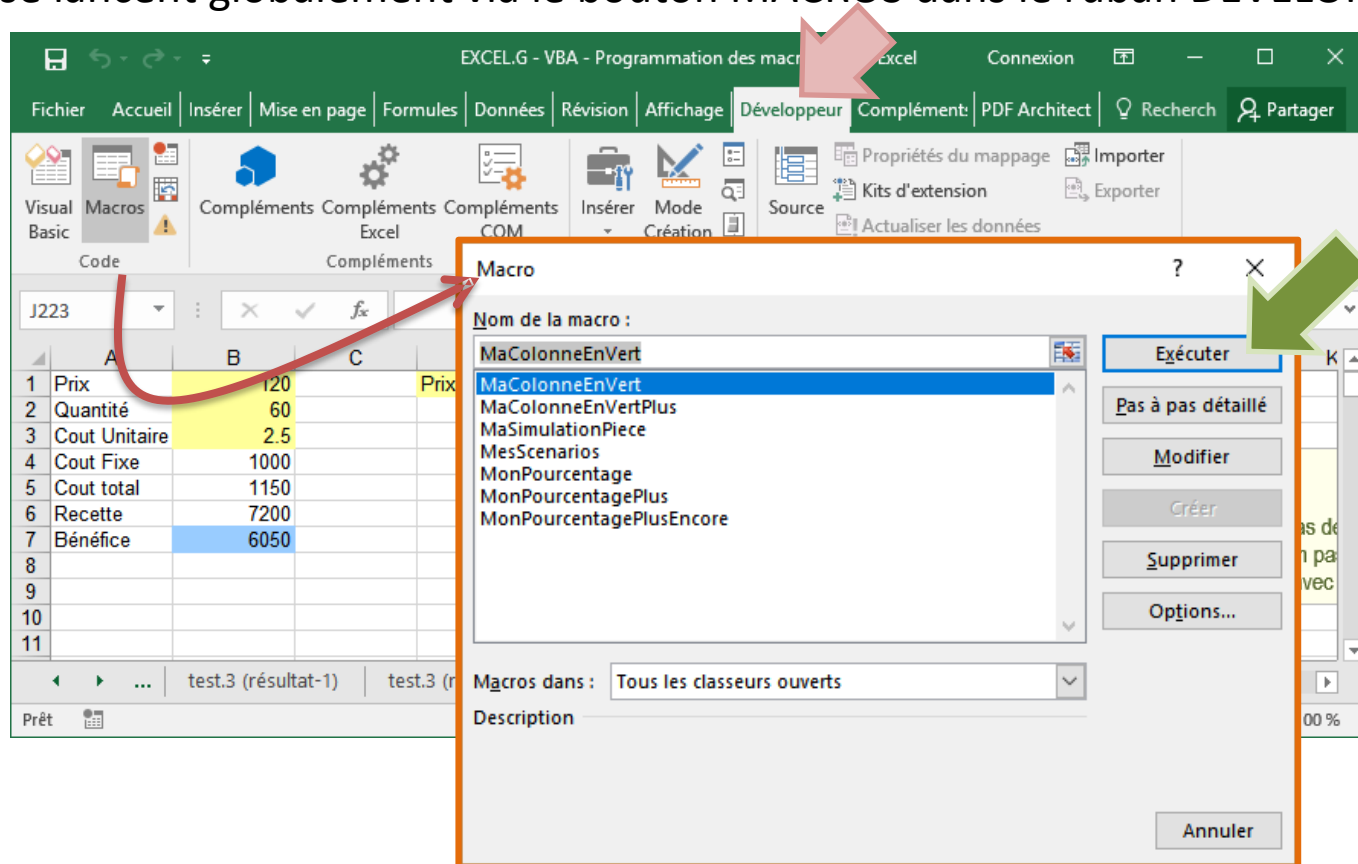


Programmation des macros – Travailler directement sur les feuilles

LES MACROS (1)

Les macros sont également des procédures que l'on crée à l'intérieur d'un module. Mais, à la différence des **Function**, ce sont des **Sub()** sans paramètres qui peuvent manipuler (accéder et modifier) directement les objets Excel (classeurs, feuilles, cellules, graphiques, scénarios, tableaux croisés dynamiques...).

Ils ne s'exécutent pas de la même manière. Au lieu de les insérer dans une cellule, ils se lancent globalement via le bouton MACROS dans le ruban DEVELOPPEUR.



Une manière simple de générer une macro est de lancer l'[enregistreur de macros](#).
Du code VBA est automatiquement généré.

Exemple : mettre en gras et vert
le contenu des cellules A1 et A2



	A
1	10
2	12

Un nouveau module
« Module1 » est
automatiquement créé.

Microsoft Visual Basic pour Applications - Exemples illustratifs.xlsm

Fichier Edition Affichage Insertion Format Débogage Exécution

Projet - VBAProject

Solver (SOLVER.XLAM)

VBAProject (Exemples illustratifs.xlsm)

- Microsoft Excel Objets
 - Feuil1 (FeuilleTest)
 - Feuil2 (Feuille Test 2)
 - Feuil3 (Feuil2)
 - Feuil4 (Feuil3)
 - ThisWorkbook
- Modules
 - Module1
 - ModuleVBA

Propriétés - Module1

Module1 Module

Alphabétique Par catégorie

(Name) Module1

Exemples illustratifs.xlsm - Module1 (Code)

(Général) Macro1

```
Sub Macro1()  
    ' Macro1 Macro  
    '   
    Range("A1:A2").Select  
    With Selection.Font  
        .Color = -11489280  
        .TintAndShade = 0  
    End With  
    Selection.Font.Bold = True  
End Sub
```

Une macro correspond à un **Sub()** sans paramètres.

Ecrire du code VBA pour les macros se révélera plus simple au final.

Avantages :

- Il n'y a pas plus simple pour produire du code, on peut créer et exécuter une macro sans aucune notion de programmation
- Il nous donne des indications précieuses sur les commandes associées aux objets Excel

Inconvénients :

- On travaille à structure fixée, si la configuration de la feuille change, il n'est pas possible de lancer la macro
- On ne bénéficie pas de la puissance des structures algorithmiques

En définitive :

- Il peut nous aider à rédiger notre code en nous donnant des pistes sur la syntaxe des commandes et les objets adéquats à manipuler (ex. imprimer automatiquement des feuilles, on lance l'enregistreur une fois, on intègre son code dans le notre à l'intérieur d'une boucle).

Ecrire directement des macros est simple une fois assimilé la philosophie de l'approche, et identifié les principaux objets et l'accès à leurs propriétés et méthodes (l'enregistreur peut nous y aider).

Classeurs

`Workbooks("classeur1.xlsm").Activate`

Activer (sélectionner) le classeur dont le nom de fichier est "classeur1.xlsm"

Feuilles

`Sheets("Feuil1").Activate`

Dans le classeur courant, activer la feuille de calcul dont le nom est "Feuil1" (visible dans la languette au bas de la feuille)

`Workbooks("classeur1.xlsm").Sheets("Feuil1").Activate`

On peut combiner les écritures.

Cellules

`Cells(1,1).Value = 15`

Dans la feuille courante du classeur courant, insérer la valeur 15 dans la cellule **ligne n°1, colonne n°1** c.-à-d. en A1, les coordonnées sont absolues ici.

!

`Sheets("Feuil1").Cells(1,1).Value = 15`

De nouveau, on peut combiner.

Exemple de macros – Simulation valeurs de TVA

Ecrire une macro qui insère différentes valeurs de TVA en **B2** et récupère les valeurs de prix TTC en **B3**.

Les différentes valeurs de TVA testées doivent être retranscrites au fur et à mesure dans la colonne D.

	A	B	C	D	E
1	PHT	100		TVA (%)	Prix TTC
2	TVA (%)	30			
3	PTTC	=B1*(1+B2/100)			
4					

Les valeurs de Prix TTC correspondantes doivent être recensées en colonne E

```
Sub SimulationTVA()  
'variables  
Dim pht As Double, pttc As Double  
Dim tva As Double  
Dim i As Long  
'début d'écriture des valeurs en ligne 2  
i = 2  
'récupérer la valeur du PHT  
pht = Cells(1, 2).Value 'en B1  
'faire varier la tva de 10% à 30% avec un pas de 5%  
For tva = 10 To 30 Step 5  
    'insérer la valeur de la TVA en B2  
    Cells(2, 2).Value = tva  
    'Récupérer le prix ttc en B3  
    pttc = Cells(3, 2).Value  
    'inscription des valeurs  
    'TVA en colonne D  
    Cells(i, 4).Value = tva  
    'PTTC en colonne E  
    Cells(i, 5).Value = pttc  
    'passage à la ligne suivante  
    i = i + 1  
Next tva  
End Sub
```

Remarque : il faut être sur la feuille adéquate avant de lancer la macro, sinon le programme ne saura pas où chercher **Cells(...)**.

A l'issue de la simulation...

	A	B	C	D	E
1	PHT	100		TVA (%)	Prix TTC
2	TVA (%)	30		10	110
3	PTTC	130		15	115
4				20	120
5				25	125
6				30	130

