



A1	BUREAUTIQUE	
	TP11	

Objectif : Utiliser les tableaux croisés dynamiques du logiciel Excel

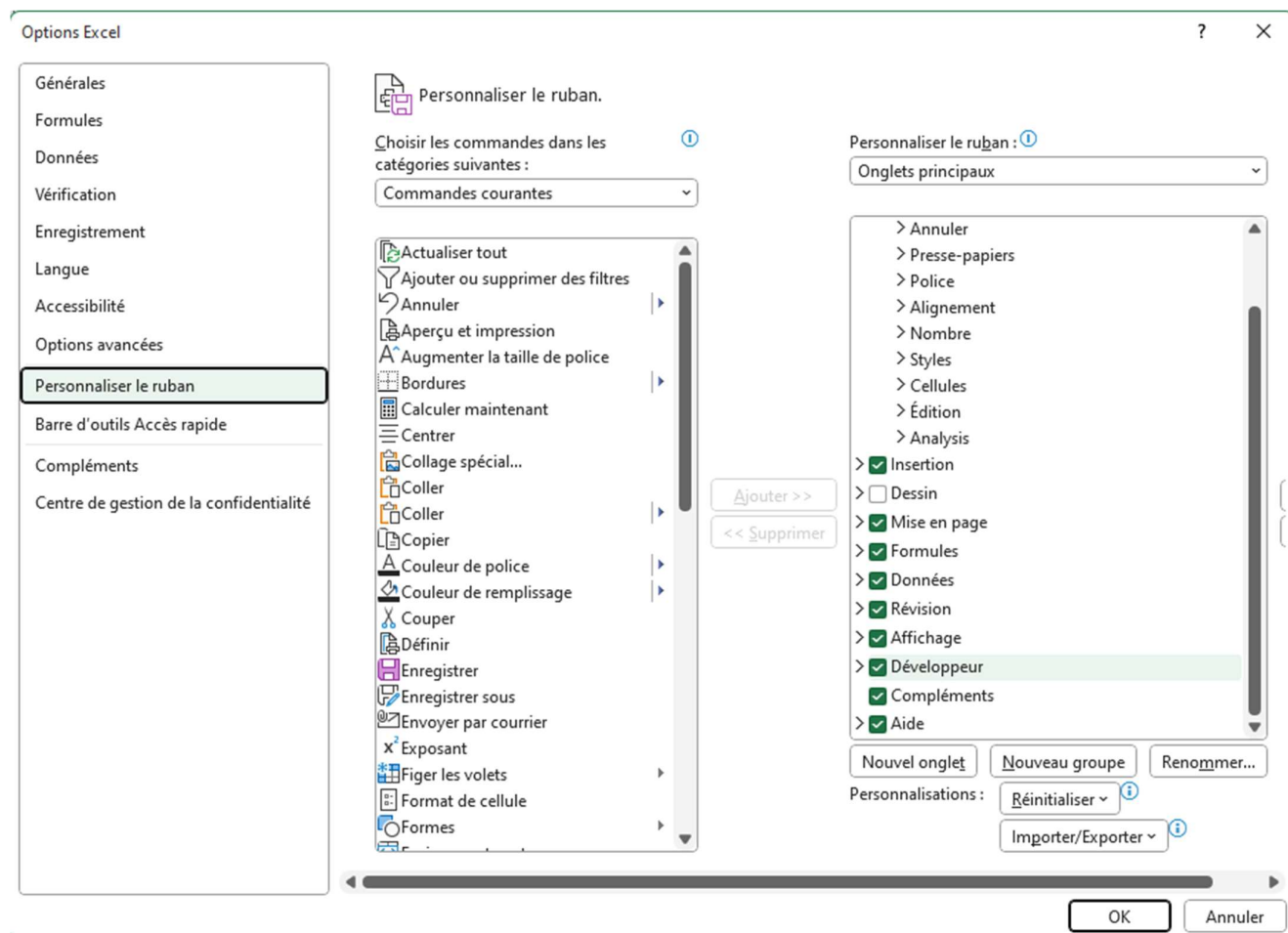
I. Démarrage

Le VBA (Visual Basic for Applications) est un langage proche du Visual Basic qui nécessite une application hôte pour s'exécuter (Excel dans notre cas).

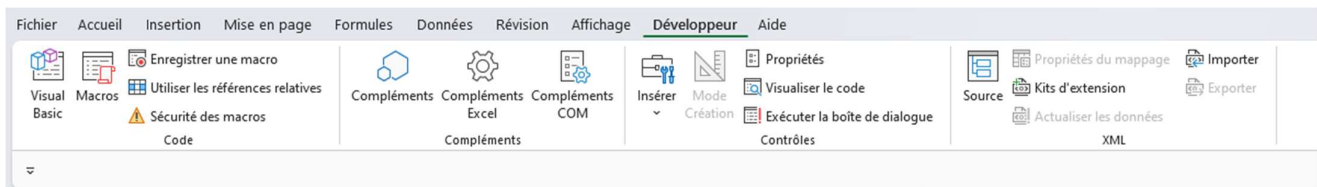
Grâce au VBA nous allons pouvoir réaliser à peu près tout ce que l'on souhaite avec Excel ...

Mais avant de démarrer, commençons par afficher les outils qui nous seront utiles.

Cliquez sur **Fichier > Options > Personnaliser le Ruban** puis cochez **Développeur**.

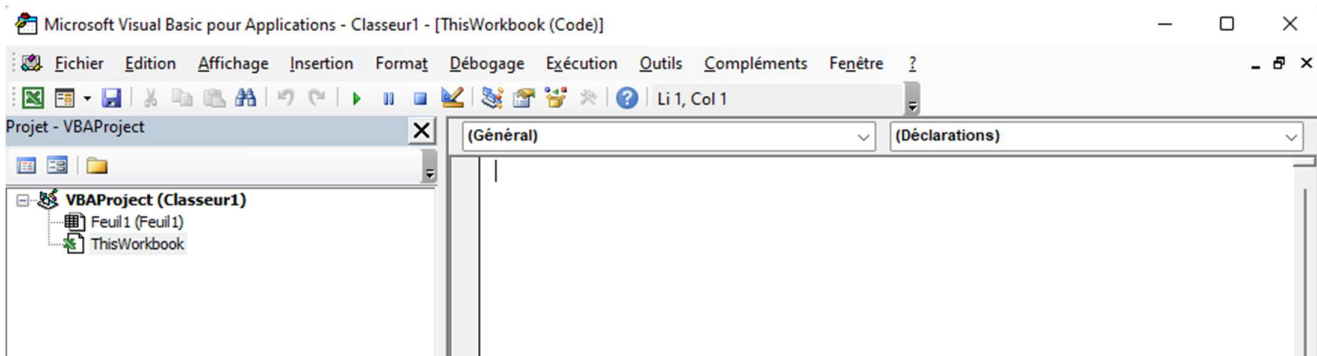


Un nouvel onglet sera ajouté :



Pour travailler avec du code VBA, nous avons besoin d'un éditeur, celui-ci est déjà installé et vous pouvez l'ouvrir avec le raccourci Alt + F11 :

Si Alt + F11 ne fonctionne pas, sélectionner **Visual Basic**



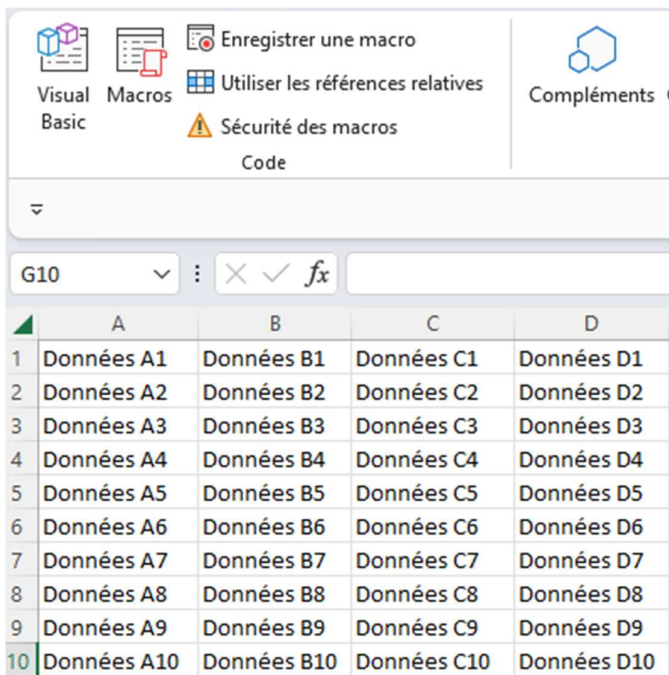
Nous y reviendrons, reprenez simplement le raccourci Alt + F11 pour le moment ...

II. Première macro

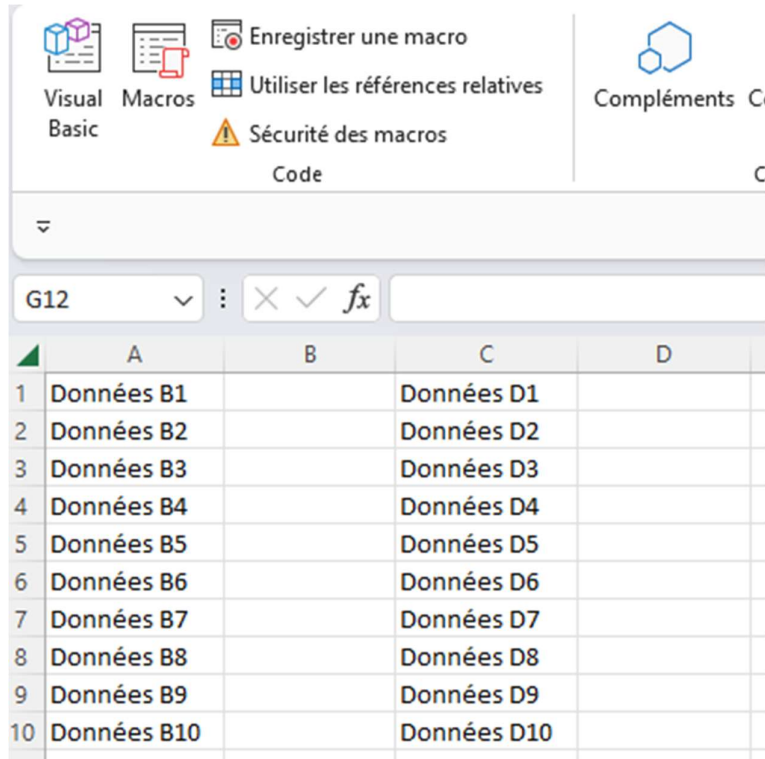
Il est possible d'automatiser certaines tâches en toute simplicité grâce à l'enregistreur de macros.

Pour prendre un exemple simple, nous allons automatiser les opérations suivantes :

- Effacer le contenu des colonnes A et C
- Déplacer le contenu de la colonne B dans la colonne A
- Déplacer le contenu de la colonne D dans la colonne C

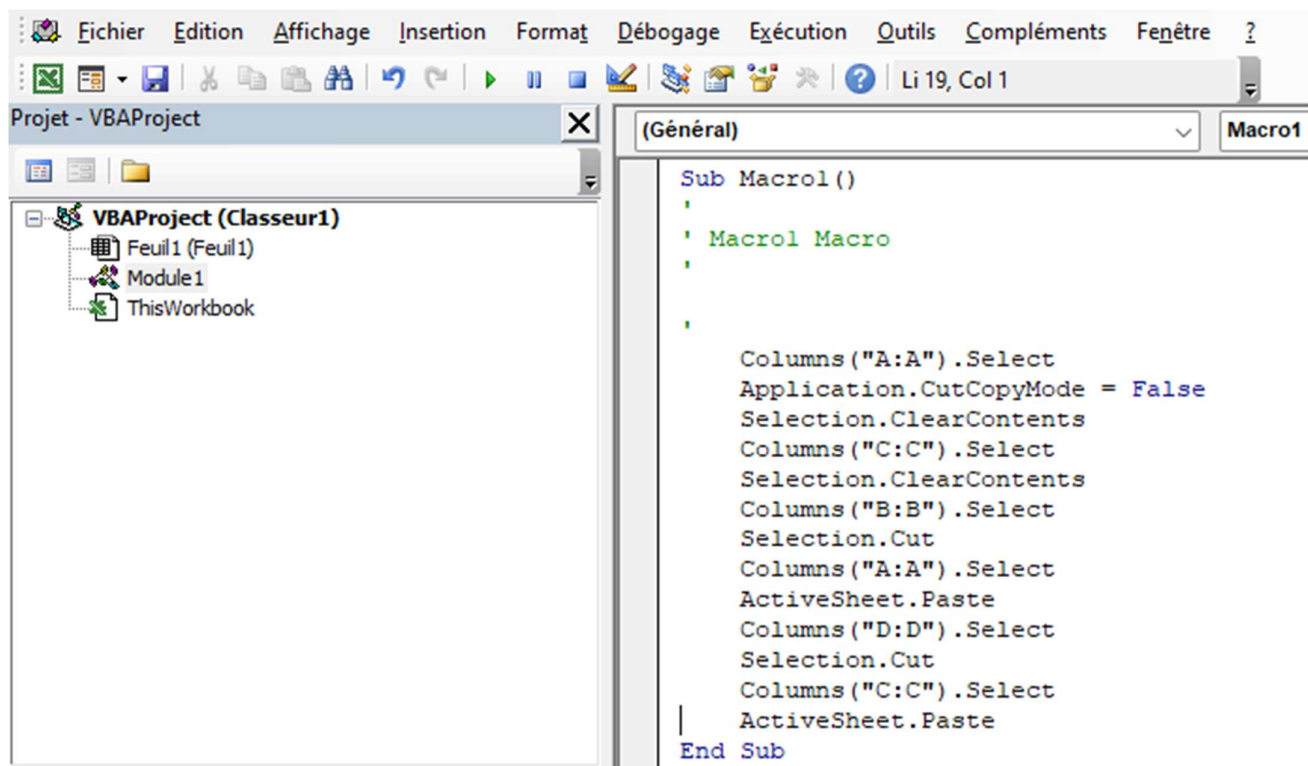


Pour ce faire, cliquez sur Enregistrer une macro puis sur Ok, exécutez les opérations décrites ci-dessus sans interruption (car toutes les manipulations sont enregistrées) et pour terminer cliquez sur Arrêter l'enregistrement.



Excel a enregistré vos manipulations et les a traduites en code VBA.

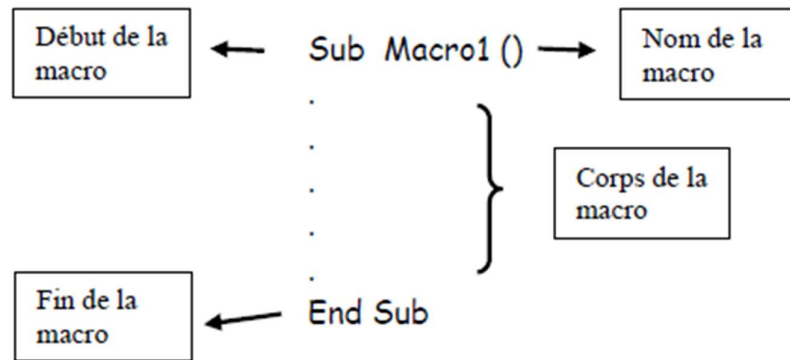
Pour voir votre macro, ouvrez l'éditeur (Alt + F11) et cliquez sur Module1 :



Ce code correspond aux manipulations enregistrées.

Nous allons nous arrêter quelques instants sur le code généré :

La macro est identifiée et délimitée



Le symbole ' indique un commentaire (sans action sur le programme) ; il s'écrit en vert

```
Sub Macro1()
'
' Macro1 Macro
'
'
    Columns("A:A").Select
    Selection.ClearContents
    Columns("C:C").Select
    Selection.ClearContents
    Columns("B:B").Select
    Selection.Cut Destination:=Columns("A:A")
    Columns("D:D").Select
    Selection.Cut Destination:=Columns("C:C")
    Columns("C:C").Select
End Sub
```

Sub et End Sub délimitent le début et la fin de la macro, Macro1 correspond au nom de cette macro :

```
Sub Macro1()
End Sub
```

Nous allons maintenant modifier le nom de cette macro et lui attribuer un nom qui soit un peu plus parlant. Pour cela, remplacez simplement Macro1 par manipulationsDesColonnes (le nom ne doit pas contenir d'espaces) :

```
Sub manipulationsDesColonnes()
```

Le texte en vert (texte précédé d'une apostrophe) est un commentaire, il n'est pas pris en compte à l'exécution du code :

```
'
' Macro1 Macro
'
```

Les commentaires sont très utiles pour s'y retrouver lorsque l'on dispose de beaucoup de code ou pour ne pas exécuter certaines lignes de code sans pour autant les supprimer.

```

Sub manipulationsDesColonnes()

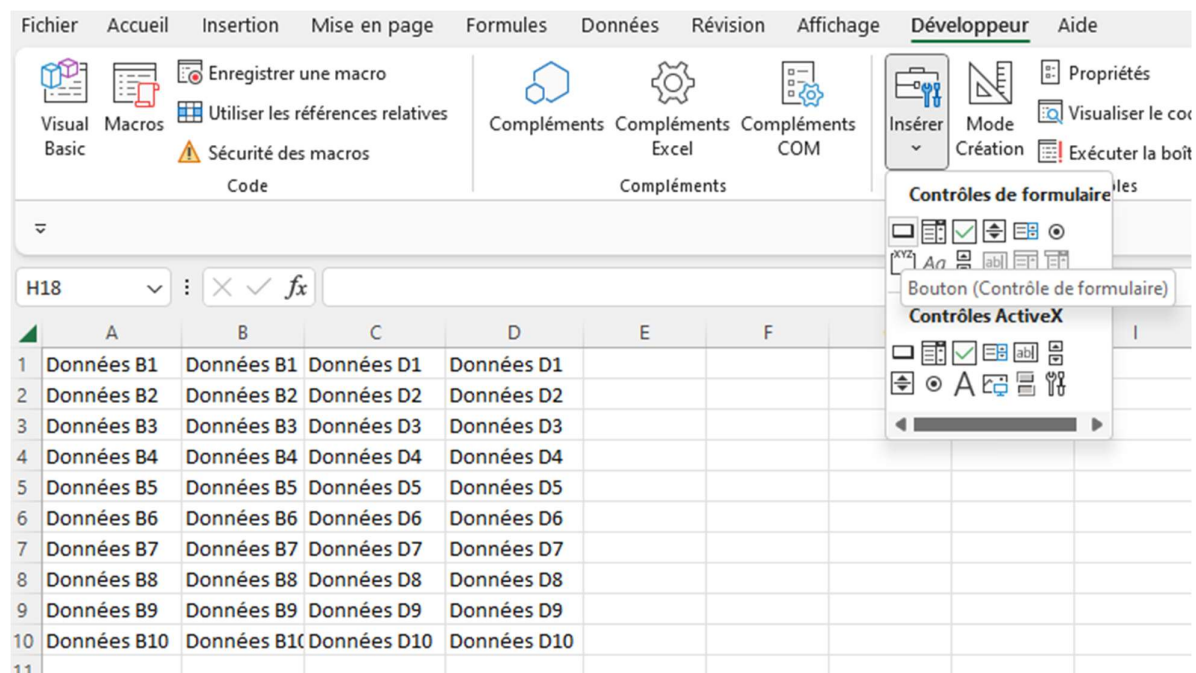
    'Mon premier commentaire !

    Columns("A:A").Select
    Selection.ClearContents
    Columns("C:C").Select
    Selection.ClearContents
    Columns("B:B").Select
    Selection.Cut Destination:=Columns("A:A")
    Columns("D:D").Select
    Selection.Cut Destination:=Columns("C:C")
    Columns("C:C").Select
End Sub

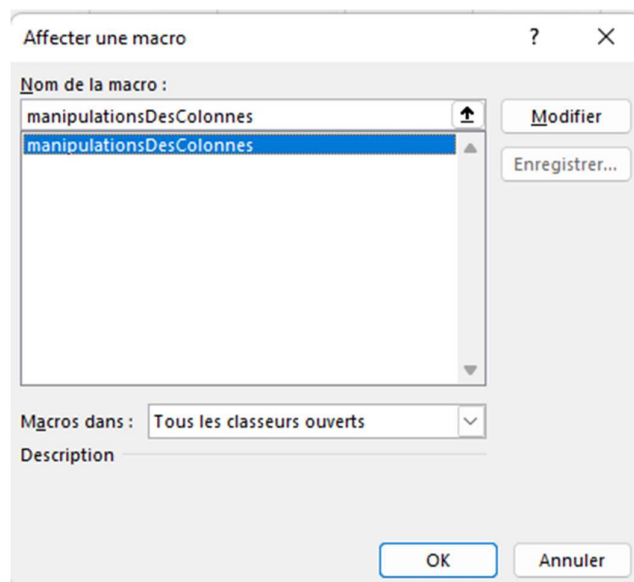
```

Nous souhaitons maintenant que cette macro s'exécute en cliquant sur un bouton.

Insérez un bouton en cliquant sur Insérer Bouton (Contrôles de formulaires) :



Tracez votre bouton et sélectionnez ensuite simplement votre macro :



Lorsque vous cliquerez sur le bouton, la macro sera exécutée :

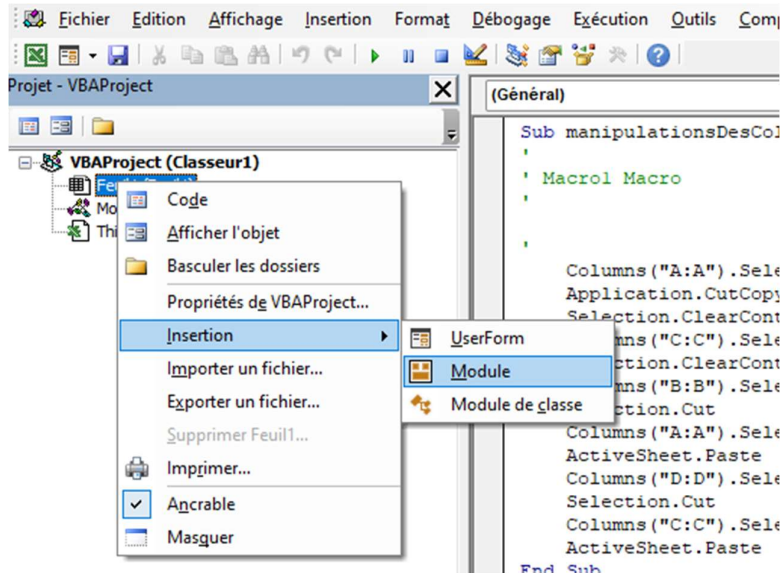
	A	B	C	D	E	F	G
1	Données B1		Données D1				
2	Données B2		Données D2				
3	Données B3		Données D3				
4	Données B4		Données D4		Bouton 1		
5	Données B5		Données D5				
6	Données B6		Données D6				
7	Données B7		Données D7				
8	Données B8		Données D8				
9	Données B9		Données D9				
10	Données B10		Données D10				
11							

III. Les feuilles et les cellules

A.Sélections

Nous allons créer une macro qui sélectionnera une cellule de notre choix.

Ouvrez l'éditeur et ajoutez-y un module :

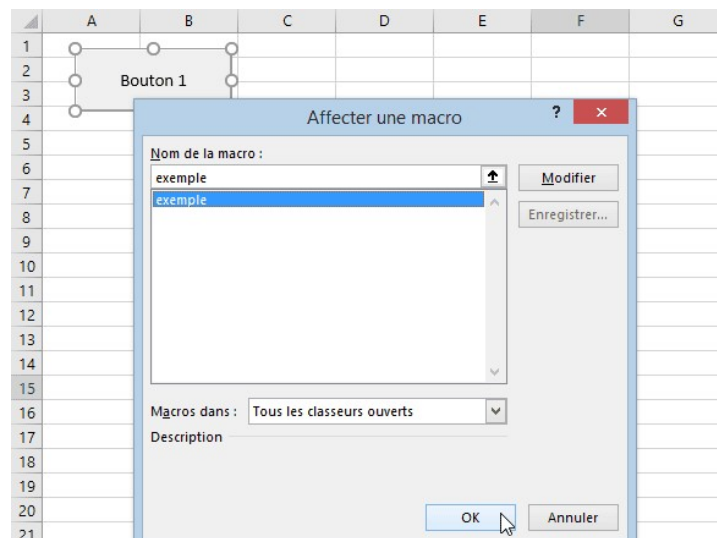


Dans le module, tapez « sub exemple » et appuyez sur Entrée.

Vous remarquerez qu'Excel a automatiquement ajouté la fin de cette nouvelle procédure :

```
Sub exemple()  
  
End Sub
```

Créez maintenant un bouton de formulaire auquel vous allez associer cette macro (vide pour le moment) :



Complétez votre macro avec ceci :

```
Sub exemple()  
  
    'Sélection de la cellule A8  
    Range("A8").Select  
  
End Sub
```

Vous pouvez tester cette macro en cliquant sur votre bouton de formulaire, la cellule A8 est alors sélectionnée.

Nous allons maintenant modifier cette macro pour sélectionner la cellule A8 de la seconde feuille :

```
Sub exemple()  
  
    'Activation de la feuille 2  
    Sheets("Feuil2").Activate  
  
    'Sélection de la cellule A8  
    Range("A8").Select  
  
End Sub
```

Excel active alors la feuille 2 avant de sélectionner la cellule A8.

Aidez-vous des commentaires (texte en vert) pour bien comprendre les macros de ce cours.

SÉLECTION D'UNE PLAGE DE CELLULES

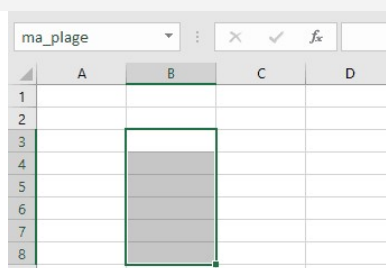
```
Sub exemple()  
  
    'Sélection des cellules A1 à A8  
    Range("A1:A8").Select  
  
End Sub
```

SÉLECTION DE CELLULES DISTINCTES

```
Sub exemple()  
  
    'Sélection des cellule A8 et C5  
    Range("A8, C5").Select  
  
End Sub
```

SÉLECTION D'UNE PLAGE DE CELLULES NOMMÉE

```
Sub exemple()  
  
    'Sélection des cellules de la plage "ma_plage"  
    Range("ma_plage").Select  
  
End Sub
```



SÉLECTION D'UNE CELLULE EN FONCTION D'UN NUMÉRO DE LIGNE ET DE COLONNE

```
Sub exemple()  
  
    'Sélection de la cellule de la ligne 8 et de la colonne 1  
    Cells(8, 1).Select  
  
End Sub
```

Cette autre manière de sélectionner permet des sélections plus dynamiques et sera bien utile par la suite.

En voici un petit exemple :

```
Sub exemple()  
  
    'Sélection aléatoire d'une cellule de la ligne 1 à 10 et de la colonne 1  
    Cells(Int(Rnd * 10) + 1, 1).Select  
  
    'Traduction :  
    'Cells([nombre_aléatoire_entre_1_et_10], 1).Select  
  
End Sub
```

Ici, le numéro de ligne est `Int(Rnd * 10) + 1`, autrement dit un nombre entre 1 et 10 (inutile de retenir ce code pour le moment).

SÉLECTION DE LIGNES

Il est possible de sélectionner des lignes entières avec `Range` ou `Rows` (`Rows` étant spécifique aux lignes) :

```
Sub exemple()  
  
    'Sélection des lignes 2 à 6  
    Range("2:6").Select  
  
End Sub  
Sub exemple()  
  
    'Sélection des lignes 2 à 6  
    Rows("2:6").Select  
  
End Sub
```

SÉLECTION DE COLONNES

Tout comme pour les lignes, il est possible de sélectionner des colonnes entières avec `Range` ou `Columns` (`Columns` étant spécifique aux colonnes) :

```
Sub exemple()  
  
    'Sélection des colonnes B à G  
    Range("B:G").Select  
  
End Sub  
Sub exemple()  
  
    'Sélection des colonnes B à G  
    Columns("B:G").Select  
  
End Sub
```

B. Propriétés

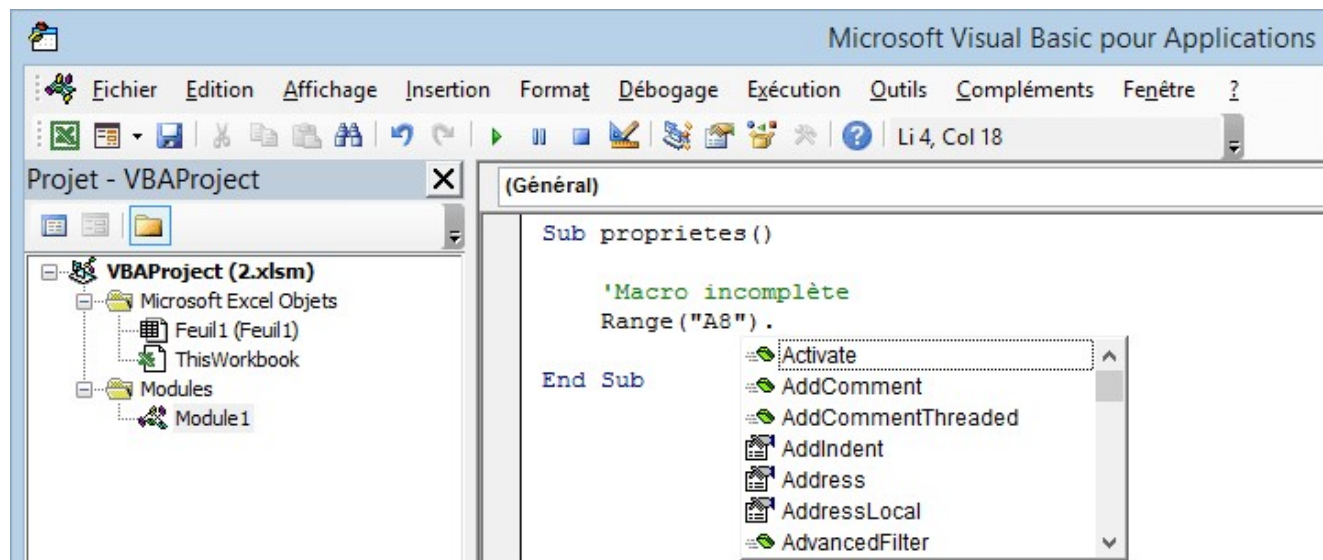
Nous allons maintenant agir sur le contenu et l'apparence des cellules et des feuilles.

Commencez par ouvrir l'éditeur, ajoutez-y un module, copiez la macro ci-dessous et associez-la à un bouton de formulaire (voir précédemment) :

```
Sub proprietes()  
  
    'Macro incomplète  
    Range("A8")  
  
End Sub
```

Nous voulons effectuer une action sur la cellule A8 avec ce début de macro.

Pour afficher la liste des possibilités que l'on peut associer à l'objet Range, ajoutez un . après Range("A8") :



L'éditeur affiche alors les différentes possibilités ...

Pour ce premier exemple, cliquez sur Value puis appuyez sur la touche Tab pour valider ce choix :

```
Sub proprietes()  
  
    'Macro incomplète  
    Range("A8").Value  
  
End Sub
```

La propriété Value représente ici le contenu de la cellule.

Nous voulons maintenant donner la valeur 48 à la cellule A8 :

```
Sub proprietes()  
  
    'Cellule A8 = 48  
    Range("A8").Value = 48  
  
    'Traduction :  
    'La valeur de la cellule A8 est désormais : 48  
  
End Sub
```

Puis, la valeur Exemple de texte à A8 (le texte doit être mis entre " ") :

```
Sub proprietes()  
  
    'Cellule A8 = Exemple de texte  
    Range("A8").Value = "Exemple de texte"  
  
End Sub
```

Dans ce cas, c'est bien la cellule A8 de la feuille où est lancée la procédure (ici, celle où se trouve le bouton formulaire) qui sera modifiée.

Si vous créez un second bouton sur la feuille 2, ce sera alors la cellule A8 de la feuille 2 qui sera modifiée.

Pour modifier la cellule A8 de la feuille 2 en cliquant sur le bouton de la feuille 1, il faut préciser le nom de la feuille en ajoutant Sheets("Nom_de_la_feuille") avant Range :

```
Sub proprietes()  
  
    'Cellule A8 de la feuille 2 = Exemple de texte  
    Sheets("Feuil2").Range("A8").Value = "Exemple de texte"  
  
End Sub
```

De même, si l'on souhaite modifier la cellule A8 de la feuille 2 d'un autre classeur ouvert, il faut préciser le nom du classeur en début de ligne à l'aide de Workbooks("Nom_du_fichier") :

```
Sub proprietes()  
  
    'Cellule A8 de la feuille 2 du classeur 2 = Exemple de texte  
    Workbooks("Classeur2.xlsx").Sheets("Feuil2").Range("A8").Value = "Exemple de texte"  
  
End Sub
```

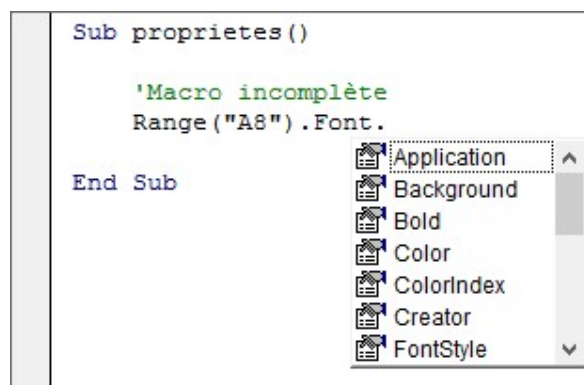
Bien que Value ait été utilisé pour illustrer ces différents exemples, il n'est pas nécessaire de l'indiquer, car c'est automatiquement la valeur de la cellule qui est modifiée si rien n'est précisé.

Ces 2 lignes génèrent un résultat identique :

```
Range("A8").Value = 48  
Range("A8") = 48
```

C. MISE EN FORME DU TEXTE

Après avoir sélectionné la propriété Font et ajouté un ., la liste des propriétés que l'on peut attribuer à la mise en forme du texte apparaît :



La modification des couleurs sera détaillée un peu plus loin

D. MISE EN FORME : TAILLE DU TEXTE

```
Sub proprietes()  
  
    'Modifier la taille du texte des cellules A1 à A8  
    Range("A1:A8").Font.Size = 18  
  
End Sub
```

E. MISE EN FORME : TEXTE EN GRAS

```
Sub proprietes()  
  
    'Mettre en gras les cellules A1 à A8  
    Range("A1:A8").Font.Bold = True  
  
End Sub
```

Bold = True signifie Caractères en gras = Oui.

Pour retirer la mise en forme Bold à un texte, il faut donc remplacer Oui par Non, autrement dit, True par False :

```
Sub proprietes()  
  
    'Enlever la mise en forme "gras" des cellules A1 à A8  
    Range("A1:A8").Font.Bold = False  
  
End Sub
```

F. MISE EN FORME : TEXTE EN ITALIQUE

```
Sub proprietes()  
  
    'Mettre en italique les cellules A1 à A8  
    Range("A1:A8").Font.Italic = True  
  
End Sub
```

G. MISE EN FORME : TEXTE SOULIGNÉ

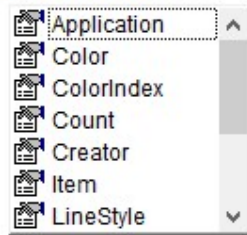
```
Sub proprietes()  
  
    'Souligner les cellules A1 à A8  
    Range("A1:A8").Font.Underline = True  
  
End Sub
```

H. MISE EN FORME : POLICE

```
Sub proprietes()  
  
    'Modifier la police de caractères des cellules A1 à A8  
    Range("A1:A8").Font.Name = "Arial"  
  
End Sub
```

I. AJOUTER DES BORDURES

```
Sub proprietes()  
  
    'Ajouter une bordure aux cellules A1 à A8  
    Range("A1:A8").Borders.  
  
End Sub
```



```
Sub proprietes()  
  
    'Ajouter une bordure aux cellules A1 à A8  
    Range("A1:A8").Borders.Value = 1  
  
    'Value = 0 : pas de bordure  
  
End Sub
```

J. MODIFIER LA MISE EN FORME DE LA SÉLECTION ACTUELLE

```
Sub proprietes()  
  
    'Ajouter une bordure aux cellules sélectionnées  
    Selection.Borders.Value = 1  
  
End Sub
```

K. FUSIONNER DES CELLULES

```
Sub proprietes()  
  
    'Fusionner des cellules particulières  
    Range("B1:B8").Merge  
  
End Sub
```

```
Sub proprietes()  
  
    'Fusionner une sélection de cellules  
    Selection.Merge  
  
End Sub
```

L. MODIFIER LES PROPRIÉTÉS D'UNE FEUILLE

```
Sub proprietes()  
  
    'Masquer une feuille  
    Sheets("Feuil3").Visible = 2  
  
    'Visible = -1 : afficher la feuille  
  
End Sub
```

N'oubliez pas que seule une toute petite partie des possibilités de personnalisation sont indiquées ici. Si la propriété dont vous avez besoin n'est pas détaillée ici, n'ayez pas peur d'utiliser l'enregistreur de macro pour vous éviter de longues recherches (en enregistrant la manipulation dont vous avez besoin, vous pourrez retrouver plus facilement la propriété recherchée pour pouvoir ensuite l'utiliser dans votre macro).

M. MODIFIER LA VALEUR D'UNE CELLULE EN FONCTION D'UNE AUTRE

	A	B	C	D	E
1	Ma valeur				
2					
3					
4					
5					
6					
7	Cible				

L'objectif ici est que A7 prenne la valeur de A1, ce qui nous donne :

```
Sub proprietes()  
  
    'A7 = A1  
    Range("A7") = Range("A1")  
  
    'Ou :  
    'Range("A7").Value = Range("A1").Value  
  
End Sub
```

Ou pour copier par exemple la taille du texte :

```
Sub proprietes()  
  
    Range("A7").Font.Size = Range("A1").Font.Size  
  
End Sub
```

Ce qui est à gauche du = prend la valeur de ce qui est à droite du =.

N. MODIFIER LA VALEUR D'UNE CELLULE EN FONCTION DE SA PROPRE VALEUR

Nous allons maintenant créer ici un compteur de clics.

A chaque clic, la valeur de A1 sera augmentée de 1 :

```
Sub proprietes()  
  
    'Compteur de clics en A1  
    Range("A1") = Range("A1") + 1  
  
End Sub
```

Cette ligne ne doit pas être interprétée comme une opération mathématique (rappelez-vous que ce qui est à gauche du = prend la valeur de ce qui est à droite du =).

Excel exécute le code ligne par ligne en respectant certaines priorités, ces commentaires devraient vous aider à mieux comprendre ce même code :

```

'Pour cet exemple : A1 vaut 10 avant l'exécution du code
Sub proprietes()

    'Un clic a été fait sur le bouton, nous entrons dans la procédure
    'Pour le moment A1 vaut encore 10
    'Pendant l'exécution de la ligne ci-dessous :
    '- la valeur à droite du = est calculée en priorité (A1 vaut toujours 10, cela
donne 10 + 1)
    '- après calcul, la valeur à droite du = vaut donc 11
    '- A1 prend ensuite la valeur à droite du = (soit la valeur 11)
    Range("A1") = Range("A1") + 1
    'A1 vaut alors 11 seulement après l'exécution de la ligne de code

End Sub

```

O.WITH

Ce code permet de définir différentes propriétés à la cellule A8 de la feuille 2 :

```

Sub proprietes()

    Sheets("Feuil2").Range("A8").Borders.Weight = 3
    Sheets("Feuil2").Range("A8").Font.Bold = True
    Sheets("Feuil2").Range("A8").Font.Size = 18
    Sheets("Feuil2").Range("A8").Font.Italic = True
    Sheets("Feuil2").Range("A8").Font.Name = "Arial"

End Sub

```

Nous pouvons utiliser With pour éviter les répétitions de Sheets("Feuil2").Range("A8") :

```

Sub proprietes()
    'Début de l'instruction avec : With
    With Sheets("Feuil2").Range("A8")
        .Borders.Weight = 3
        .Font.Bold = True
        .Font.Size = 18
        .Font.Italic = True
        .Font.Name = "Arial"
    'Fin de l'instruction avec : End With
    End With

End Sub

```

Sheets("Feuil2").Range("A8") n'est donc plus répété.

Bien que ce ne soit pas indispensable dans ce cas, il est également possible de faire de même pour .Font, ce qui nous donnerait :

```

Sub proprietes()

    With Sheets("Feuil2").Range("A8")
        .Borders.Weight = 3
        With .Font
            .Bold = True
            .Size = 18
            .Italic = True
            .Name = "Arial"
        End With
    End With

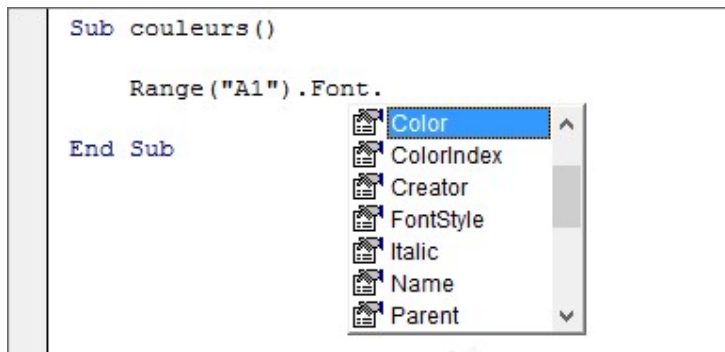
End Sub

```

P.COULEURS

Nous allons commencer par attribuer une couleur au texte en A1.

Après avoir ajouté Font., nous obtenons :



Nous avons 2 possibilités pour définir la couleur : ColorIndex et ses 56 couleurs ou Color qui nous permettra d'utiliser n'importe quelle couleur.

Q.COLORINDEX

Voici les 56 couleurs disponibles avec ColorIndex :

	A	B	C	D	E	F	G	H	I	J	K	L
1	1	2	3	4	5	6	7	8				
2	9	10	11	12	13	14	15	16				
3	17	18	19	20	21	22	23	24				
4	25	26	27	28	29	30	31	32				
5	33	34	35	36	37	38	39	40				
6	41	42	43	44	45	46	47	48				
7	49	50	51	52	53	54	55	56				

Pour appliquer à notre texte l'une de ces 56 couleurs, nous écrirons :

```
Sub couleurs()  
    'Couleur du texte en A1 : vert (couleur 10)  
    Range("A1").Font.ColorIndex = 10  
End Sub
```


Ce qui nous donne :

	A	B	C	D	E	F
1	Exemple					
2						
3						
4						

R. COLOR

Voici un exemple similaire avec Color :

```
Sub couleurs()  
  
    'Couleur du texte en A1 : RGB(50, 200, 100)  
    Range("A1").Font.Color = RGB(50, 200, 100)  
  
End Sub
```

La couleur ici est RGB(50, 200, 100).

RGB en français signifie **RVB** (Rouge Vert Bleu), les valeurs vont de 0 à 255 pour chaque couleur.

Quelques exemples de couleurs pour mieux comprendre :

- RGB(0, 0, 0) : noir
- RGB(255, 255, 255) : blanc
- RGB(255, 0, 0) : rouge
- RGB(0, 255, 0) : vert
- RGB(0, 0, 255) : bleu

Heureusement pour nous, il existe différentes solutions qui nous permettent de trouver facilement les valeurs RGB de la couleur qui nous intéresse.

Vous trouverez par exemple une liste de valeurs RGB sur la page suivante : [liste de valeurs RGB](#).

Pour donner une couleur violette à notre texte, nous pouvons donc rechercher les valeurs RGB de cette couleur sur la liste de couleurs et entrer :

```
Sub couleurs()  
  
    'Couleur du texte en A1 : RGB(192, 32, 255)  
    Range("A1").Font.Color = RGB(192, 32, 255)  
  
End Sub
```

Ce qui nous donne :

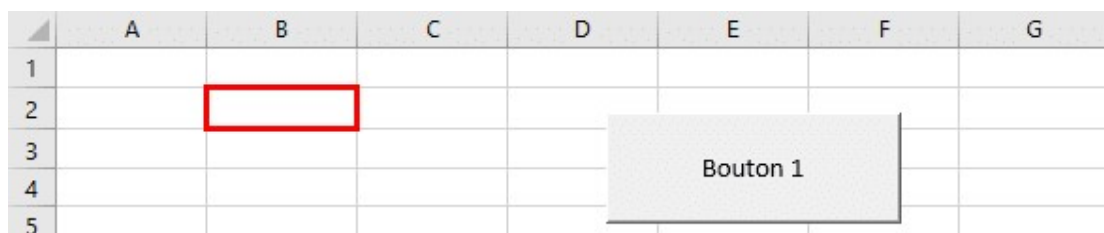
	A	B	C	D	E	F
1	Exemple					
2						
3						
4						

S. CRÉER UNE BORDURE COLORÉE

Nous allons créer une macro qui va ajouter une bordure à la cellule active avec ActiveCell.

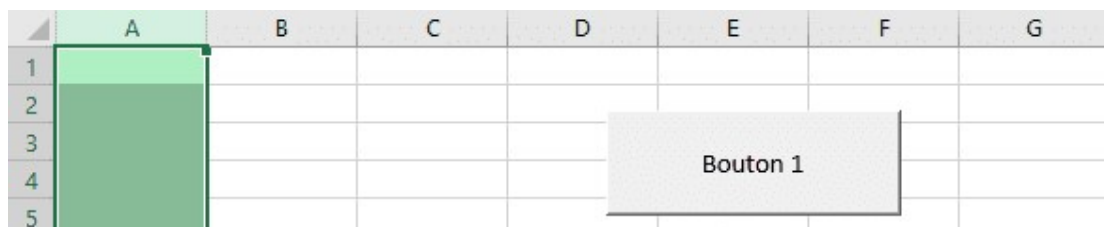
La bordure sera rouge et épaisse :

```
Sub couleurs()  
    'Epaisseur de la bordure  
    ActiveCell.Borders.Weight = 4  
  
    'Couleur de la bordure : rouge  
    ActiveCell.Borders.Color = RGB(255, 0, 0)  
End Sub
```



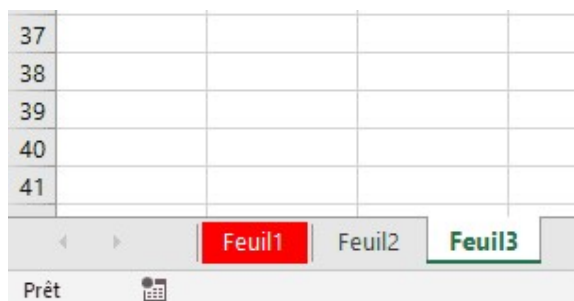
T. COLORER LE FOND DES CELLULES SÉLECTIONNÉES

```
Sub couleurs()  
    'Colorer le fond des cellules sélectionnées  
    Selection.Interior.Color = RGB(174, 240, 194)  
End Sub
```



U. COLORER L'ONGLET D'UNE FEUILLE

```
Sub couleurs()  
    'Colorer l'onglet de la feuille "Feuil1"  
    Sheets("Feuil1").Tab.Color = RGB(255, 0, 0)  
End Sub
```



Exercise:

Une extraction brute d'une base de données est fournie sous une forme générique.

Vous devez automatiser le traitement pour faire une remise en forme similaire à l'exemple fourni ci-dessous.

L'utilisation de votre document sera la suivante :

- Cliquer sur un bouton de mise en forme pour obtenir la transformation en feuille 2

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1														
2				NOM :		DURAND				Bac :	S.SI			
3										Mention :	AB			
4				Prénom :		Pierre								
5														
6														
7		Semestre 1			Semestre 2			Semestre 3			Semestre 4			
8		Science	Général	Technique	Science	Général	Technique	Science	Général	Technique	Science	Général	Technique	
9		11	12	10	12	12	11	13	12	13	12	13	14	
10		Moyen. S1			Moyen. S2			Moyen. S3			Moyen. S4			
11		10,75			11,5			12,75			13,25			
12														
13		Moyen. A1			11,125						Moyen. A2			13
14														
15								Moyen.		12,063				
16														
17														