
    Close #1

End Sub
```

En écriture

Ecrit une ligne dans le fichier

Fonctions mathématiques

- Valeur absolue: Abs(-9) retourne 9
- Signe: Sgn(-18) retourne –1 (ou 0 ou 1)
- Partie entière: Fix(-18.3) = -18
  Fix(18.3) = 18
  - Enlève la partie décimale

Fonctions mathématiques

- Sqr, Exp, Log
  - Sqr(4) retourne 2, Exp(5) retourne 148.413…,
    Log(9) retourne 2.197224… (en base e)
- Nombres aléatoires
  - Rnd retourne un nombre aléatoire entre 0 (compris) et 1 (non compris)
  - a = Rnd
- Sin, Cos, Tan, Atn (arc-tangente)

Fonctions

- Syntaxe:

```
Public Function nom_fonct (variable As String) As Integer
    --- instructions
    'valeur de retour:
    nom_fonct=val
End Function
```

- Appel:

```
b= nom_fonct("test")
call nom_fonct(param)
```

Exercice (1)

La valeur future au temps T d'un bien sans rendement est de:

$$f_t = e^{r(T-t)} S^t$$

Avec S le prix actuel du bien, t le temps actuel et r le taux d'intérêt

- Ecrire un Programme:
  - Qui contient une fonction de calcul de la valeur future d'un bien
  - Qui teste cette fonction pour connaître la valeur dans 15 mois d'un bien valant aujourd'hui 1200000€ avec un taux d'intérêt de 4% par an

Passage par valeur, Passage par référence

- Par défaut, le passage de paramètre se fait par référence
- Pour forcer le passage par valeur, écrire le mot-clé ByVal dans la déclaration de fonction:
  Public Function Toto(ByVal a As Integer)

- Le mot-clé ByRef indique le passage par référence

Un language procédural...
... qui utilise des objets!

VBA et Objets

- Ce sont les interfaces vers les applications Office
- Ils sont spécifiques à chaque application de la suite:
  - Access: tables, requêtes, formulaires...
  - Excel: Feuilles, cellules...
- Sont parfois des équivalents de librairies

Un objet d'office...

- ... est complexe
- ... a des propriétés
  nom de l'objet, valeur de la zone de texte...
- ... dispose de méthodes
- ... réagit à des événements
  un clic sur l'objet, avant ou après une mise à jour, en arrivant ou en quittant l'objet...

Principaux objets d'Excel

- Application: contient des Workbooks
- Workbook: contient des Worksheets
- Worksheet: contient des Range

Application

- Attributs:
  - WorkBooks
  - Windows
  - Propriétés comme CutCopyMode (booléen)
- Methodes:
  - ActiveCell
  - ActiveSheet
  - Quit
  - ...

Workbooks

- Actions:
  - Close, Add, Count
  - Open filename:= stringval
- Selection:
  - Workbooks(String name)
  - Workbooks(Integer index)
- Actions sur un Workbook:
  - Activate
  - Close
  - ActiveWorksheet

Worksheets

- C'est un attribut d'un WorkBook
- Fonctionne et possède des fonctions identiques à Workbooks
- WorkSheet:
  - Attributs name, visible...
  - Méthodes activate

Range: ambiguité

- Deux syntaxes pour accéder:
  - Range(arg) où *arg* est une chaîne de caractères de la forme CL, par exemple A1
  - Range(cell1, cell2) pour sélectionner une plage de données
- Value et Formula permettent de récupérer ou d'affecter la valeur / formule
- ClearContents
- Cells

Cells

- Attribut de WorkSheet
- Cells(L,C) pour sélectionner la cellure LC
- Pour chaque Cell:
  - Value et Formula
  - Offset(h,v) retourne la cellule située à 3 lignes et 1 colonne
  -

Graphiques: chart

- Set objChart = Charts.Add
  objChart.ChartType = xlXYScatterSmooth
  objChart.Name = "toto"

  objChart.SetSourceData Source:=Range("B2:C8")

  With ActiveChart
  .HasTitle = True
  .ChartTitle.Characters.Text = "allongement en fonction du temps"
  .Axes(xlCategory, xlPrimary).HasTitle = True
  .Axes(xlCategory, xlPrimary).AxisTitle.Characters.Text = "blabla"
  .Axes(xlValue, xlPrimary).HasTitle = True
  .Axes(xlValue, xlPrimary).AxisTitle.Characters.Text = "tata"
  End With

Utiliser les fonctions d'Excel

- Le prémisse est:
  Application.WorksheetFunction
- Auquel on ajoute le nom de fonction:
  Exemple:
  Application.WorksheetFunction.Sum(Range("B2:B157"))

- Verifier ce que font les fonctions!
  - Round(0.5,0) : 0 *VBA*
  - Round(0,5;0) : 1 *Excel*

Ne pas oublier d'utiliser les fonctions d'Excel

- De nombreuses fonctions sont déjà disponibles. Inutile de les recoder en VBA!
- Exemple: matrices et calculs matriciels
  - Initialisation par ={4\5\3;2\2\2;1\8\6}
  - Opérateurs +,-,*,/ applicables entre une matrice et une cellule, une colonne, ou une autre matrice de même taille
  - Fonctions, par exemple MINVERSE(Range)

Exercice (2)

extrait du fichier « exo2.xls », feuille « test »

$$C_u = max(0, S_u - K)\ avec\ S_u = uS_0$$
$$C_d = max(0, S_d - K)\ avec\ S_d = dS_0$$
$$C_0 = e^{-r}(qC_u + (1-q)C_d),\ avec\ q = \frac{e^r - d}{u - d}$$

	A	B	C	D	E	F	G
1						Taux d'int	4,5
2	Nom	Valeur actuel	Option	Eval haute	Eval basse		
3	Action 1	154,02	200,22				
4	Action 2	168,23	218,7				
5	Action 3	161,31	209,7				
6	Action 4	103,71	134,82				
7	Action 5	125,29	162,87				
8	Action 6	53,08	69,01				
	n 7	55,31	71,9				
	n 8	142,13	184,77				
11	Action 9	110,17	143,22				
12	Action 10	93,51	121,56				
13	Action 11	142,18	184,83				
14	Action 12	117,03	152,14				
15	Action 13	139,22	180,98				
16	Total	1565,19	2034,74				
17							
18							
19							
20							
21							
22							
23							
24							
25							

adient Dialog.

- Ecrire une Macro qui:
  - Ouvre la feuille
  - Créée des eval aléatoires:
    - basse: entre 0.3 et 0.9
    - Haute: entre 1.1 et 1.5
  - Calcule le prix des options selon les formules ci dessus pour toutes les actions listées (Binomial Option pricing)