

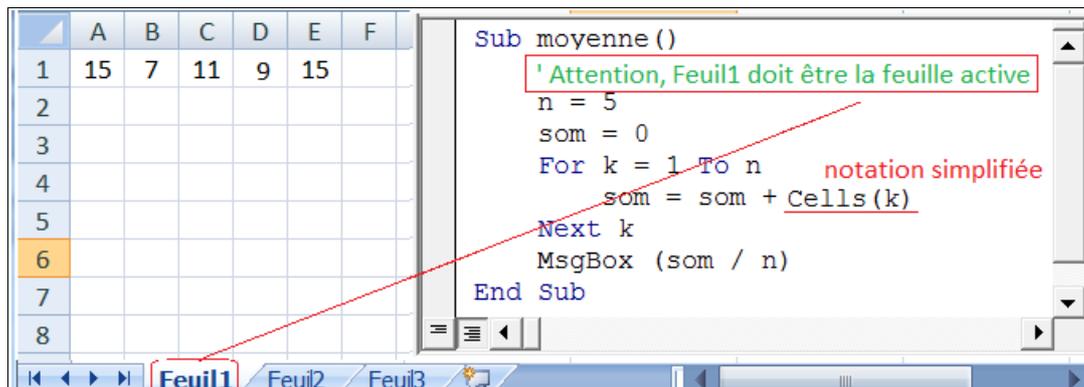
Simplification des notations sous OpenOffice

<http://revue.sesamath.net/spip.php?article775>

patrick.raffinat@univ-pau.fr

A) Exemple 1 : moyenne

Pour mémoire, je rappellerai que le programme VB (Excel) de l'article principal était le suivant :



Sous OpenOffice, le code équivalent est le suivant :

```
Sub moyenne()
    Set F = ThisComponent.CurrentController.ActiveSheet
    n = 5
    som = 0
    For k=1 To n
        som = som + F.getCellByPosition(k-1,0).value
    Next k
    MsgBox som/n
End Sub
```

Si on ne le simplifiait pas, mieux vaudrait donc utiliser un autre langage de programmation au lycée ! C'est pourquoi j'ai introduit dans le module « Module2 » du fichier « exemple1.ods » une fonction nommée Cells :

```
' renvoie la valeur de la cellule ligne l, colonne col
' Rq : ne peut donc être utilisé pour mettre une valeur dans la cellule
Function Cells(col As Integer)
    Dim F As Object
    Set F = ThisComponent.CurrentController.ActiveSheet
    Cells = F.getCellByPosition(col-1,0).value
End Function
```

Le module « Module1 » contient le programme de calcul de la moyenne, identique à celui proposé pour Excel. Il ne reste plus qu'à en supprimer le contenu avant de fournir le fichier à vos élèves pour qu'ils traitent cet exemple (ou un exemple analogue)...

B) Exemple 2 : suites récurrentes

Pour mémoire, je rappellerai que le programme VB (Excel) de l'article principal était le suivant :

```
Sub suite()  
    Cells(1) = 100  
    For k=2 To 5  
        Cells(k) = 0.5*Cells(k-1) + 10  
    Next k  
End Sub
```

Sous OpenOffice, la fonction Cells de l'exemple 1 doit être adaptée, car il faut non seulement extraire le contenu de cellules, mais aussi écrire dans la feuille de calcul active. La nouvelle fonction Cells est disponible dans le module « Module2 » du fichier « exemple2.ods » :

```
' renvoie la référence de la cellule ligne 1, colonne col  
' Rq : ne pas confondre la cellule avec sa valeur  
Function Cells(col As Integer)  
    Dim F As Object  
    Set F = ThisComponent.CurrentController.ActiveSheet  
    Set Cells = F.getCellByPosition(col-1,0)  
End Function
```

La fonction ne renvoie plus la valeur de la cellule, mais se contente de la référencer. Il faut donc ajouter « .value » pour accéder à la valeur, comme le montre le code modifié du programme « suite » (disponible dans le module « Module1 »).

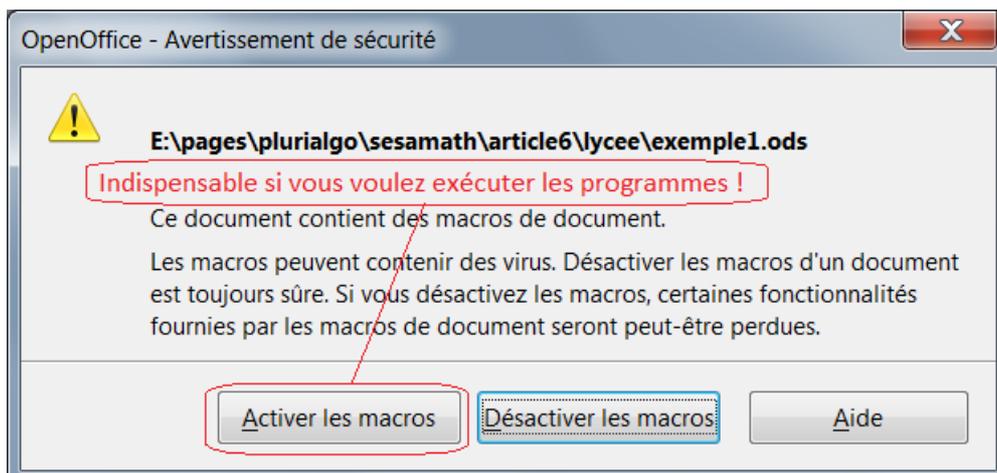
```
Sub suite()  
    Cells(1).value = 100  
    For k=2 To 5  
        Cells(k).value = 0.5*Cells(k-1).value + 10  
    Next k  
End Sub
```

Il est à noter que ce code reste valable sous Excel et qu'il est même plus rigoureux que celui donné en introduction, où Excel « devine » d'après le contexte que le programmeur pense en fait à la valeur de la cellule et non à la cellule elle-même. J'ai longtemps imposé le « .value » à mes étudiants d'IUT, avant d'y renoncer par pragmatisme : certains d'entre eux se sont en effet aperçus un jour qu'Excel ne l'imposait pas, et comme en plus ils ne comprenaient pas bien la distinction entre cellule et valeur d'une cellule...

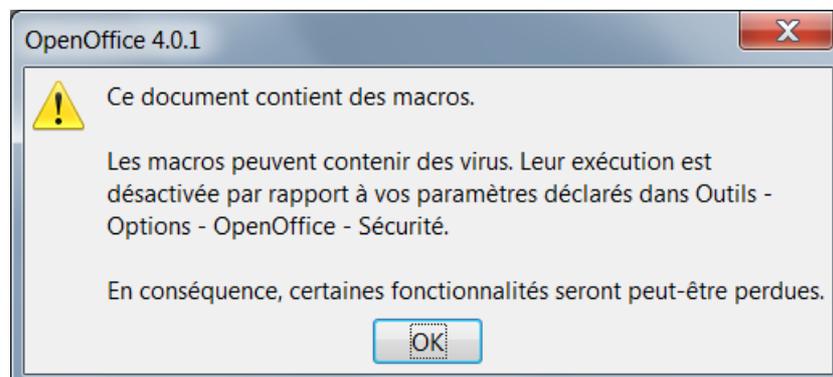
C) Exécution des programmes

C.1) Sécurité

Normalement, vous devriez voir cette fenêtre à l'ouverture des fichiers fournis :

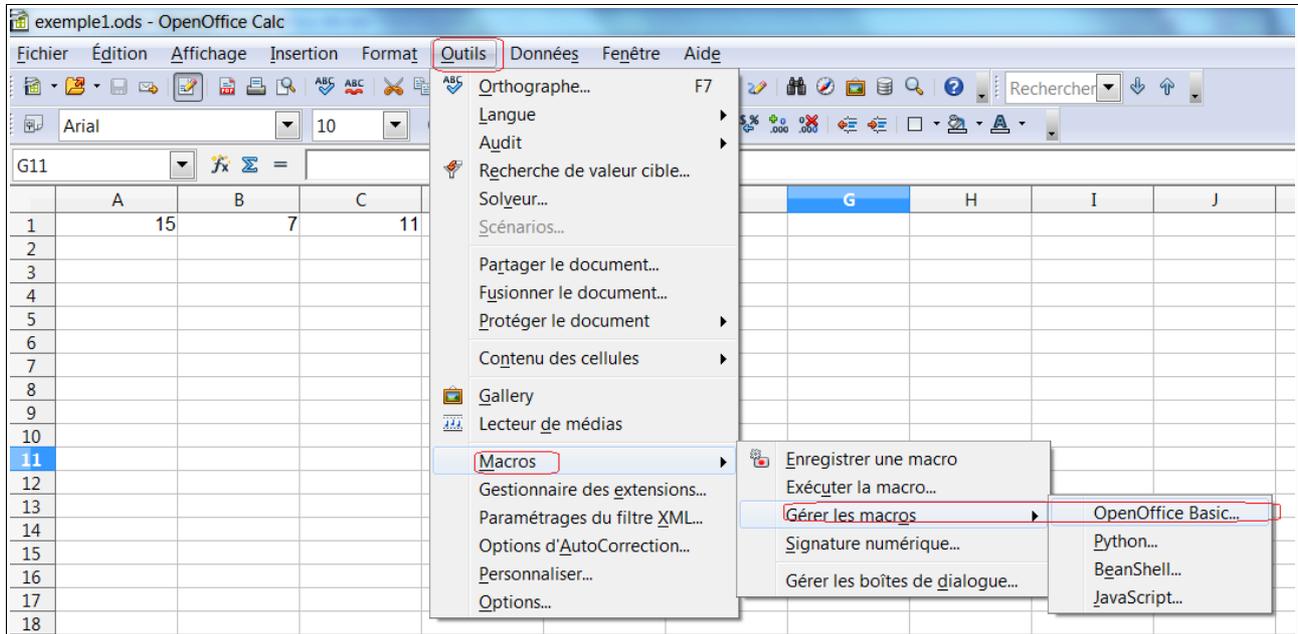


Il ne faut alors surtout pas oublier d'activer les macros si vous voulez exécuter ou modifier les programmes. Mais en ouvrant sur mon ordinateur professionnel les programmes que j'avais créés sur mon ordinateur personnel, j'ai eu la surprise de voir ceci :

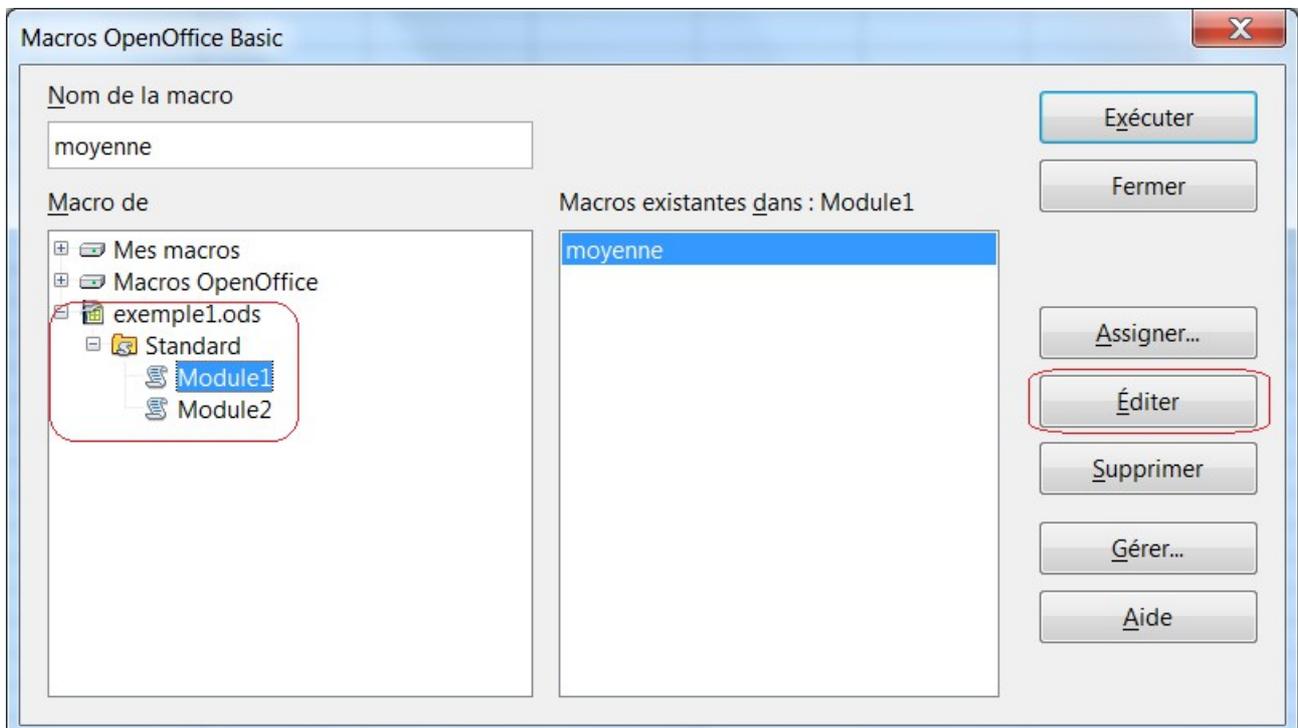


Donc, pour ceux qui n'ont pas l'habitude de programmer sous OpenOffice, voici ce qu'il faut faire dans une telle situation : changer le niveau de sécurité (mis ici à « très élevé par un technicien) en « niveau de sécurité moyen », puis fermer et rouvrir le fichier.

C.2) *Editeur Visual Basic*



Puis :



Puis ... débrouillez-vous !