

Travaux Pratiques du Chapitre 1

Manipulation des Macros sous Excel

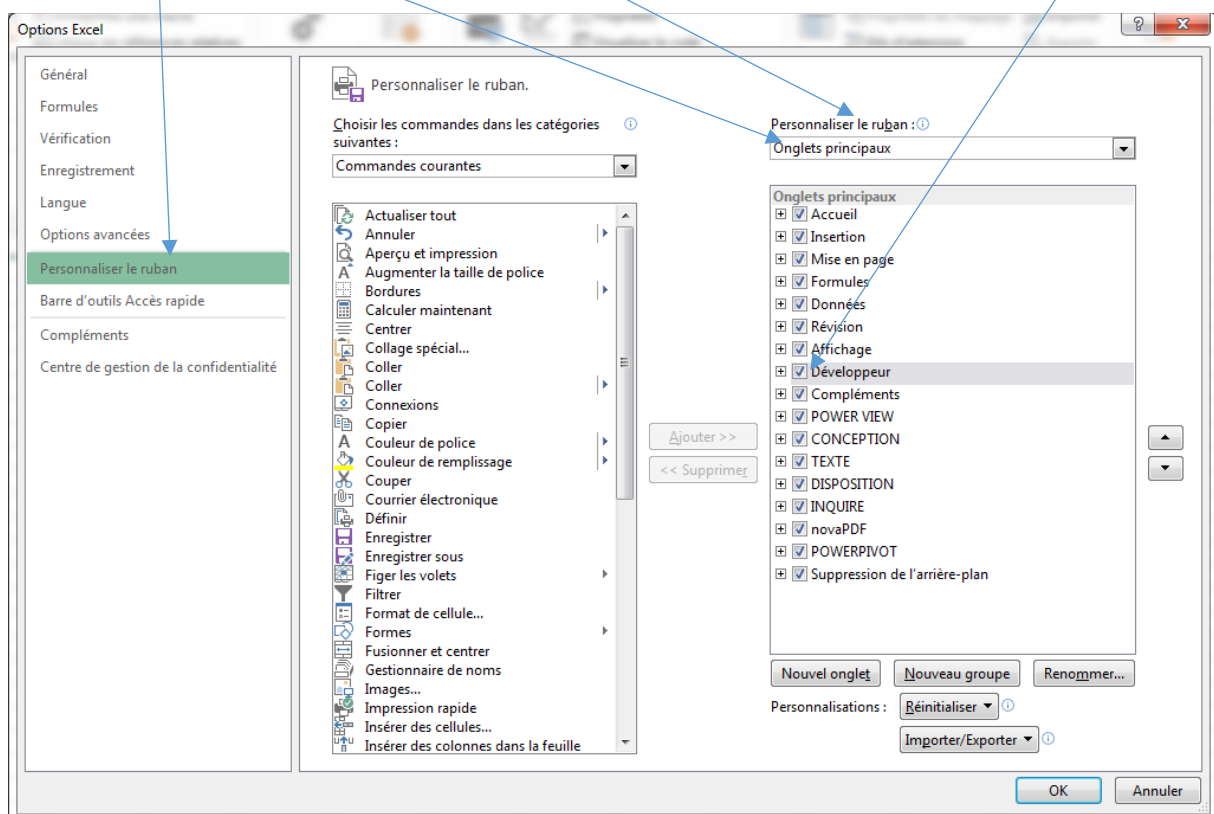
I-Ajout de l'onglet "DEVELOPPEUR" au ruban d'Excel

Sous Excel 2010-2013, pour ajouter l'onglet "DEVELOPPEUR" au ruban d'Excel, faire:

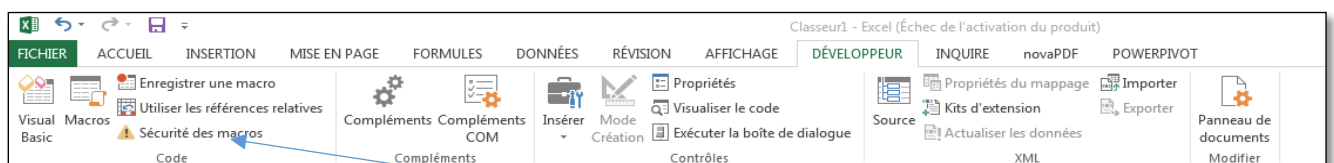
1-Menu **Fichier → Options**

2- Dans la boîte de dialogue qui s'affiche sélectionner, dans le volet gauche, la commande **Personnaliser le ruban**

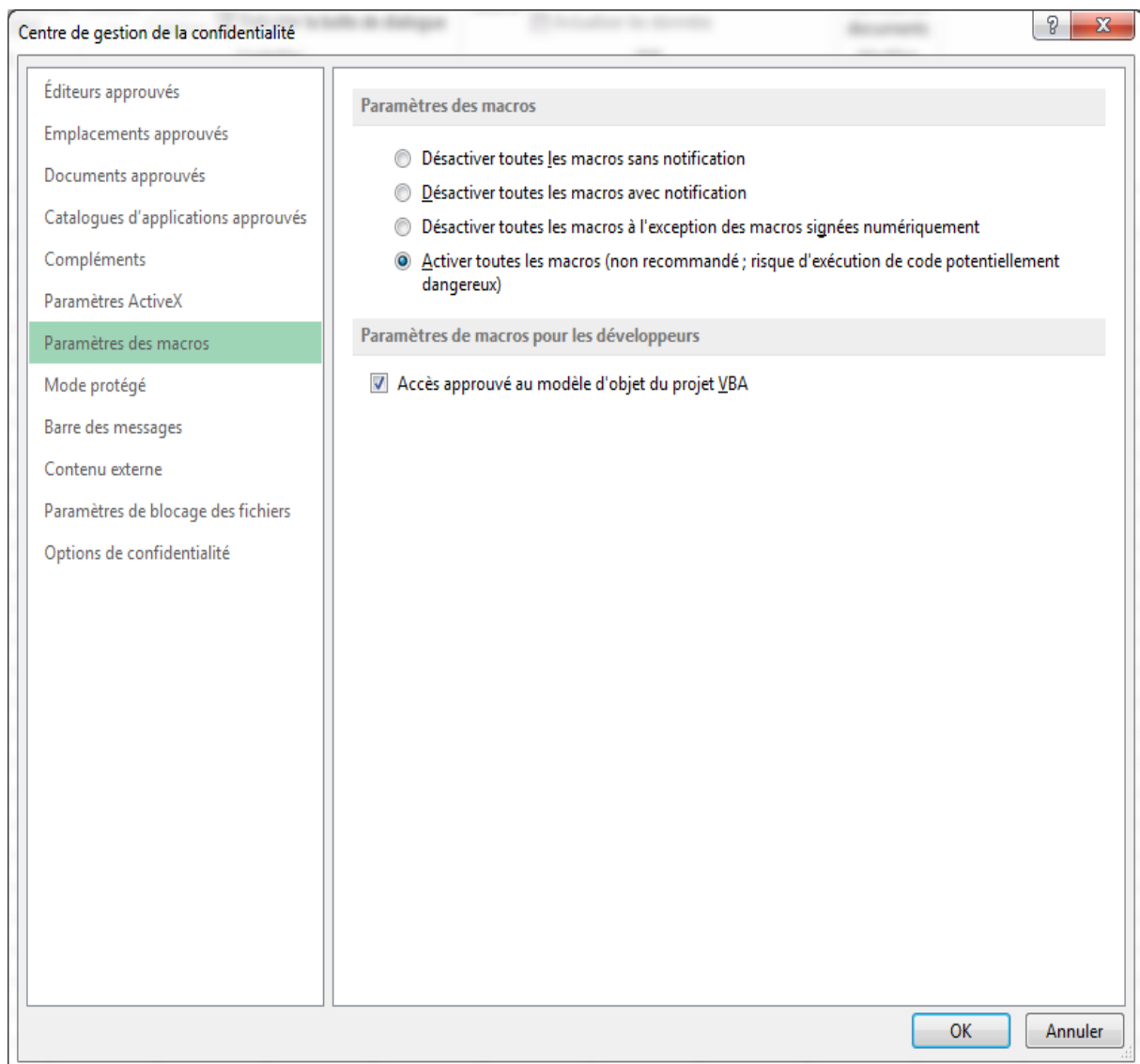
3- Dans la zone déroulante **Personnaliser le ruban** qui se trouve dans le volet de droite, sélectionner **Objets principaux**, puis dans la liste de dessous, cocher la case **Développeur**



Ci-dessous l'onglet **DEVELOPPEUR** ajouté au ruban



Pour exécuter des macros sans avertissement de sécurité, cliquez sur **Sécurité des macros** dans le panneau Code. Sélectionnez l'option **Activer toutes les macros** et n'oubliez pas de cocher la case **Accès approuvé au modèle d'objet du projet VBA**

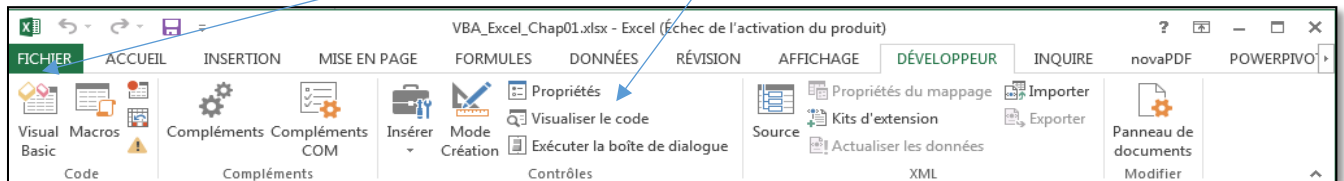


Maintenant que vous avez ajouté l'onglet Développeur au Ruban, vous pouvez faire des choses comme appeler Visual Basic Editor, exécuter des macros, enregistrer des macros et insérer des objets de formulaire dans vos feuilles de calcul. Tout d'abord, jetons un coup d'œil à l'environnement de développement Visual Basic. C'est, après tout, là où vous allez écrire tout votre code.

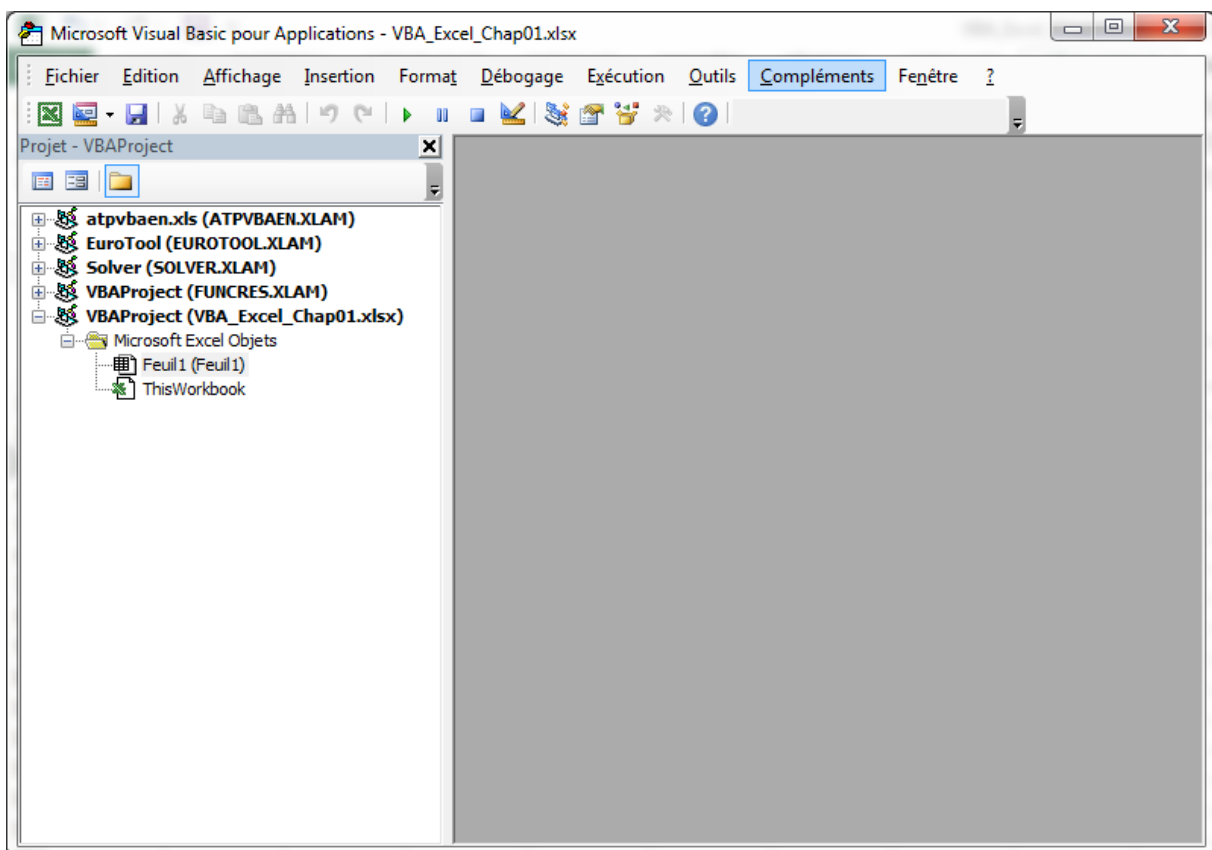
II- L'Environnement de Développement Intégré de VBA pour Excel

Pour ouvrir l'EDI de VBA Excel vous pouvez procéder de trois façons différentes:

1. Onglet **DEVELOPPEUR** → Groupe **Code** → Bouton **Visual Basic**
2. Onglet **DEVELOPPEUR** → Groupe **Code** → Bouton **Visual Basic**
3. Maintenir la touche du clavier **ALT enfoncée** → Appuyer sur la touche de fonction **F11**



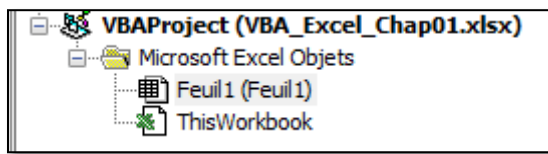
Quelle que soit la méthode que vous auriez choisie, vous devriez obtenir un écran comme celui-ci:



La fenêtre de l'IDE VBA_Excel contient deux volets:

Le volet blanc à gauche intitulé "**Project - VBA Project**". Dans ce volet, vous trouvez tous les objets que votre projet contient actuellement. (Nous expliquerons en détail ultérieurement le concept d'objet et les différents types d'objets qu'un projet peut contenir).

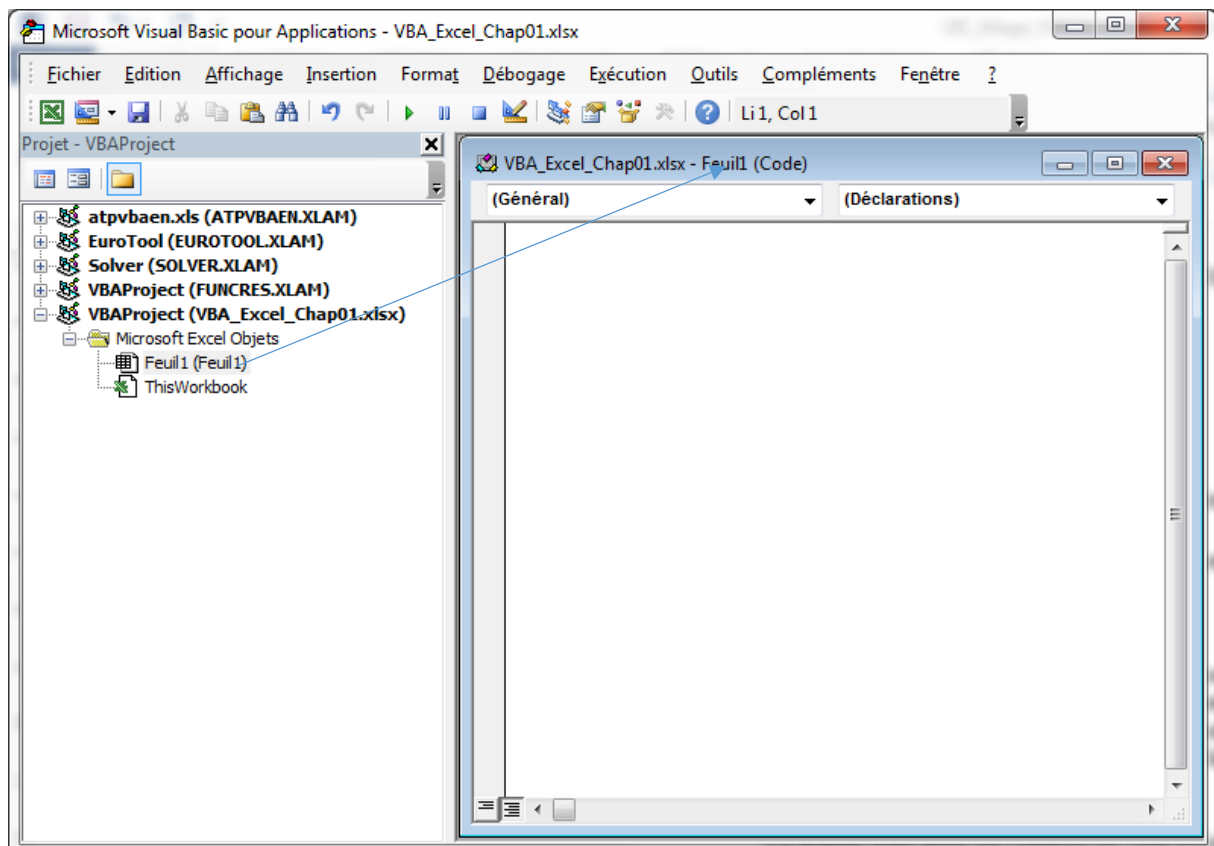
Par défaut, un nouveau projet VBA_Excel 2013 contient deux objets :



1. Un objet **Sheet** (Feuil1) qui représente un objet **WorkSheet** (Feuille de travail) d'Excel
2. Un objet **ThisWorkbook** qui représente le classeur sur lequel vous travaillez.

Un autre Objet qui n'est pas visible est le classeur "**Personal Macro Workbook**". Ce classeur apparaîtra une fois vous aurez créé une macro. **Personal Workbook** peut être utilisé pour stocker les macros qui sont souvent utilisées. Ces macros seront disponibles même si vous fermez Excel, puis ouvrez un nouveau classeur. Les objets **Sheets** sont spécifiques au classeur qui les contient.

La grande zone grise est l'endroit où vous écririez votre code. Elle est actuellement grise car aucune fenêtre de codage n'a été encore ouverte. On distingue plusieurs types de fenêtre de codage. Un premier type est celle associée à un objet **WorkSheet** (feuille de calcul). Pour ouvrir ce type de fenêtre, il suffit de "Double Cliquer" sur sa feuille WorkSheet dans le volet "**Project - VBA Project**". Nous obtenons la fenêtre ci-dessous:

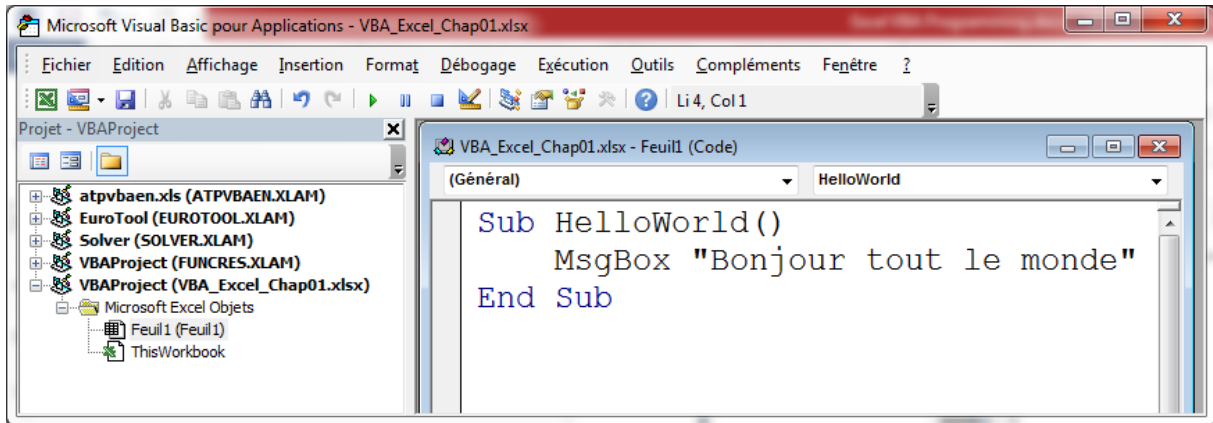


Toutes les macros que vous écrirez doivent être écrites dans une fenêtre de code. Les macros écrites dans une fenêtre de code associée à une feuille de calcul (WorkSheet) n'affecteront que cette feuille de calcul.

Pour vous donner une idée sur le code VBA, tapez dans la fenêtre intitulée "VBA_Excel_Chap01.xlsx-Feuil1(Code)" le code ci-dessous :

```
Sub HelloWorld()  
  
    MsgBox "Bonjour tout le monde"  
  
End Sub
```

Votre fenêtre de code aura l'allure ci-dessous :



Le mot **Sub** au début du code est l'abréviation de Subroutine (Sous routine ou procédure).

Une Subroutine est l'équivalent d'une procédure dans les langages Pascal/Delphi, ou une fonction qui ne retourne pas de valeur en C, C++, C#, Java, etc.

Le corps d'une Subroutine est toujours délimité par les mots **Sub** et **End Sub**.

Une Subroutine est identifiée par un nom qui est celui que vous devez mettre après le mot Sub. Ce nom doit être suivi par des parenthèses.

Le nom de la subroutine doit respecter les règles ci-dessous:

- Il ne doit pas commencer par un nombre, seul un caractère alphabétique peut être placé au début du mot. Les nombres peuvent se trouver ailleurs dans le nom.
- Les caractères point, deux points, espace, et #, \$, %, &, ! ne sont pas autorisés.

Pour exécuter le code de votre subroutine, vous pouvez procéder de trois façons différentes :

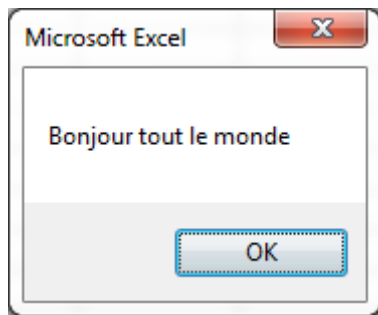
1- Cliquer sur le triangle vert de la barre d'outil de votre Editeur.:



2- Sélectionner la commande : **Exécution→Exécuter Sub/UserForm**

3- Appuyer sur le raccourci clavier : Touche **F5**

L'exécution fait apparaître la fenêtre ci-dessous.



Le mot **MsgBox** est l'abréviation de **Message Box**. C'est une subroutine intégrée VBA qui permet d'afficher une boîte de dialogue qui affiche le texte que nous avons placé entre guillemets.

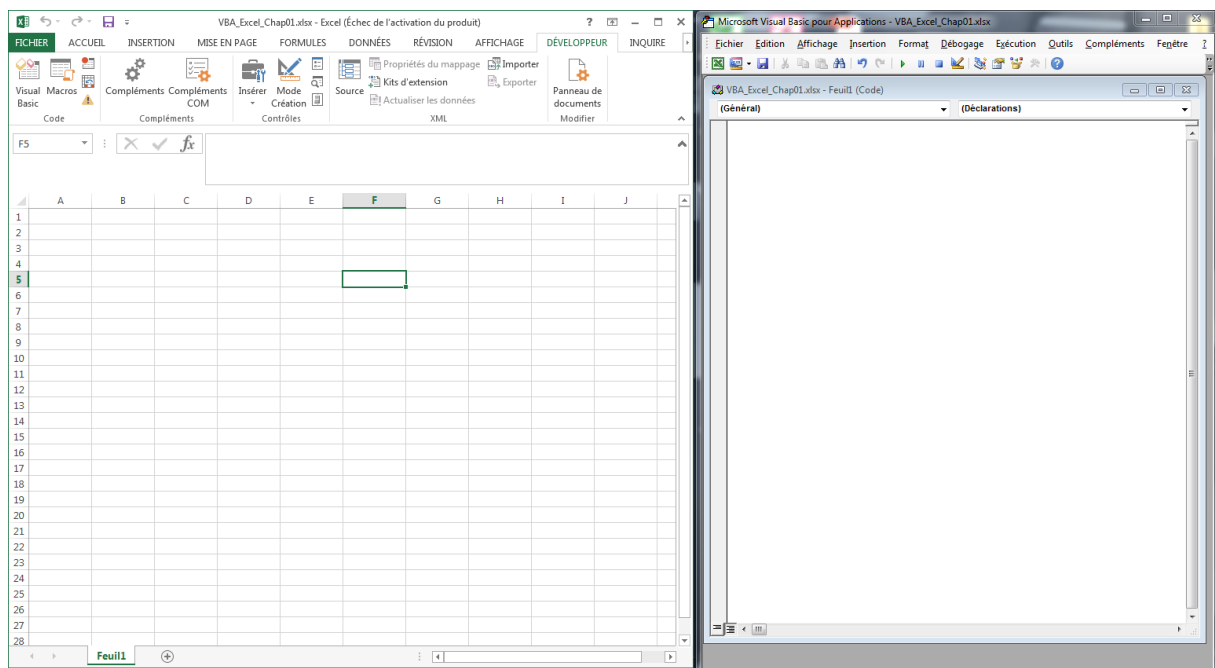
Dans la suite du cours, nous allons apprendre à enregistrer une macro et voir comment Excel traduit vos actions en code VBA d'Excel.

III- Enregistrement d'une Macro sous Excel

Il est possible de voir comment Excel traduit les actions que vous effectuez lors de l'enregistrement d'une macro en instructions VBA. Ce moyen vous permettra d'apprendre beaucoup plus facilement le langage VBA d'excel.

Le code que produit Excel est très bavard. Une fois vous aurez appris le langage, vous produirez un code beaucoup plus court.

Commencer par mettre les fenêtres Excel et de VBA côte à côte comme le montre la figure ci-dessous:



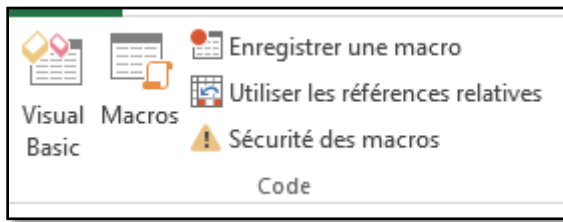
Dans ce qui suit nous allons créer une macro appelée **"AlignerDroite"** qui permet d'aligner à droite une suite de 7 prénoms saisie dans la plage A1:A7.

Pour cela commencer par saisir les prénoms suivants :

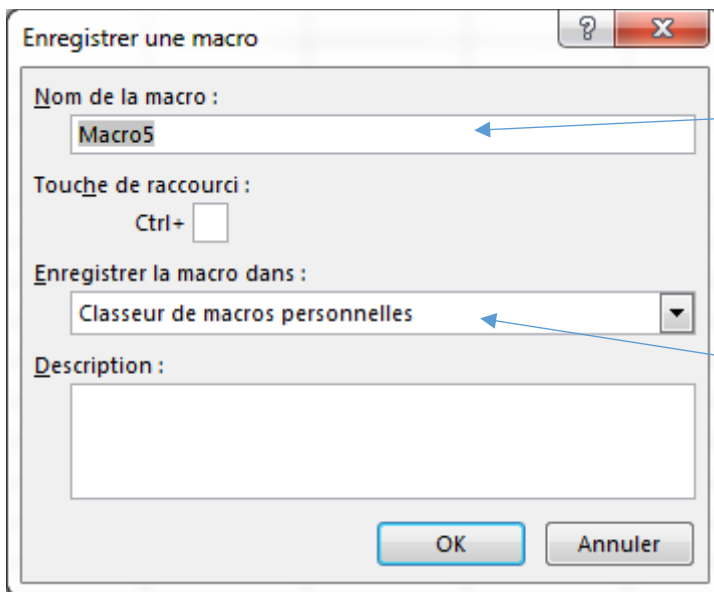
	A	B	C	D	E	F	G
1	Brahim						
2	Adil						
3	Meryem						
4	Najib						
5	Yasmine						
6	Zakaria						
7	Salma						
8							
9							
10							

Pour commencer l'enregistrement de la macro, sélectionner la commande:

Onglet **DEVELOPPEUR** → Bouton **Enregistrer une macro**



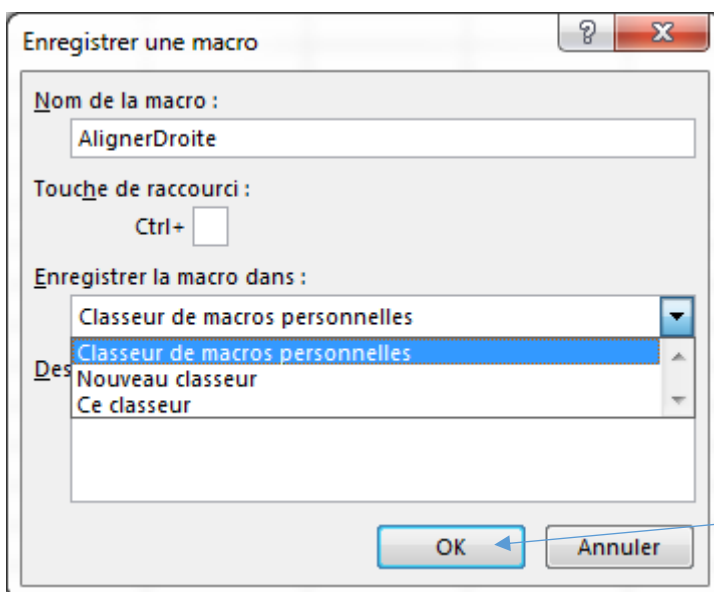
La boîte de dialogue ci-dessous apparaît:



Taper le nom de la macro dans cette zone de texte : **AlignerDroite**

dans cette zone déroulante
Sélectionner :

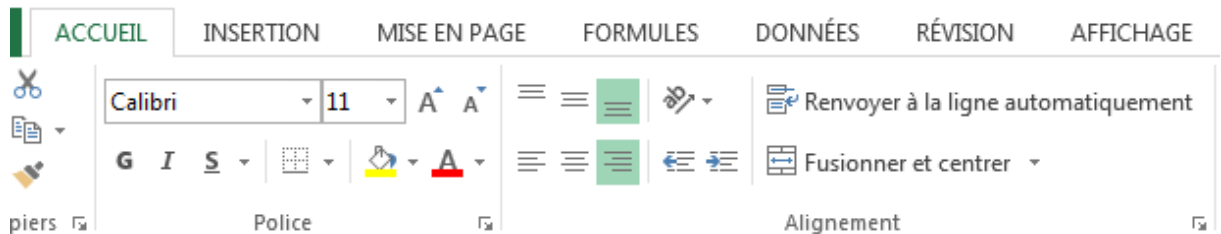
"Classeurs de macros personnelles"



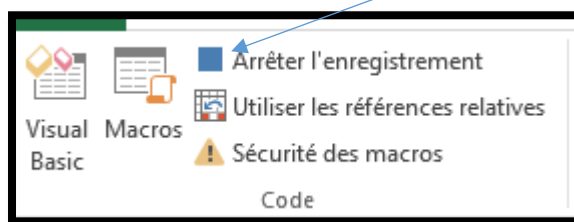
Cliquer sur le bouton **"OK"**

Actions à enregistrer lors de l'enregistrement de la macro:

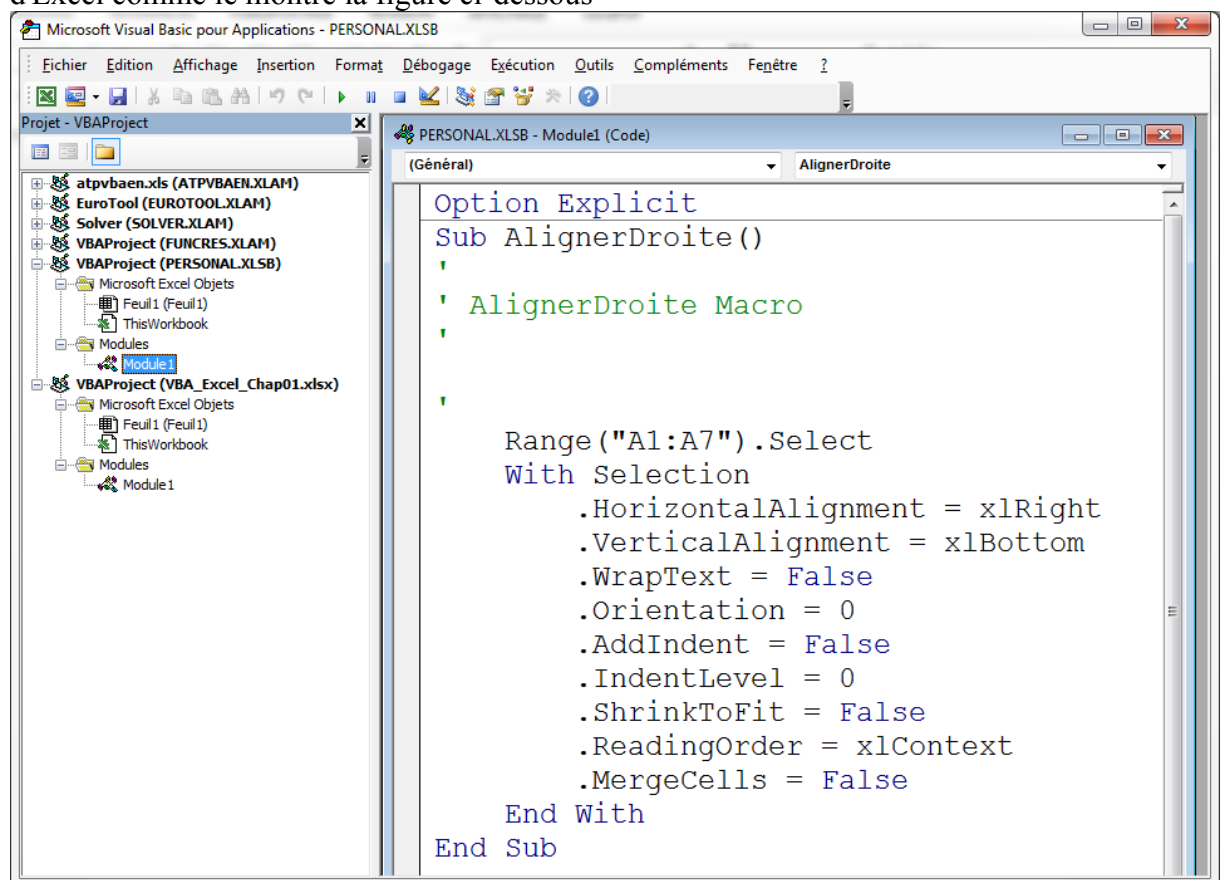
1. Sélectionner la plage de cellules **A1:A7**
2. Cliquer sur l'onglet "**ACCUEIL**"
3. Cliquer sur le bouton "**Aligner à droite**" du groupe "**Alignement**"



4. Cliquer sur l'onglet "**DEVELOPPEUR**"
5. Cliquer sur le bouton "**Arrêter l'enregistrement**"



Pour voir le code de la macro enregistré automatiquement par l'Enregistreur de macros d'Excel comme le montre la figure ci-dessous



Procédez comme suit:

1. Agrandissez la fenêtre de Visual Basic
2. Dans le volet Project-VBA Project, repérer l'objet **PERSONAL.XLSB**
3. Cliquez sur le bouton + pour dérouler puis cliquez deux fois sur **Module1** du dossier **Modules**
Le code de la macro apparaît dans la fenêtre de code

Dans la fenêtre de Visual Basic, Enregistrez le classeur des macros personnalisés.