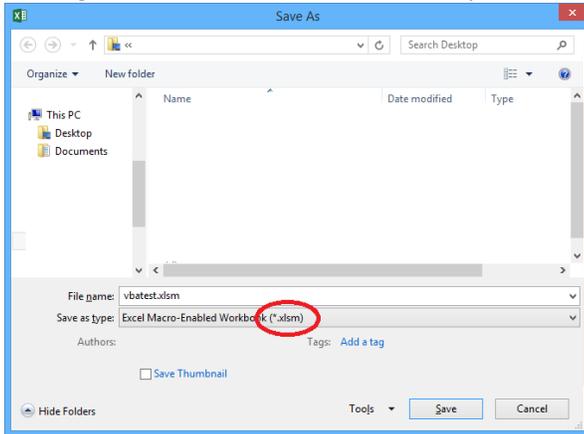


Ce document présente une courte introduction à la programmation de macros VBA (Visual Basic Application) sous Excel. Ceci vous permettra d'automatiser diverses tâches et de vous donner accès à une interface de programmation simple à utiliser pour différents projets.

1. Accéder à l'interface VBA

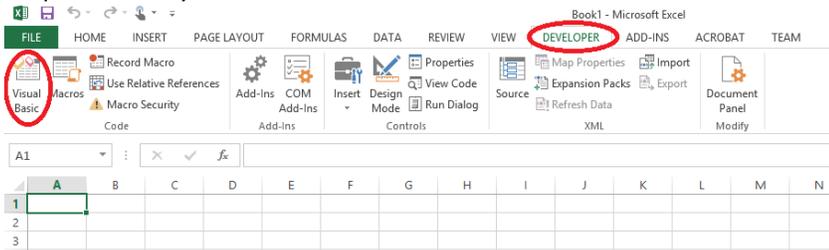
Cette section présente comment accéder à l'interface VBA avec la version Excel 2013 anglaise. Il se peut que certaines options ne soient pas placées au même endroit que sur votre version. Vous pouvez retrouver les informations requises en cherchant sur Google.

Sauvegarder votre fichier Excel avec l'option « Macro Enabled » (fichier .xlsm)

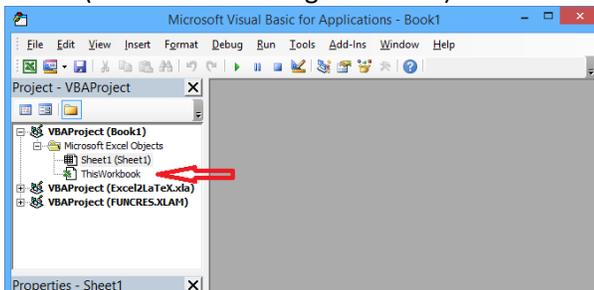


Ouvrir Excel et aller au menu « Developer » ou « Développeur ». Si celui n'apparaît pas, suivre les indications suivantes : *To display the **DEVELOPER** tab, click on the File menu and then select Options from the drop down menu. When the **Excel Options** window appears, click on the Customize Ribbon option on the left. Click on the **Developer** checkbox under the list of Main **Tabs** on the right. Then click on the OK button.*

Cliquer sur le symbole Visual Basic



Vous obtiendrez une fenêtre comme ceci, double-cliquer sur ThisWorkbook et une fenêtre blanche apparaîtra à droite (au lieu de la zone grise ici-bas).



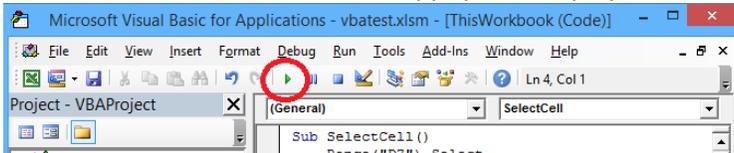
2. Exemple : Sélection d'une cellule

Dans la page blanche, entrer le code suivant :

```
Sub SelectCell()  
    Range("D7").Select  
End Sub
```

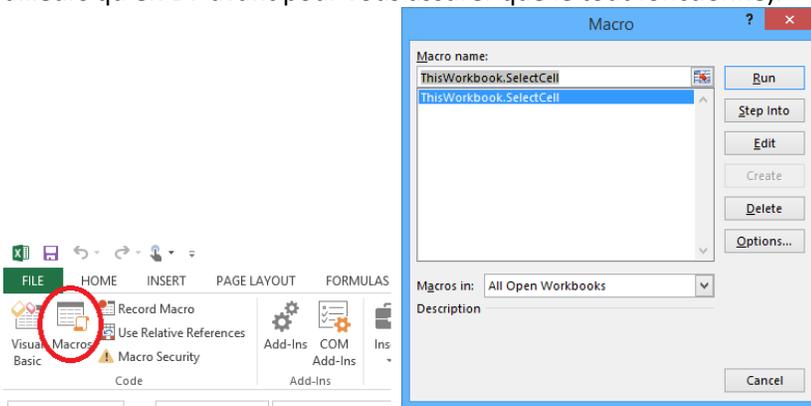
Vous avez maintenant 3 options pour faire « jouer » cette macro qui sélectionnera la cellule D7. Essayer les trois méthodes.

a) Directement de interface VBA, appuyer sur « play »

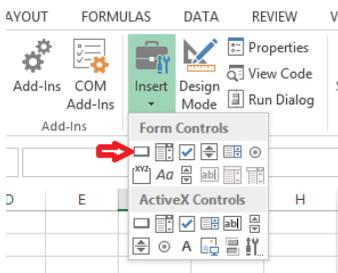


Si vous pouvez voir votre fichier Excel en même temps (2 fenêtres côte à côte) vous verrez la sélection de la cellule D7 se faire. Sinon, en revenant au fichier, vous verrez que la cellule a été sélectionnée.

b) Dans Excel, cliquer sur Macros. Dans l'interface qui apparaîtra, choisir votre macro et faire « run ». (Cliquer ailleurs qu'en D7 avant pour vous assurer que le tout fonctionne).



c) Afin de faciliter la tâche au lieu d'avoir à faire ces étapes, vous pouvez placer un bouton de contrôle. Toujours dans l'onglet Developer, aller dans Insert, puis choisir le bouton de commande :



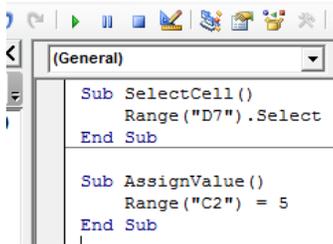
Vous devez ensuite dessiner le bouton (cliquer, tenir, déplacer la souris, dé-cliquer). Une fenêtre apparaîtra automatiquement. Dans la liste de choix de macros, sélectionner la vôtre. Lorsque vous cliquez sur le bouton, votre macro s'exécutera. Pour déplacer ou redimensionner votre bouton, faire un clic droit sur celui-ci.

3. Exemple : Attribution d'une valeur

Nous allons maintenant assigner une valeur à une cellule donnée avec différentes méthodes.

a) Entrer le code suivant :

En exécutant ce code, la cellule C2 prendra la valeur « 5 ». Vous n'êtes pas obligé de sélectionner la cellule avant de lui assigner une valeur.



b) au lieu de `Range("C2") = 5`, entrer

```
Sub AssignValue()  
    'Range("C2") = 5  
    Cells(2, 3) = 5  
End Sub
```

Cette méthode est de loin supérieure car elle permet de sélectionner la cellule par des numéros que l'on peut changer facilement plus tard si l'on apporte des modifications au fichier Excel ou si nous voulons sélectionner une cellule à partir de variables ex : `Cells(myrow, mycol)`.

Remarquer le texte en vert, ce code ne sera pas exécuté, c'est un commentaire (placer ' en avant du texte pour la placer en commentaires).

c) Nous allons maintenant affecter une valeur à partir d'une variable. Noter qu'il faut d'abord créer la variable et lui assigner un type. Dans ce cas nous créons une valeur entière (-1,0,1,2) Integer. Pour une variable à point flottant il aurait fallu choisir `Dim tempval As Double`.

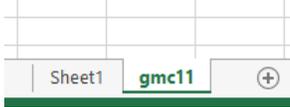
```
Sub AssignValue()  
    'Methode B  
    Cells(2, 3) = 5  
  
    'Methode C  
    Dim tempval As Integer  
    tempval = 10  
    Cells(2, 4) = tempval  
  
End Sub
```

d) Nous allons maintenant attribuer une valeur à partir d'une autre cellule.

Dans le case E2 et E3 entrer respectivement 100 et 200. Entrer le code suivant pour obtenir 300 dans le case E4.

```
Cells(4, 5) = Cells(2, 5) + Cells(3, 5)
```

e) Créer une nouvelle page appelée « gmc11 ». Sélectionner ensuite une case sur votre page originale.



Entrer le code suivant :

```
Sheets("gmc11").Cells(2, 1) = Cells(4, 5)
```

La cellule A2 de la feuille gmc11 prendra la valeur 300. Remarquer que dans la partie de droite, nous n'avons pas eu à spécifier la page car il s'agit de la page actuellement sélectionnée.

Pour sélectionner une nouvelle page, faire `Sheets("gmc11").Select`

f) Pour sélectionner plusieurs cellules, essayer une des méthodes suivantes :

```
Range("D8:H22").Select
```

ou

```
Range(Cells(4, 4), Cells(10, 10)).Select
```

e) Pour effacer le contenu d'une cellule. La première fonction efface la valeur alors que la seconde efface aussi le formatage de la cellule.

```
Cells(4, 5).ClearContents
```

```
Cells(4, 5).Clear
```

4. Exemple : Boucles IF

Voici un exemple d'une boucle IF. Entrer une valeur (par exemple « 5 ») dans la cellule C2.

```
Sub iftest()
```

```
    If Cells(2, 3) < 0 Then  
        Cells(2, 4) = "Negatif"  
    Else  
        Cells(2, 4) = "Positif"  
    End If
```

```
End Sub
```

5. Exemple : Boucle FOR

Entrer les valeurs 1 à 10 dans la colonne A à partir de A1 puis le code suivant :

```
Sub fortest()
```

```
    Dim compt As Integer  
    Dim addedval As Integer  
    addedval = 0  
  
    For compt = 1 To 10  
        addedval = Cells(compt, 1) + addedval  
    Next compt
```

```
    Cells(1, 2) = addedval
```

```
End Sub
```

Ceci écrira la somme des valeurs dans la colonne A. Par contre, si on change une valeur dans la colonne A, la somme ne sera pas mise à jour.

Il aurait été possible dans ce cas d'écrire directement une formule tel que : `Cells(1, 2) = "=sum(A1:A10) "`

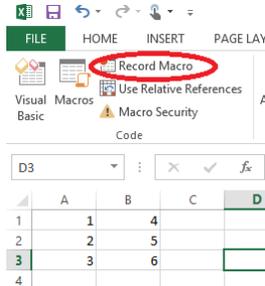
6. Apprendre d'autres techniques

Il est possible de faire énormément de procédures avancées avec les macros VBA. L'objectif de ce document n'est que d'introduire la base des macros VBA pour que vous ayez une idée de ce qui existe et ainsi continuer d'apprendre par vous-même par la suite. Deux stratégies de base s'offrent à vous pour découvrir d'autres fonctions.

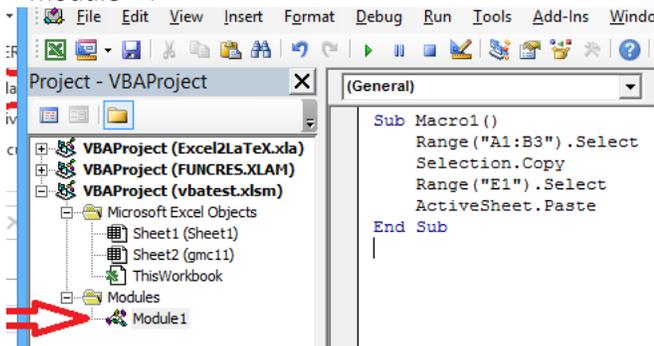
a) L'enregistrement de macro

Il est possible d'enregistrer vos manipulations sous Excel pour que ce code soit répété par la suite. Ceci peut être utilisé pour comprendre comment faire certaines procédures. Par exemple vous avez besoin de copier plusieurs valeurs d'un fichier et de les copier ailleurs mais vous ne savez pas comment faire.

Entrer les valeurs suivantes puis cliquer sur Record Macro



Donner un nom à la macro. Le logiciel commence à enregistrer vos commandes. Sélectionner les cellules A1 à B3 et faire « copier ». Cliquer sur la cellule E1 puis faites « coller ». Arrêter l'enregistrement (à la même place où vous avez cliqué sur Record Macro sera inscrit « Stop recording »). La macro se trouvera probablement sous « Module » :



Vous voyez maintenant comment sélectionner une plage de cellule, de la copier, puis de la copier ailleurs.

I don't always do Excel macros, but when I do I don't remember how to do anything.

Il y a beaucoup de fonctions à retenir et une nomenclature particulière pour appeler les différentes fonctions. Il est très difficile de se rappeler comment faire ce que l'on veut. Il existe bien entendu beaucoup de tutoriels sur internet qui peuvent vous aider. Par contre, une des manières les plus efficaces de trouver l'information est simplement de chercher ce que vous voulez faire sur Google et vous trouverez fort probablement directement un tutoriel sur ce que vous voulez faire ou une question déjà répondue à ce sujet sur un forum.

EXERCICES

Si votre programme ne fonctionne pas, vous pouvez cliquer dans la marge de l'interface VBA pour placer un "break point". Vous pourrez alors visualiser les valeurs de votre algorithme en passant la souris par-dessus ces valeurs et ainsi comprendre ce qui ne fonctionne pas en faisant fonctionner votre programme étape par étape.

Ouvrir le fichier « Excel GMC3006_VBAExerc1.xlsm »

Dans l'onglet « data » vous avez

	A	B	C	D	E	F	G
1							
2			MSFT	GOOG	AAPL	KO	PEP
3		Prix	58.03	769.64	106.94	43.32	107.45
4		Ventes	84,695,000,000	73,590,000,000	231,283,000,000	43,651,000,000	63,053,000,000
5		Profits bruts	51,915,000,000	45,426,000,000	89,026,000,000	26,316,000,000	34,527,000,000
6		Profits nets	16,798,000,000	15,826,000,000	53,394,000,000	7,351,000,000	5,452,000,000
7		Profits par actions	2.1	22.84	9.22	1.67	3.67
8							

Aller à l'onglet « result » :

	A	B	C
1			
2			
3		Prix	
4		Ventes	
5		Profits bruts	
6		Profits nets	
7		Profits par actions	
8		PE	
9		Marge brute	
10		Marge nette	

Effacer d'abord le contenu des cellules C3 à i10 de l'onglet result (avec une macro).

Ensuite, remplir les données de l'onglet result à partir de l'onglet data.

Il faut copier uniquement les colonnes dont les variables PE sont plus petites ou égales à 28 et dont la marge nette est supérieure ou égale à 18.

La variable PE se calcule ainsi : $(\text{Prix}) / (\text{profits par actions})$ et représente le nombre de dollar payé lors de l'achat d'une action pour chaque dollar de profit par action.

La marge brute se calcule ainsi : $(\text{Profits bruts}) / (\text{Ventes})$

La marge nette se calcule ainsi : $(\text{Profits nets}) / (\text{Ventes})$

Le résultat que vous devriez obtenir ainsi qu'un exemple de solution est disponible sur le portail. Vous avez grand avantage à ne regarder le solutionnaire qu'en dernier recours. Bien comprendre le solutionnaire présenté même si votre propre solution fonctionne bien.

RÉSULTAT :

	A	B	C	D
1				
2			MSFT	AAPL
3		Prix	58.03	106.94
4		Ventes	84695000000	2.31E+11
5		Profits bruts	51915000000	8.9E+10
6		Profits nets	16798000000	5.34E+10
7		Profits par actions	2.1	9.22
8		PE	27.63333333	11.5987
9		Marge brute	61.29641655	38.49224
10		Marge nette	19.83352028	23.086
11				

CODE :

```

Sub exercicel()
  Sheets("result").Select
  Range("C2:I10").ClearContents

  Sheets("data").Select

  Dim i As Integer
  Dim j As Integer

  Dim selectval As Integer
  selectval = 0

  Dim PE As Double
  Dim margebrute As Double
  Dim margenette As Double

  Dim numberok As Integer
  numberok = 0

  For i = 3 To 7
    PE = Cells(3, i) / Cells(7, i)
    margebrute = 100 * Cells(5, i) / Cells(4, i)
    margenette = 100 * Cells(6, i) / Cells(4, i)

    If PE <= 28 And margenette >= 18 Then
      For j = 2 To 7
        Sheets("result").Cells(j, 3 + numberok) = Cells(j, i)
      Next j
      Sheets("result").Cells(8, 3 + numberok) = PE
      Sheets("result").Cells(9, 3 + numberok) = margebrute
      Sheets("result").Cells(10, 3 + numberok) = margenette
      numberok = numberok + 1
    End If
  Next i

  Sheets("result").Select
End Sub

```