

Les macro-commandes dans Excel



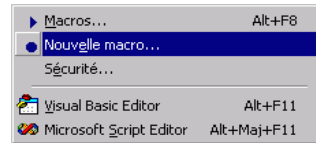
© Yvan Péguiron - HEP-VD - nov. 2006

Définition

- *Une macro, c'est la mise en mémoire d'une instruction ou d'une suite d'instructions de manière à pouvoir l'exécuter automatiquement et à volonté.*
- *Les macros Excel sont programmées en Visual Basic Applications (VBA)...*

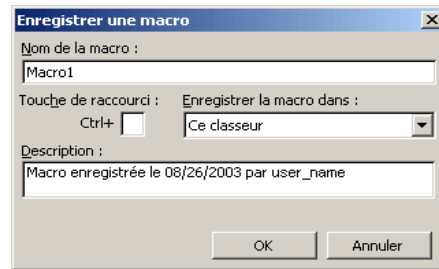
2. Enregistrer une macro

- Dans le menu "Outils", pointez sur "Macro", puis cliquez sur "Nouvelle macro..."



- Dans la zone "Nom de la macro", tapez le nom de la macro. Le nom de la macro doit commencer par une lettre.

Les autres caractères peuvent être des lettres, des chiffres ou le caractère de soulignement. Les espaces ne sont pas autorisés dans un nom de macro, mais le caractère de soulignement peut très bien servir de séparateur de mots.



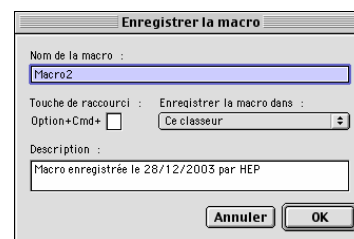
2. Enregistrer une macro

- Dans le menu "Outils", pointez sur "Macro", puis cliquez sur "Nouvelle macro..."



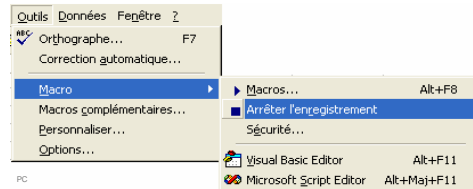
- Dans la zone "Nom de la macro", tapez le nom de la macro. Le nom de la macro doit commencer par une lettre.

Les autres caractères peuvent être des lettres, des chiffres ou le caractère de soulignement. Les espaces ne sont pas autorisés dans un nom de macro, mais le caractère de soulignement peut très bien servir de séparateur de mots.



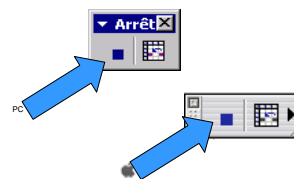
Fin de l'enregistrement

- Dans le menu "Outils", pointez sur "Macro", puis cliquez sur « Arrêter l'enregistrement »



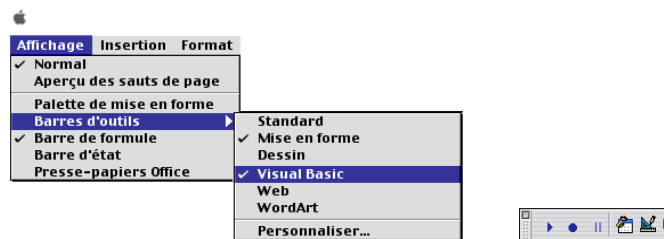
ou

- si la barre de menu ad hoc est affichée, cliquez sur le carré bleu.



Conseil pour gagner du temps

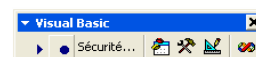
Pour accéder plus rapidement aux boîtes de dialogue utiles :



Sur PC, si vous aimez les raccourcis clavier...

Utilisez les raccourcis clavier suivants :

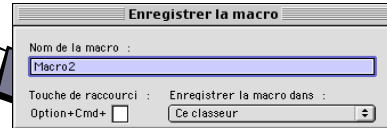
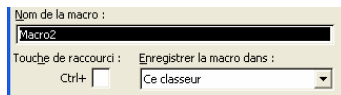
- ALT + O + M + E = enregistrer
- ALT + O + M + R = arrêter
- ALT + O + M + M = modifier
- ALT F8 = créer
- ALT F11 = éditeur VBA



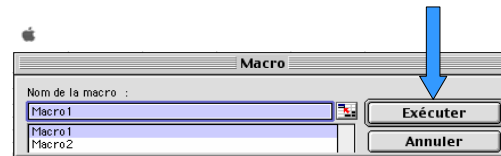
3. Exécuter une macro

Trois possibilités de lancer une macro-commande :

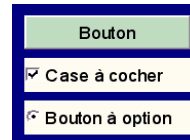
- La touche de raccourci définie à la création : Ctrl +



- Outils - Macro - Macros...



- Utilisation d'un bouton de commande.



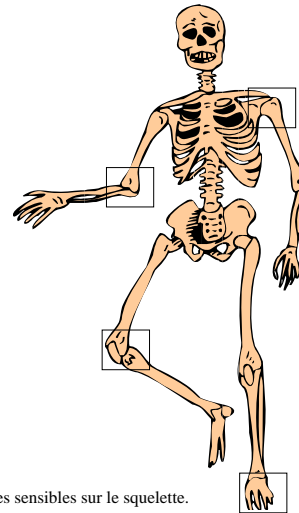
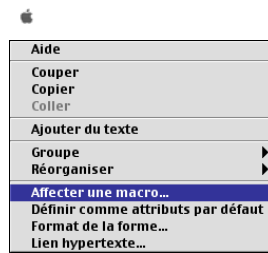
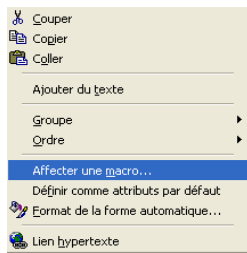
4. Créer des boutons pour lancer les macros

The diagram illustrates the process of creating a button for a macro. It shows the following steps:

- Open the **Outils** menu.
- Select **Barres d'outils**.
- Select **Formulaires**.
- Click on **Personnaliser...** to create a custom toolbar.
- The resulting custom toolbar is shown with a button that can be used to launch macros.

5. Affecter une macro à un objet graphique

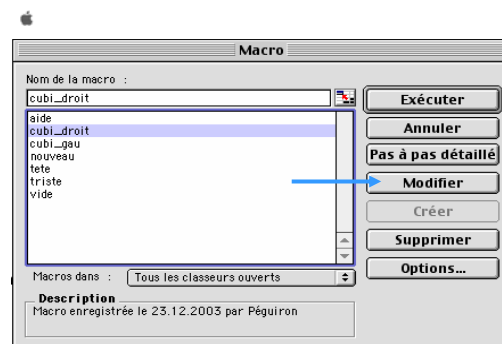
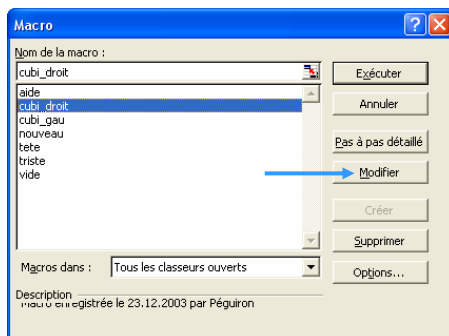
Des objets qui déclenchent des macros ...



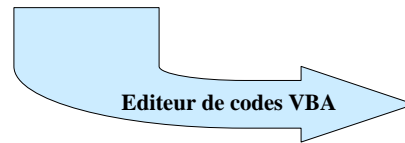
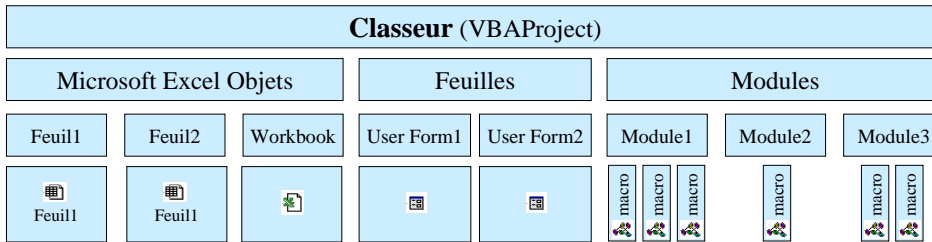
Créer des zones sensibles sur le squelette.

6. Modifier une macro

- Dans le menu "Outils", pointez sur "Macro", puis cliquez sur « Macros... »



Vous arrivez dans le monde envoûtant de Visual Basic !



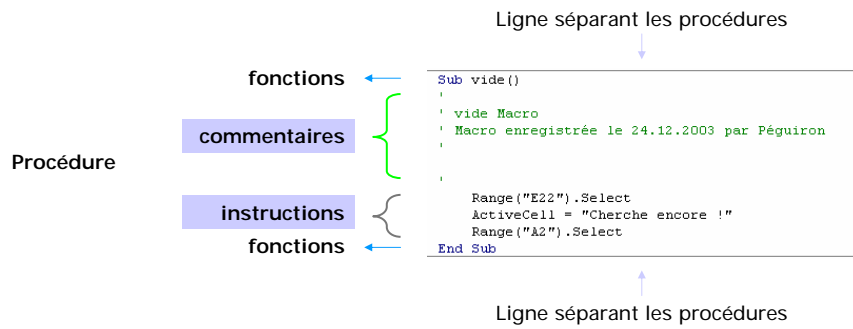
Contenu du classeur

```
Sub cubi_droit()  
' cubi_droit Macro  
' Macro enregistrée le 23.12.2003 par Péguiron  
  
Range("E22").Select  
ActiveCell.FormulaR1C1 = "Non, c'est le droite !"  
Range("A2").Select  
End Sub  
Sub vide()  
' vide Macro  
' Macro enregistrée le 24.12.2003 par Péguiron  
  
Range("E22").Select  
ActiveCell = "Faux !"  
Range("A2").Select  
End Sub
```

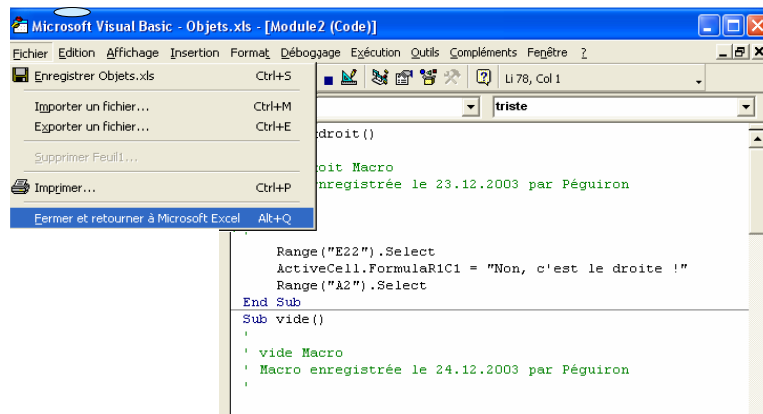
Fenêtre de propriétés

Fenêtre Code

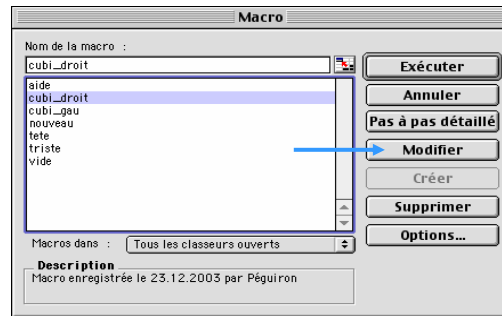
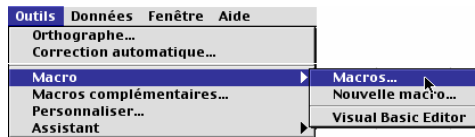
Codage d'une macro (sous-routine)



Pour revenir dans la
feuille Excel



7. Programmer en VBA dans les macros



- Les variables
- Lire et écrire dans les cellules
- Tests
- Boucles
- Fonction aléatoire
- Macros imbriquées

7a) Les variables

Les variables servent à stocker certaines valeurs durant l'exécution d'un programme afin de pouvoir les réutiliser à un autre endroit du programme.

Pour distinguer les différentes valeurs, on donne des noms aux variables.

Exemples :

```
A = 3 * 4
B = "Victor"
Varia1 = A + 10
Varia2 = "Hugo"
Conca = "Hugo " & B
```

Les noms de variables doivent :

- commencer par une lettre
- ne pas contenir plus de 255 caractères
- ne pas contenir les caractères ., ! @ &...
- ne pas être un mot réservé

7a2) Les variables suites

Les variables peuvent être implicites, c'est à dire non déclarées, ou explicites.
Si la première manière est plus simple, la deuxième est recommandée pour éviter les risques d'erreurs liées entre autre à des fautes de frappe.

Variables explicites :

Exemples :

Dim PARDEF As Variant	(double précision ou chaîne)
Dim NOM\$ As String	(chaîne de caractères)
DIM MONNAIE As String * 3	(maximum 3 caractères)
Dim NAISSANCE As Date	(de 01.01.0100 à 31.12.9999)
DIM VRAI_FAUX As Boolean	(True ou False)
DIM PIECE2 As Integer	(entiers de -32'768 à 32'767)
DIM MOYENNE As Currency	(+ - 922'337'203'685'477,5808)
DIM PI As Decimal	(Currency à 14 décimales)

7a3) Les variables matricielles

Les variables à 2 ou 3 dimensions (tableau) doivent être déclarées avec la commande Dim

3 lignes, 5 colonnes

Exemples :

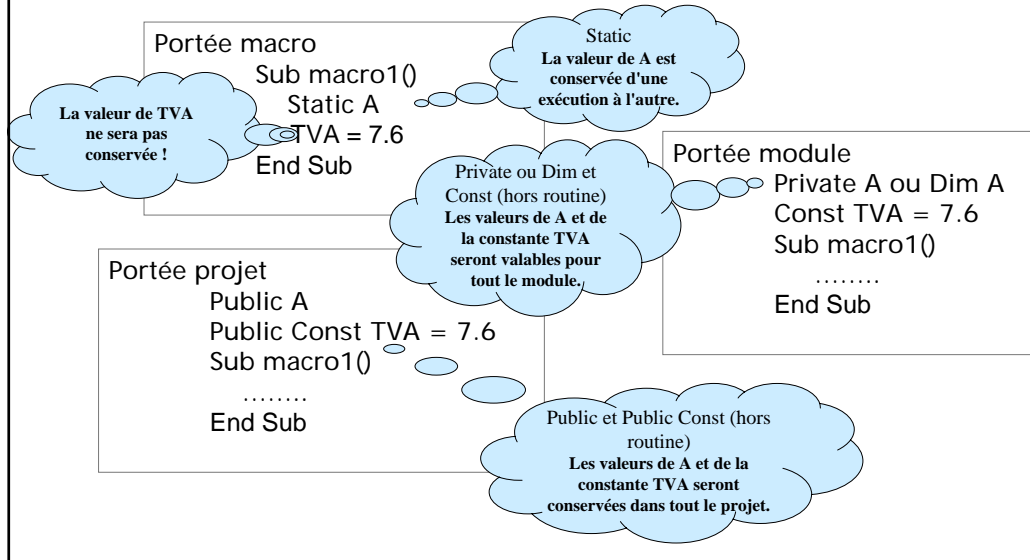
Dim A = (3, 5)
A(2, 3) = "Hello"
A(1,2) = 5
A(3, 5) = "Fin"

	5			
		Hello		
				Fin

7a) Portée des variables et des constantes

Par défaut, les variables (et les constantes) ne sont valables que dans la macro où elles ont été définies (Dim A) et leur valeur est remise à zéro à chaque exécution.

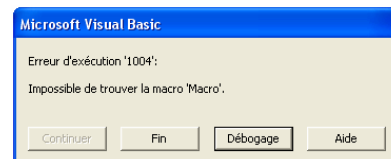
On peut conserver la valeur des variables et des constantes à 3 niveaux :



7b) Débogage

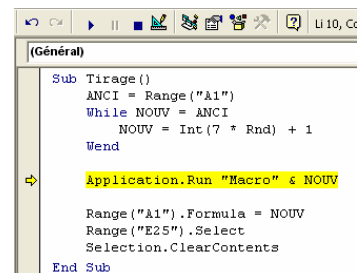
Il arrivera inévitablement que des erreurs de programmation surviennent lors de l'élaboration d'une macro...

- Erreurs de compilation (instruction inconnue).
- Erreurs d'exécution (données incompatibles comme une opération arithmétique sur une chaîne).
- Erreurs logiques (attention, la macro s'exécute normalement).

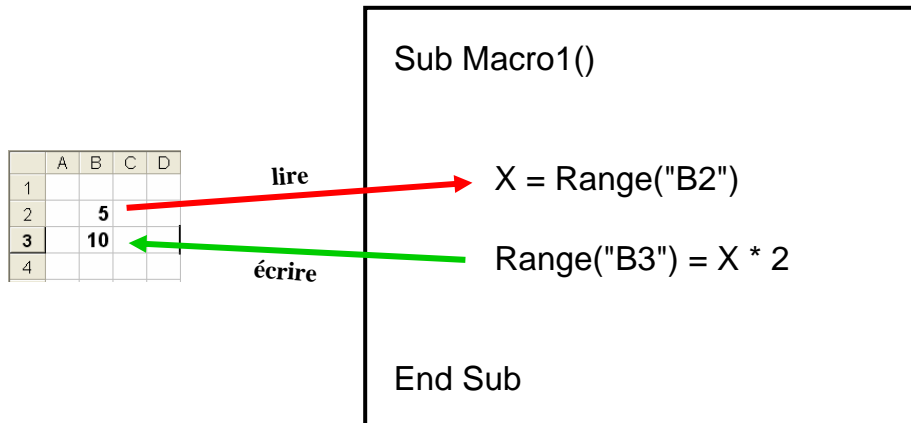


Heureusement, Excel offre de l'aide pour le débogage comme :

- la mise en évidence de la ligne douteuse
- le mode pas à pas (F8 pour avancer)
- la fenêtre des variables
- l'espion...
- etc...(cf aide)



7c1) Lire et écrire dans les cellules



7c2) Lire dans les cellules

Dans les applications qui vont suivre, il est important de pouvoir prendre une valeur de la feuille Excel pour la traiter dans une macro, puis de retourner l'information dans le tableur.

Il y a plusieurs façons de lire et d'écrire dans les cellules.

Pour les découvrir : un exemple simpliste...

- Lire la valeur de B1 et la stocker dans la variable X.
- Mettre le double de X en C2.

Sub lire()

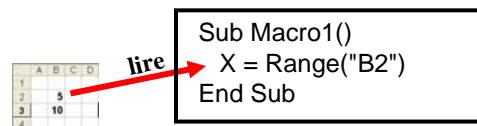
' Lire la valeur de la cellule B1
X = Range("B1").Value

' ou
X = Cells(1, 2).Value

' ou
Range("A1").Select
X = Selection.Offset(0, 1).Value

' ou
Range("B1").Select
X = ActiveCell.Value

End Sub



7c3) Ecrire dans les cellules

Dans les applications qui vont suivre, il est important de pouvoir prendre une valeur de la feuille Excel pour la traiter dans une macro, puis de retourner l'information dans le tableur.

Il y a plusieurs façons de lire et d'écrire dans les cellules.

Pour les découvrir : un exemple simpliste...

- Lire la valeur de B1 et la stocker dans la variable X.
- Mettre le double de X en C2.

Sub écrire()

' Ecrire le double de X dans la cellule C2

Range("C2").Formula = X * 2

' ou

Cells(2, 3).Formula = X * 2

(L,C)

' ou

Range("A1").Select
Selection.Offset(1, 2).Formula = X * 2

' ou

Range("C2").Select
ActiveCell.Formula = X * 2

' ou encore

LIGNE = 2
Range("C" & LIGNE).Select
ActiveCell.FormulaR1C1 = X * 2

End Sub

écrire

```
Sub Macro1()
  Range("B3") = X * 2
End Sub
```

7c4) Style de référence

Standard



L1C1



Écriture de formules dans une cellule de calcul

```
Sub écrire()
  Range("D5").Formula = "=B5*C5"
End Sub
```

```
Sub écrire()
  Range("D5").FormulaR1C1 = "=LC(-2)*LC(-1)"
End Sub
```

Row / Column

Manière simple et claire de traiter les références relatives

L1C1	cellule A1 (ligne 1, colonne 1)
LC(1)	une colonne plus à droite
LC(-1)	une colonne plus à gauche
L(1)C(1)	une ligne plus bas, une colonne à droite

7d) Couper - coller des macros

Exercice

Créer un chronomètre

- Ecrire une première macro pour le départ.
- Recopier et modifier cette macro pour l'arrivée.
- Idem pour mettre le chrono. à zéro.

	A	B	C	D	E
1					
2			Chronomètre		
3					
4		Départ	21:48:16		
5					
6		Arrivée	21:56:38		
7					
8		Temps	0:08:22		
9					
10					

Sub depart()

```
Range("C4").Formula = ""
Range("C6").Formula = ""
```

```
DEP = Now()
Range("C4").Formula = DEP
```

End Sub

	A	B	C	D
1				
2			Chronomètre	
3				
4				
5				
6				
7		Départ		
8				
9		Arrivée		
10				
11		Temps		
12				
13				

1) Préparer une grille selon le modèle ci-contre.

2) Ecrire dans l'éditeur VBA (créer une macro vide si nécessaire) les commandes suivantes :

Sub depart()

```
Range("C7") = ""
Range("C9") = ""
Range("C7") = Now()
Selection.NumberFormat = "h:mm:ss"
```

End Sub

Sub arrivee()

```
Range("C9") = Now()
Selection.NumberFormat = "h:mm:ss"
```

End Sub

Sub nouveau()

```
Range("C7") = ""
Range("C9") = ""
```

End Sub

3) Affectation des boutons

- Affecter au bouton "Départ" la macro depart()
- Affecter au bouton "Arrivée" la macro arrivee()
- Affecter au bouton vide la macro nouveau()

4) Format des cellules

- Sélectionner les cellules C7:C11
- Menu : Format/Cellule...
- Onglet : Nombre
- Catégorie : Heure
- Type : 13:30:55
- OK

5) Programmer la cellule C11 comme suit : = si(C9="";"";C9-C7)

6) Tester le chronomètre !

7e1) Test « IF THEN »

L'instruction " if " effectue une tâche si une certaine condition est remplie.

Le complément d'instruction " else" permet de faire une sélection entre 2 choix.

If Then

OU

If Then

.....
.....

Else

.....
.....

End If

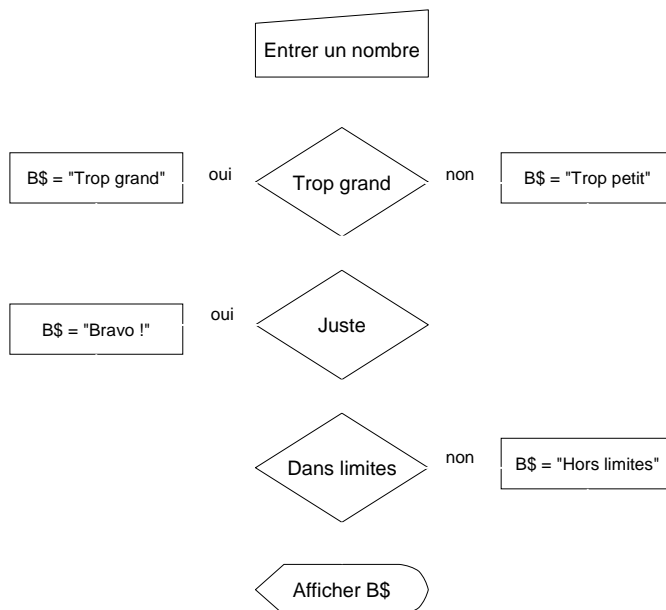
7e2) Test « IF THEN »

Organigramme

Exercice

Jeu du nombre caché

- Il faut trouver un nombre entre 1000 et 10000 caché dans la cellule E1.
- Une macro "message" indique à chaque exécution si le nombre proposé est plus petit ou plus grand que le nombre caché.



7e3) Test « IF THEN »

Exercice

Jeu du nombre caché

- Il faut trouver un nombre entre 1000 et 10000 caché dans la cellule E1.
- Une macro "message" indique à chaque exécution si le nombre proposé est plus petit ou plus grand que le nombre caché.

```
Sub Test_if()
```

```
N = Range("E1")  
R = Range("C5")
```

```
-----  
If R > N Then  
    B$ = "Trop grand"
```

```
Else  
    B$ = "Trop petit"
```

```
End If
```

```
-----  
If N = R Then B$ = "B r a v o !"
```

```
If R < 1000 Or R > 10000 Then B$ = "Hors limites!"
```

```
-----  
Range("D5").Formula = B$
```

```
End Sub
```

7e4) Test « Select Case »

Exercice

Jeu du nombre caché

Idem exercice no 13 mais
avec l'instruction
Select Case

Ajouter un compteur !

```
Sub Select_case()
```

```
N = Range("E1")  
R = Range("C5")  
C = Range("B11")  
C = C + 1
```

```
-----  
Select Case R
```

```
Case 0
```

```
    B$ = "Entrez un nombre !"
```

```
Case Is < 1000, Is > 10000
```

```
    B$ = "Hors limites!"
```

```
Case Is < N
```

```
    B$ = "Trop petit"
```

```
Case Is > N
```

```
    B$ = "Trop grand"
```

```
Case Else
```

```
    B$ = "B r a v o !"
```

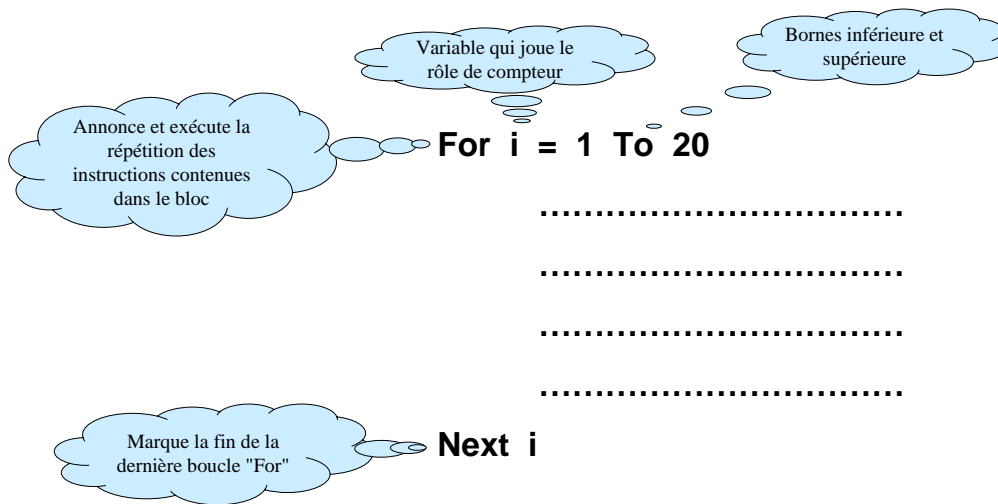
```
End Select
```

```
-----  
Range("D5").Formula = B$
```

```
Range("B11").Formula = C
```

```
End Sub
```

7g₁) La boucle « FOR NEXT » pour les itérations



7g₂) La boucle « FOR NEXT » pour les itérations

```
For i = 1 To 20  
.....  
.....  
For k = 100 To 150  
.....  
Next k  
.....  
.....  
Next i
```

7g₃) La boucle « FOR NEXT » pour les itérations

Il est possible de forcer la sortie d'une boucle.

For i = 1 to 20

.....

.....

If Then Exit For

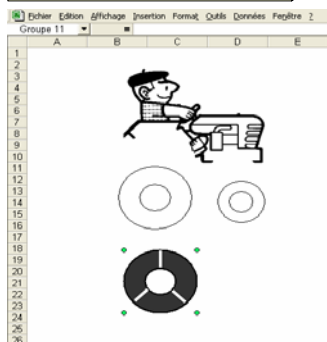
.....

Next i

7g₅) La boucle « FOR NEXT »

On utilise cette boucle chaque fois que l'on doit répéter une instruction ou une suite d'instruction un nombre déterminé de fois.

Faire tourner les roues
du tracteur.



Sub roue()

For i = 1 **To** 1000

ActiveSheet.Shapes("Group 14").Select
Selection.ShapeRange.IncrementRotation 6

ActiveSheet.Shapes("Group 15").Select
Selection.ShapeRange.IncrementRotation 7

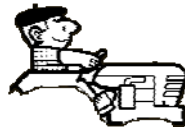
Range("A1").Select

Next i

End Sub

Exercice tracteur (boucle For...Next)

1) Insérer l'image "tracteur" ci-contre :



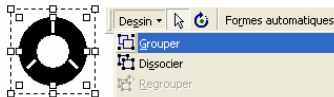
2) Formes automatiques / Formes de base /



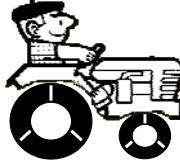
3) Sélectionner la roue et choisir la couleur de remplissage noire.



4) Dessiner 3 traits blancs d'une épaisseur de 2 1/4 pt dans la roue.



5) Sélectionner la roue et les traits puis "Dessin / grouper"



6) Faire une copie réduite de la roue et placer les deux roues sous le tracteur.

7) Créer une macro qui tourne les deux roues de quelques degrés.

Sélectionner une roue, cliquer sur "rotation libre" puis tourner l'objet de quelques degrés. Idem pour l'autre roue.



8) Dans la macro, encadrer le bloc de commandes par l'instruction "For...Next" comme suit :

```
Sub Roue()
  For I = 1 To 4000
    ActiveSheet.Shapes("Group 1").Select
    Selection.ShapeRange.IncrementRotation 2
    ActiveSheet.Shapes("Group 6").Select
    Selection.ShapeRange.IncrementRotation 3
    Range("A1").Select
  Next I
End Sub
```

7h) Fonction aléatoire

Programmation obligatoire car on ne peut pas figer un nombre aléatoire dans une cellule Excel avec la fonction ALEA() !

Les tirages au sort sont très utiles dans les applications pédagogiques et dans les jeux.

La fonction **Rnd** permet de générer un nombre aléatoire compris en 0.000000 et 0,999999. Pour obtenir un nombre entier il faudra lui associer la fonction **Int** (*Entier*).

Le 1 n'est pas inclus dans le nombre.

Un nombre entier
entre 1 et 45

$$\text{Lotto} = \text{Int}(\text{Rnd} * 45) + 1$$

Un nombre entier
entre 20 et 120

$$\text{Tirage} = \text{Int}(\text{Rnd} * 101) + 20$$

7h₂) Fonction aléatoire

Exercice

Tirer un nombre au sort entre deux bornes données et l'afficher.

```
Sub Tirage()
```

```
'Bornes
```

```
MINI = Range("E6")
```

```
MAXI = Range("E8")
```

```
'Coup de sac
```

```
Randomize
```

```
'Tirage
```

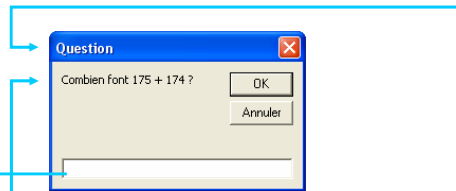
```
TIRSORT = Int((MAXI + 1 - MINI) * Rnd) + MINI
```

```
Range("C13").Formula = TIRSORT
```

```
Range("A1").Select
```

```
End Sub
```

7j₂) Boîte de saisie



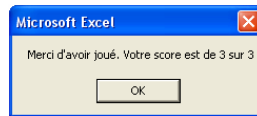
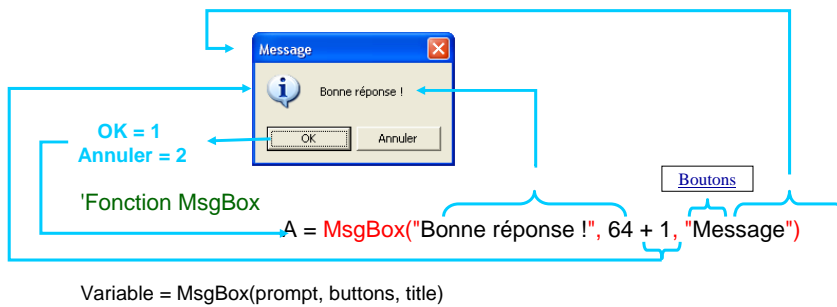
'Fonction de saisie

```
REP$ = InputBox("Combien font " & N1 & " + " & N2 & " ?", "Question")
```

Variable = InputBox(prompt, title, default, xpos, ypos)

prompt	Chaîne qui s'affiche comme message
title	Texte de la barre de titre
default	Texte initial dans la zone de saisie
x,ypos	Coordonnées de la boîte sur l'écran (par défaut, milieu horizontal, tiers vertical)

7j₃) Fenêtre de message



'Instruction MsgBox

MsgBox "Merci d'avoir joué. Votre score est de " & SCOR & " sur " & COMP - 1"

MsgBox prompt, buttons, title

Résumé

Les variables explicites:

Dim PARDEF As Variant
(double précision ou chaîne)

Dim NOM\$ As String
(chaîne de caractères)

Dim MONNAIE As String * 3
(maximum 3 caractères)

Dim NAISSANCE As Date
(de 01.01.0100 à 31.12.9999)

Dim VRAI_FAUX As Boolean
(True ou False)

Dim PIECE2 As Integer
(entiers de -32768 à 32767)

Dim MOYENNE As Currency
(+ 922'337'203'685'477,5808)

Dim PI As Decimal
(Currency à 14 décimales)

La portée des variables :

Portée macro
Sub macro1()
Static TVA = 7.6
End Sub

Portée module
Private A ou Dim A
Const TVA = 7.6
Sub macro1()
A =
End Sub

Portée projet
Public A
Public Const TVA = 7.6
Sub macro1()
A =
End Sub

Mettre la valeur de B1 dans la variable X :

X = Range("B1").Value
ou
X = Cells(1, 2).Value
ou
Range("A1").Select
X = Selection.Offset(0, 1).Value
ou
Range("B1").Select
X = ActiveCell.Value

Ecrire la valeur A dans la cellule C2 :

Range("C2").Formula = A
ou
Cells(2, 3).Formula = A
ou
Range("A1").Select
Selection.Offset(1, 2).Formula = A
ou
Range("C2").Select
ActiveCell.Formula = A
ou encore
LIGNE = 2
Range("C" & LIGNE).Select
ActiveCell.Formula = A

Résumé	
<p><u>Les tests :</u></p> <div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"> <pre>If condition Then</pre> </div> <div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"> <pre>If condition Then Else End If</pre> </div> <div style="border: 1px solid black; padding: 5px;"> <pre>Select Case Variable Case 0 Case 1 Case Is <N Case Is <P, >R Case Else End Select</pre> </div>	<p><u>Les boucles :</u></p> <div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"> <pre>For I = 1 to 20 Next I</pre> </div> <div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"> <pre>For I = 20 to 0 step -2 Next I</pre> </div> <div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"> <pre>For J = 20 to 100 If condition Then Exit For Next J</pre> </div> <div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"> <pre>For I = 1 to 20 For K = X to Y Next K Next I</pre> </div> <div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"> <pre>While A <100 A = A + 1 Wend</pre> </div> <div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"> <pre>Do While condition (tant que) Loop</pre> </div> <div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"> <pre>Do Loop While condition (tant que)</pre> </div> <div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"> <pre>Do Until condition (jusqu'à) Loop</pre> </div> <div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"> <pre>Do Loop Until condition (jusqu'à)</pre> </div> <p><u>Tirage au sort d'un nombre T compris entre Min et Max:</u></p> <div style="border: 1px solid black; padding: 5px;"> <pre>Randomize T = Int(Rnd * (Max + 1 - Min)) + Min</pre> </div>

Vade-mecum	
<p><u>Sélectionner le groupe de cellules B4 à C6:</u> avec Range : Range("B4:C6").Select avec Cells : Range(Cells(4, 2), Cells(6, 3)).Select avec une variable : Var = 6 Range("B4:C" & Var).Select</p>	<p><u>Sélectionner la zone de B1 jusqu'à la cellule active :</u> Range("B1:" & ActiveCell.Address).Select</p> <p><u>Activer la cellule utilisée la plus éloignée de A1 :</u> ActiveCell.SpecialCells(xlLastCell).Select</p>
<p><u>Valeur de la cellule A2 d'une autre fenêtre :</u> avec Range : A = Range("Nomfenêtre!A2").Value avec Cells : A = Worksheets("Nomfenêtre").Cells(2,1).Value</p>	<p><u>Date du jour :</u></p> <p>Date figée; ne change pas à chaque ouverture : Range("A1").Value = Now()</p> <p>Date mobile; change à chaque ouverture : Range("A2").Formula = "=Now()"</p>
<p><u>Sortir...</u></p> <p>d'une boucle : If Then Exit For d'une macro : If Then Exit Sub d'un test : If Then rien Else ElseIf End If</p>	<p><u>Exécuter une macro à l'ouverture du classeur :</u></p> <p>Il suffit de la nommer : Auto_Open() ou, autre méthode :</p> <p>Placer la macro Private Sub Workbook_Open() dans le ThisWorkbook</p>
<p><u>Arrondir N au 5 ct :</u> A = Round(N * 2 , 1) / 2</p>	<p><u>Exécuter une macro depuis une macro :</u> Application.Run "NomMacro"</p>
<p><u>Le nombre A est-il pair ?</u> If A / 2 = A \ 2 Then Text = "Pair" Else Text = "Impair" ou, autre méthode : If A / 2 = Int(A / 2) Then Text = "Pair" Else Text = "Impair"</p>	