

VBA : Les macros

Laure Tougne

Plan du cours

- Présentation de VBA
- Créer, exécuter et enregistrer une macro
- Modifier le code d'une macro
- Affecter une macro

PRÉSENTATION DE VBA

Introduction

- VBA : **Visual Basic pour Application**
- Langage commun à toutes les applications de la suite Microsoft Office (Word, Access, Excel, Outlook, Powerpoint)
- Dans le cadre de ce cours : **Excel**

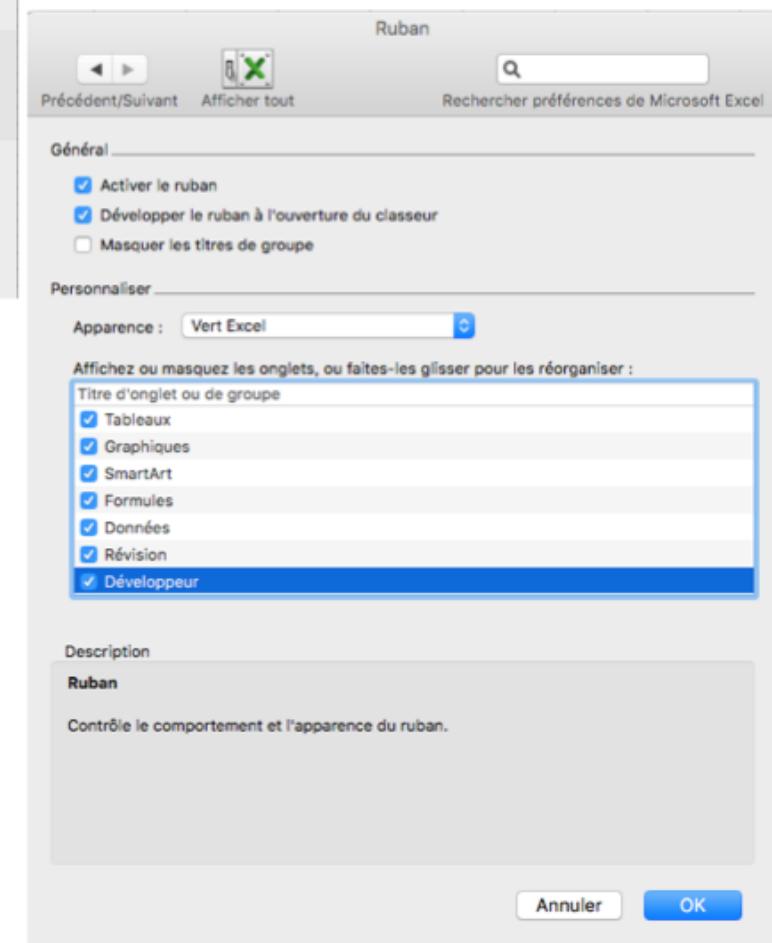
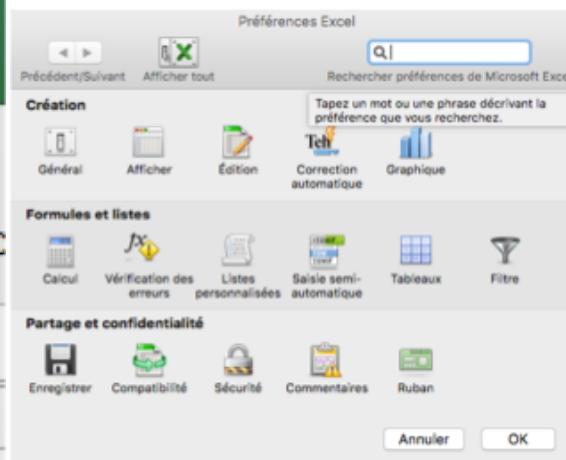
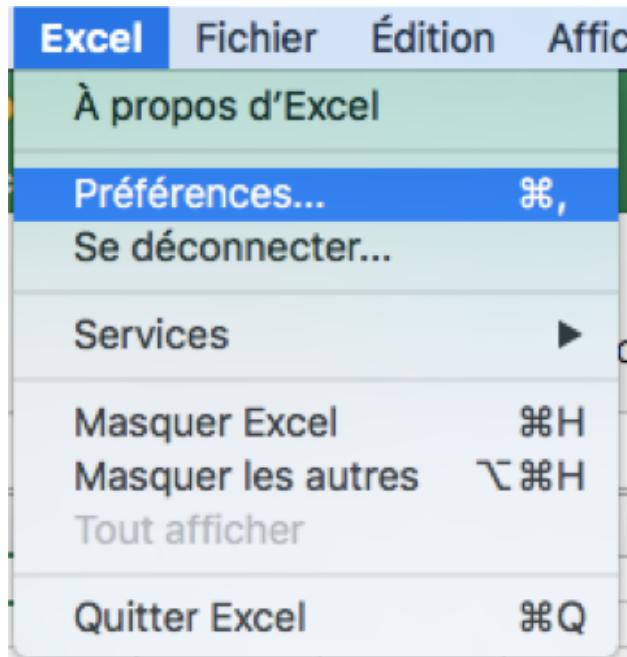
Objectifs du langage VBA

- Automatiser des actions répétitives
- Interagir sur les objets d'Excel (contenu et présentation)
- Créer des formulaires personnalisés

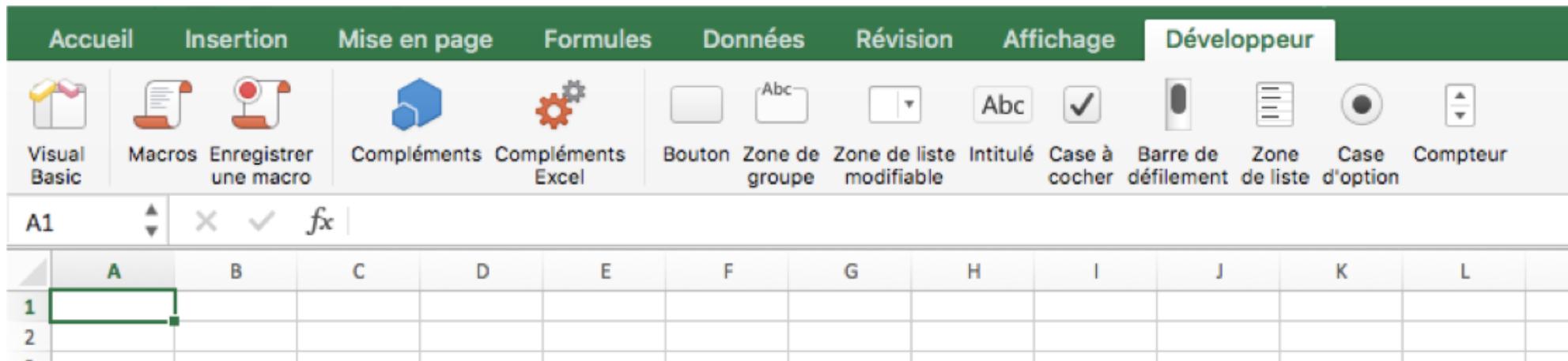
Ecriture de code VBA

- 2 possibilités pour créer des procédures VBA
 - Générer automatiquement le code à partir de l'**enregistreur de macros**
 - Saisir directement le code de votre procédure dans l'**environnement Visual Basic Editor (VBE)**

Configuration (1)



Configuration (2)



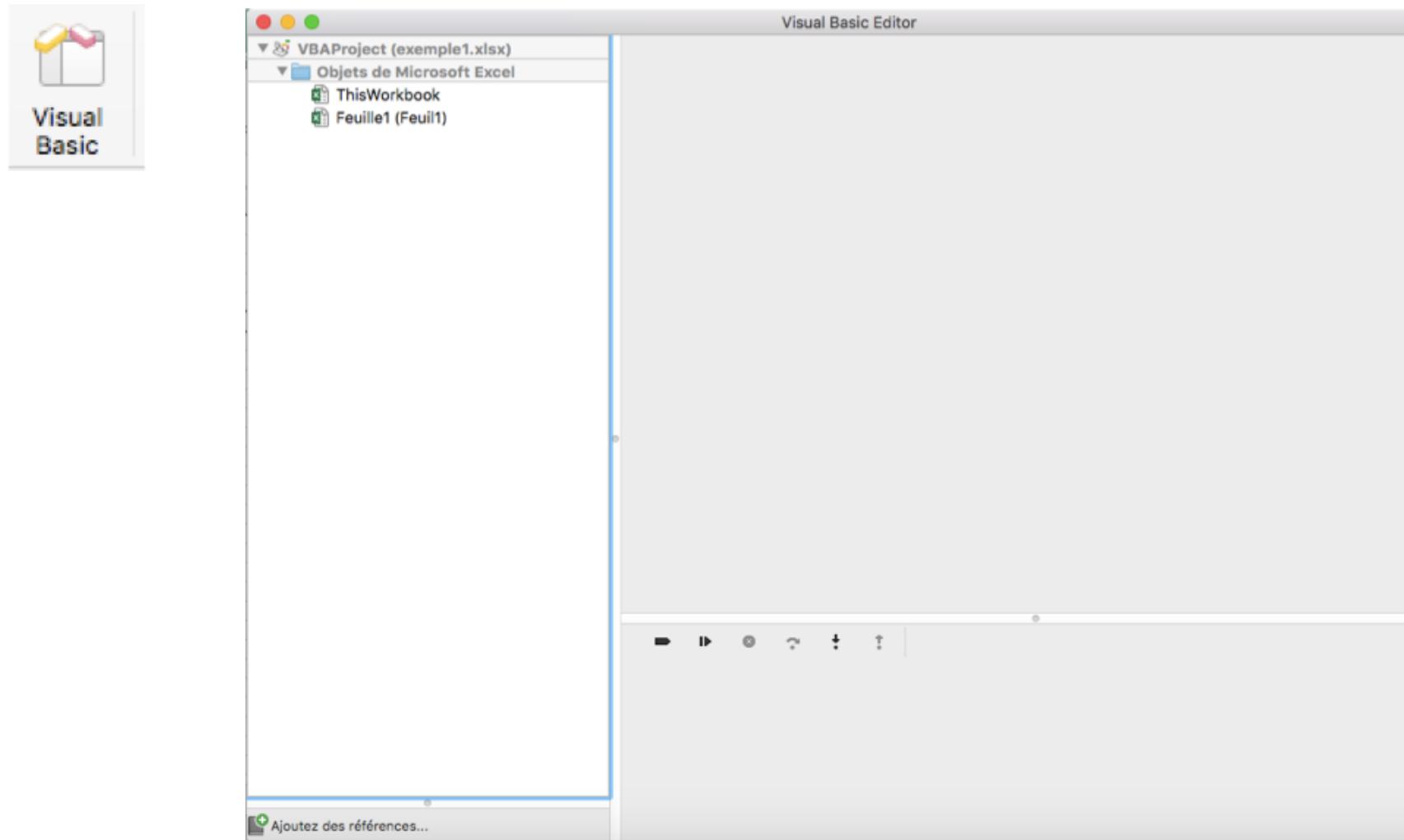
Accueil Insertion Mise en page Formules Données Révision Affichage Développeur

Visual Basic Macros Enregistrer une macro Compléments Excel Bouton Zone de groupe Zone de liste modifiable Intitulé Case à cocher Barre de défilement Zone de liste Case d'option Compteur

A1

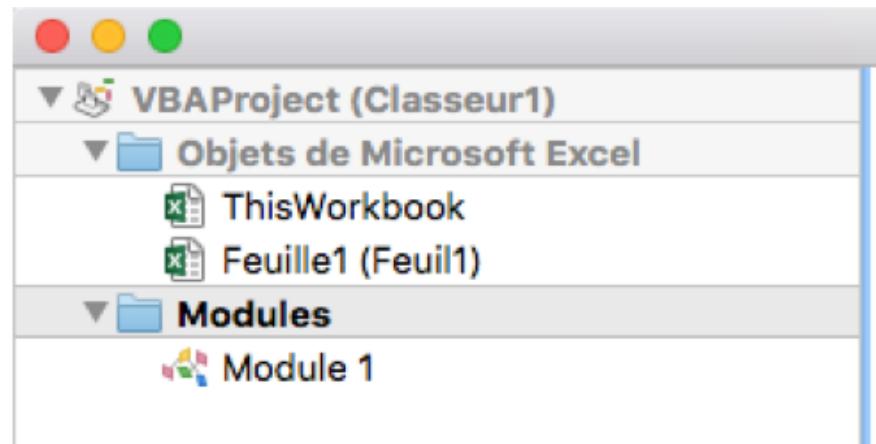
	A	B	C	D	E	F	G	H	I	J	K	L
1												
2												
-												

Configuration (3)



Quelques définitions

- **Procédures** : ce sont les **sous-programmes écrits en VBA**. Chaque macro génère une procédure. Il est également possible de créer ses propres procédures en utilisant l'instruction **Sub**.
- **Fonctions** : ce sont des **procédures retournant une valeur**. Elles commencent par l'instruction **Function**.
- **Module** : les modules contiennent les **macros enregistrées** et, les **procédures et fonctions écrites en VBA**. Ils peuvent être exportés sous forme de fichiers indépendants pour être utilisés dans d'autres classeurs.
- **Projet** : à chaque classeur ouvert dans Excel est associé un projet contenant tous les modules de code VBA regroupés par catégorie.



Exemples (procédure, fonction)

- Procédure

```
Sub Change_couleur()  
    Cells(1,1).Interior.Color=255
```

```
End Sub
```

- Fonction

```
Function Nom_feuille()  
    Nom_feuille=ActiveSheet.Name
```

```
End Function
```

Le résultat du calcul est stocké dans
le nom de la fonction

Qu'est-ce qu'un objet ?

- En VBA, un objet correspond à l'un des éléments de l'application Excel
 - Un classeur (**Workbook**)
 - Une feuille de calcul (**Worksheet**)
 - Une colonne (**Column**)
 - Une ligne (**Row**)
 - Une plage de cellules (**Range**)
 - Une cellule (**Cells**)
 - ...
- A ces objets vont être associées des **propriétés** et des **méthodes**

Propriété d'un objet

- Une propriété est une **caractéristique d'un objet**.
- Exemples :
 - Cellule : valeur, couleur, taille, police de caractères, *etc.*
 - Feuille : nom, visibilité (cachée ou non), sa couleur d'onglet, *etc.*
- Les propriétés sont différentes d'un objet à l'autre

Méthode d'un objet

- Une méthode correspond à **une action effectuée sur un objet**
- Exemples :
 - Cellule : copiée, collée, effacée, remplie, *etc.*
 - Feuille : copiée, supprimée, renommée, *etc.*

Règle de codification

- Un objet est associé à une propriété ou à une méthode en les séparant par un point

Objet.Méthode

Ou

Objet.Propriété

- Exemple :

Sheets("Feuille_Laure").Delete

Associations d'objets, de propriétés ou de méthodes

- Il est possible de faire des associations d'objets entre eux ou d'associer à un objet plusieurs propriétés ou méthodes
- Exemples

```
Sheets("Feuille_Laure").Cells(7,3).Copy
```

```
Workbooks("Class_Laure").
```

```
Sheets("Feuil1").Cells(7,3)=Workbooks("Class_Laure2").Sheets("Feuil2").Cells(4,2)  
Cells(1,1).CurrentRegion.Rows.Count
```

Codification des cellules et des plages de cellules

- Une cellule se code de la manière suivante :
Cells(numero de ligne, numero de colonne)
- Une plage de cellule :
Range(référence)
- Exemples :
Cells(1,1)
Range("A1:C6").Select
Range("A1").Select
[A1:C6].Select
Range(Cells(1,1):Cells(3,6)).Select

Affectation

- Le signe **=** permet l'affectation
- Exemples :
 - Cells(1,1)=15
 - Cells(1,1).Interior.Color=255
 - Cells(1,4).Interior.Color=Cells(1,1).Interior.Color

CRÉER, EXÉCUTER ET ENREGISTRER UNE MACRO

Intérêt d'une macro (commande)

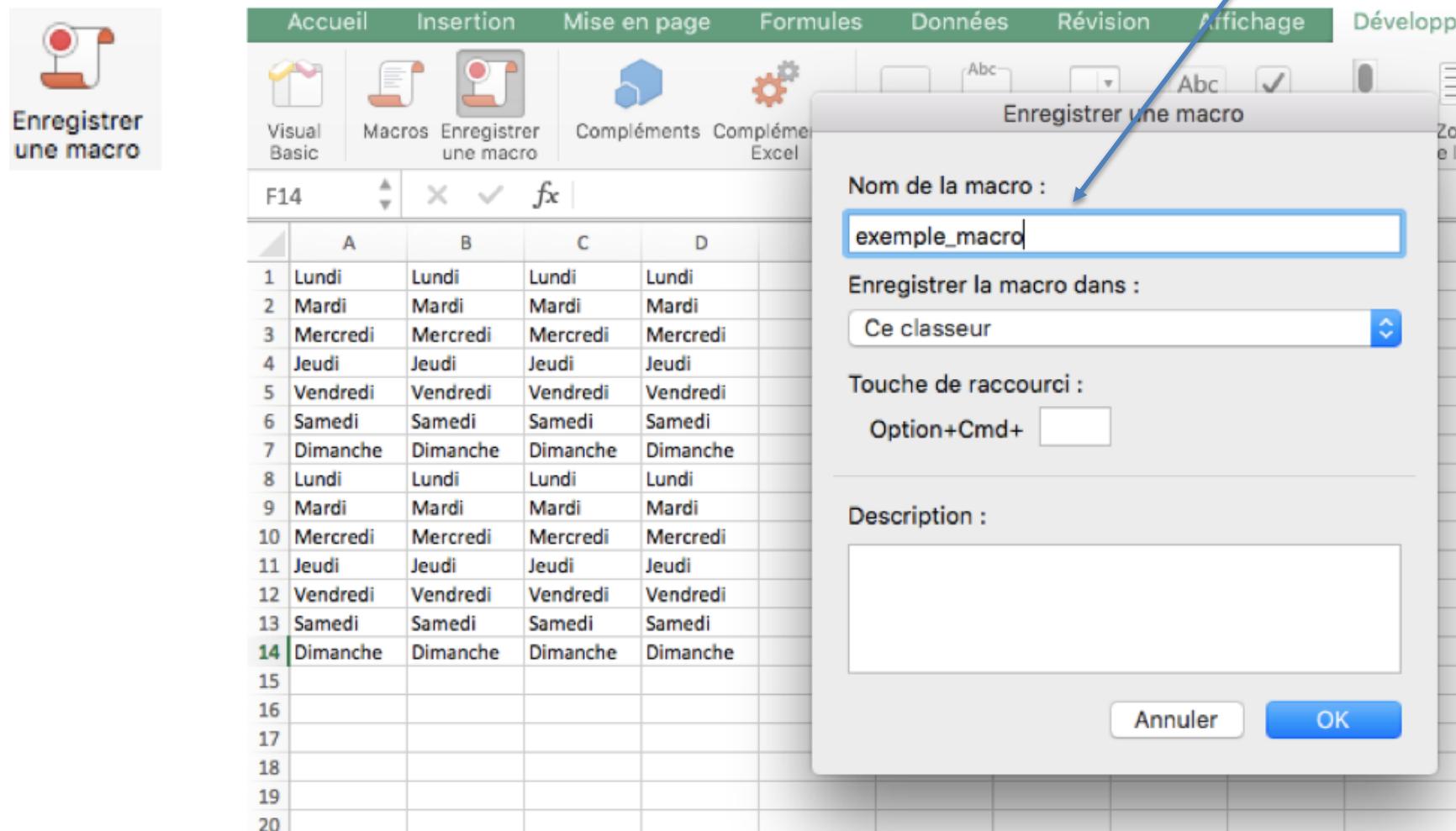
- Excel va enregistrer toutes les manipulations dans une macro grâce à l'**enregistreur de macros**.
- Il sera alors possible d'exécuter la suite de manipulations en un seul clic ou grâce à une combinaison de touche.

Exemple

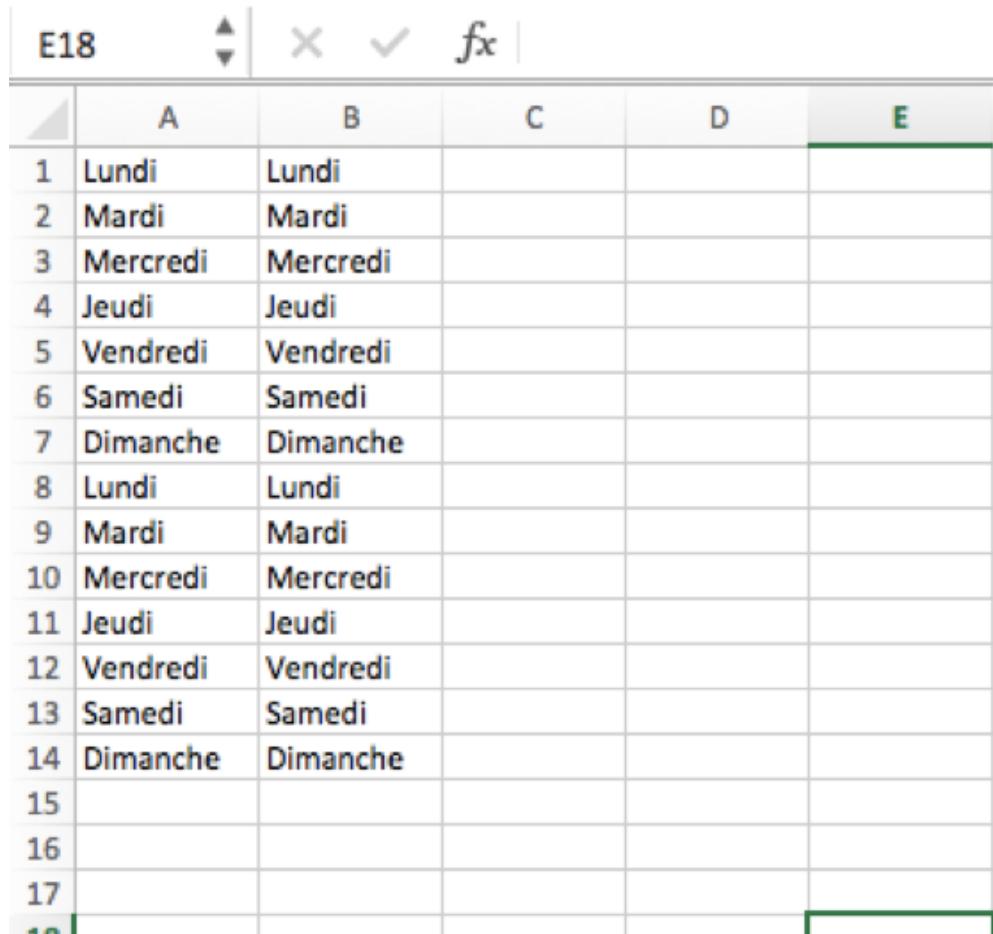
- Supprimer le contenu des colonnes B et D
- Déplacer le contenu de la colonne C dans la colonne B

Lancement de l'enregistreur de macros

Nommer la macro



Faire les manipulations



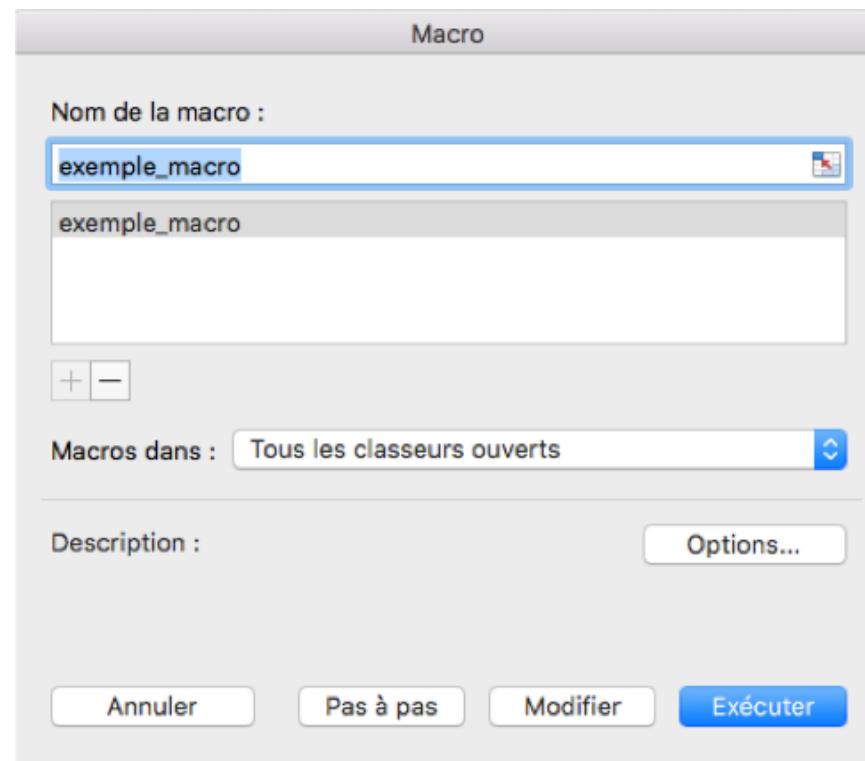
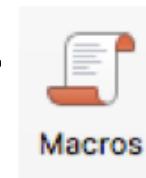
	A	B	C	D	E
1	Lundi	Lundi			
2	Mardi	Mardi			
3	Mercredi	Mercredi			
4	Jeudi	Jeudi			
5	Vendredi	Vendredi			
6	Samedi	Samedi			
7	Dimanche	Dimanche			
8	Lundi	Lundi			
9	Mardi	Mardi			
10	Mercredi	Mercredi			
11	Jeudi	Jeudi			
12	Vendredi	Vendredi			
13	Samedi	Samedi			
14	Dimanche	Dimanche			
15					
16					
17					

Elles
s'enregistrent...
jusqu'à ce que
l'utilisateur clique
sur



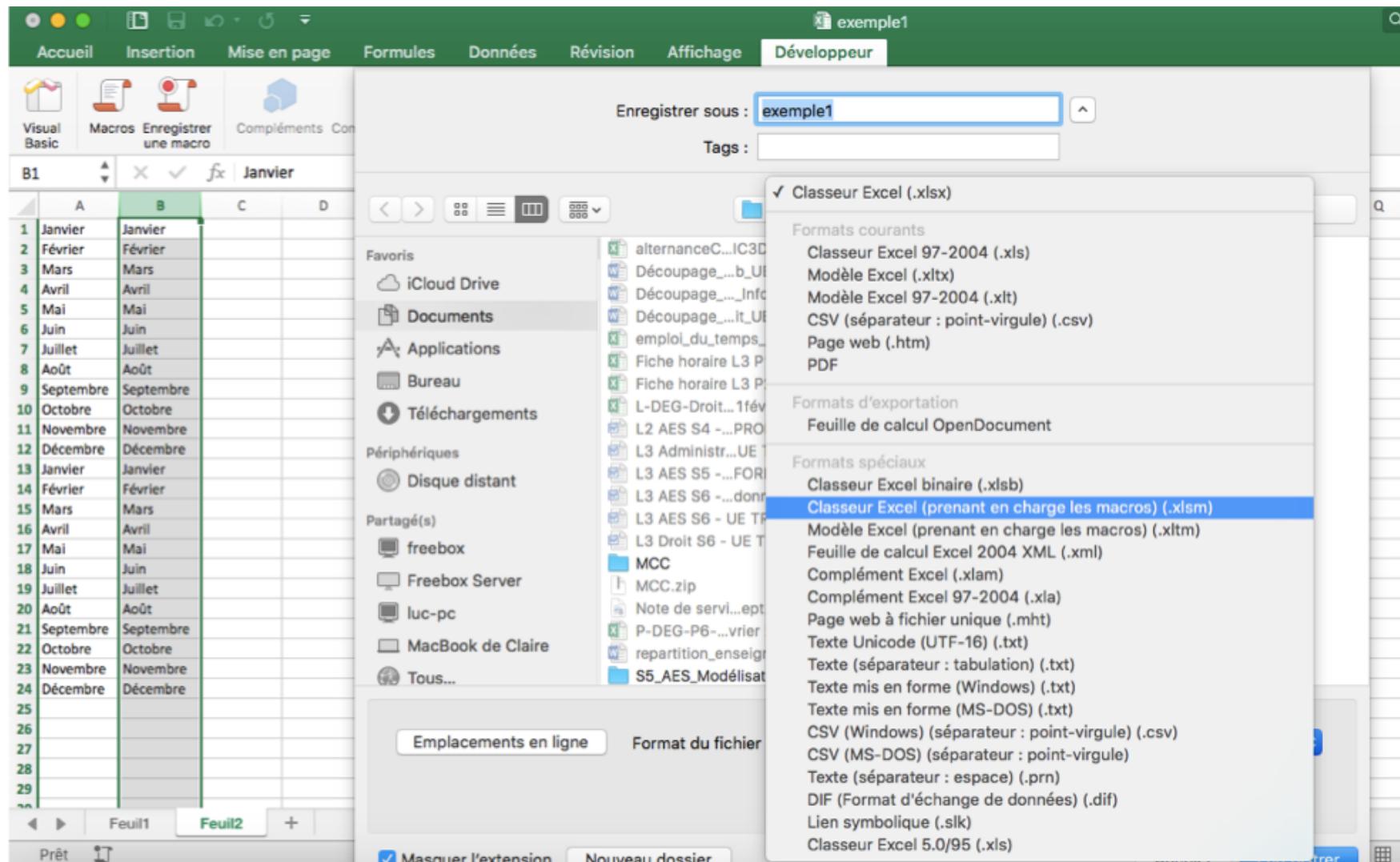
Exécuter une macro

- Une première solution est de cliquer sur
- Puis de choisir la macro à exécuter
- Et d'appuyer sur *Exécuter*



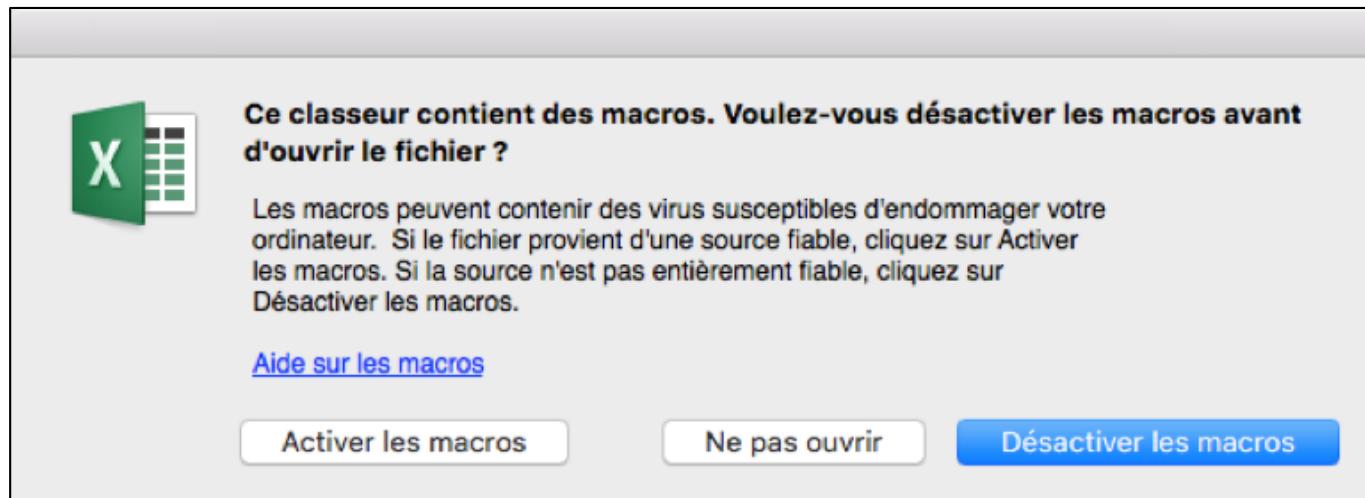
Un autre exemple en live...

Enregistrer un classeur avec une macro



Remarques

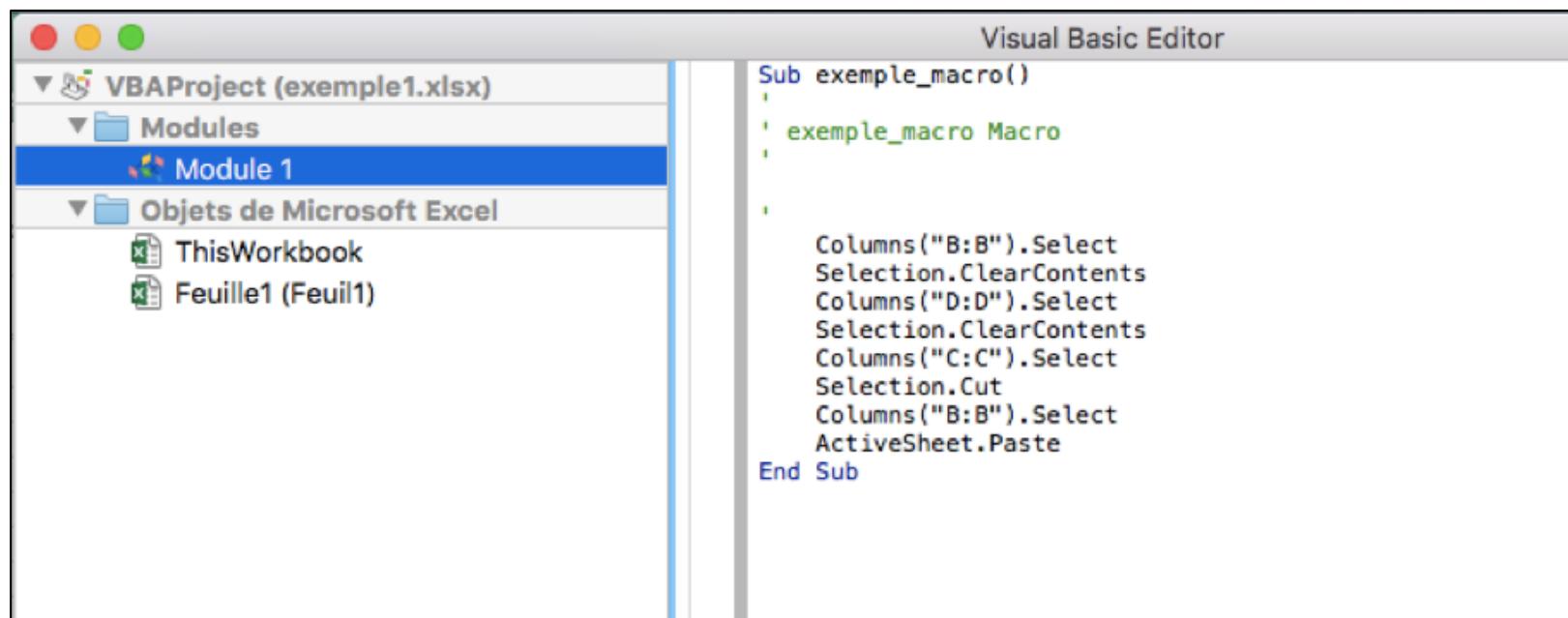
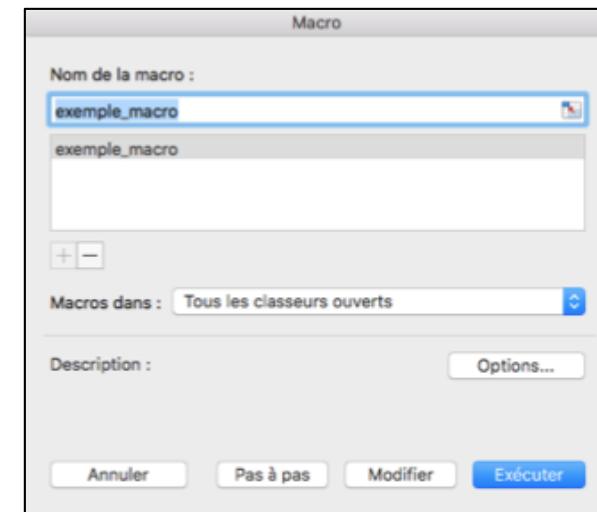
- Les classeurs contenant des macros ont pour extension *xlsm* (au lieu de *xlsx*).
- Lorsque vous ouvrez un classeur contenant une ou des macros, Excel vous demande si vous voulez activer les macros. Il existe en effet des macrovirus...



MODIFIER LE CODE D'UNE MACRO

Accéder au code d'une macro

- Cliquer sur Macro/Modifier



Plus précisément...

```
Sub exemple_macro()
'
' exemple_macro Macro
'
'
Columns("B:B").Select
Selection.ClearContents
Columns("D:D").Select
Selection.ClearContents
Columns("C:C").Select
Selection.Cut
Columns("B:B").Select
ActiveSheet.Paste
End Sub
```

Ajoutons un message

```
Sub exemple_macro()
'
' exemple_macro Macro
'

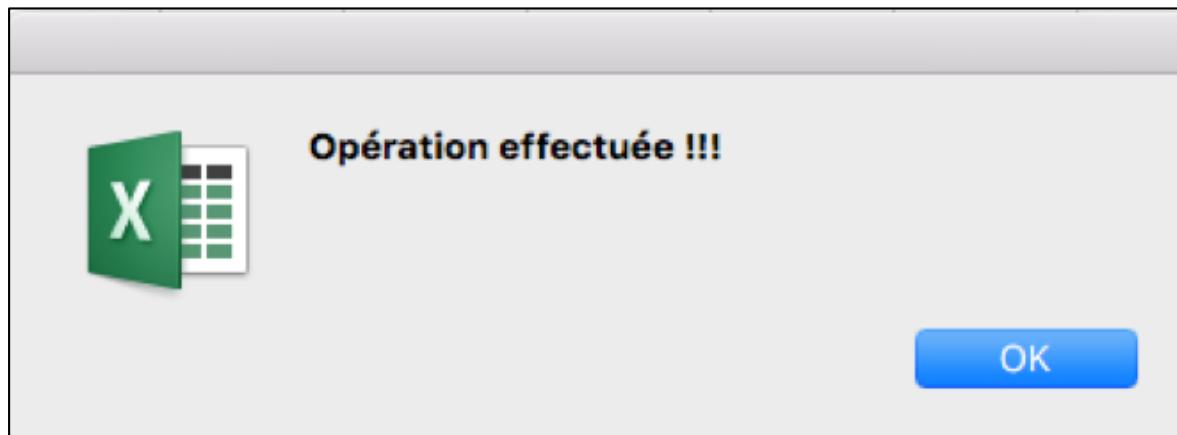
'
Columns("B:B").Select
Selection.ClearContents
Columns("D:D").Select
Selection.ClearContents
Columns("C:C").Select
Selection.Cut
Columns("B:B").Select
ActiveSheet.Paste

MsgBox ("Opération effectuée !!!")

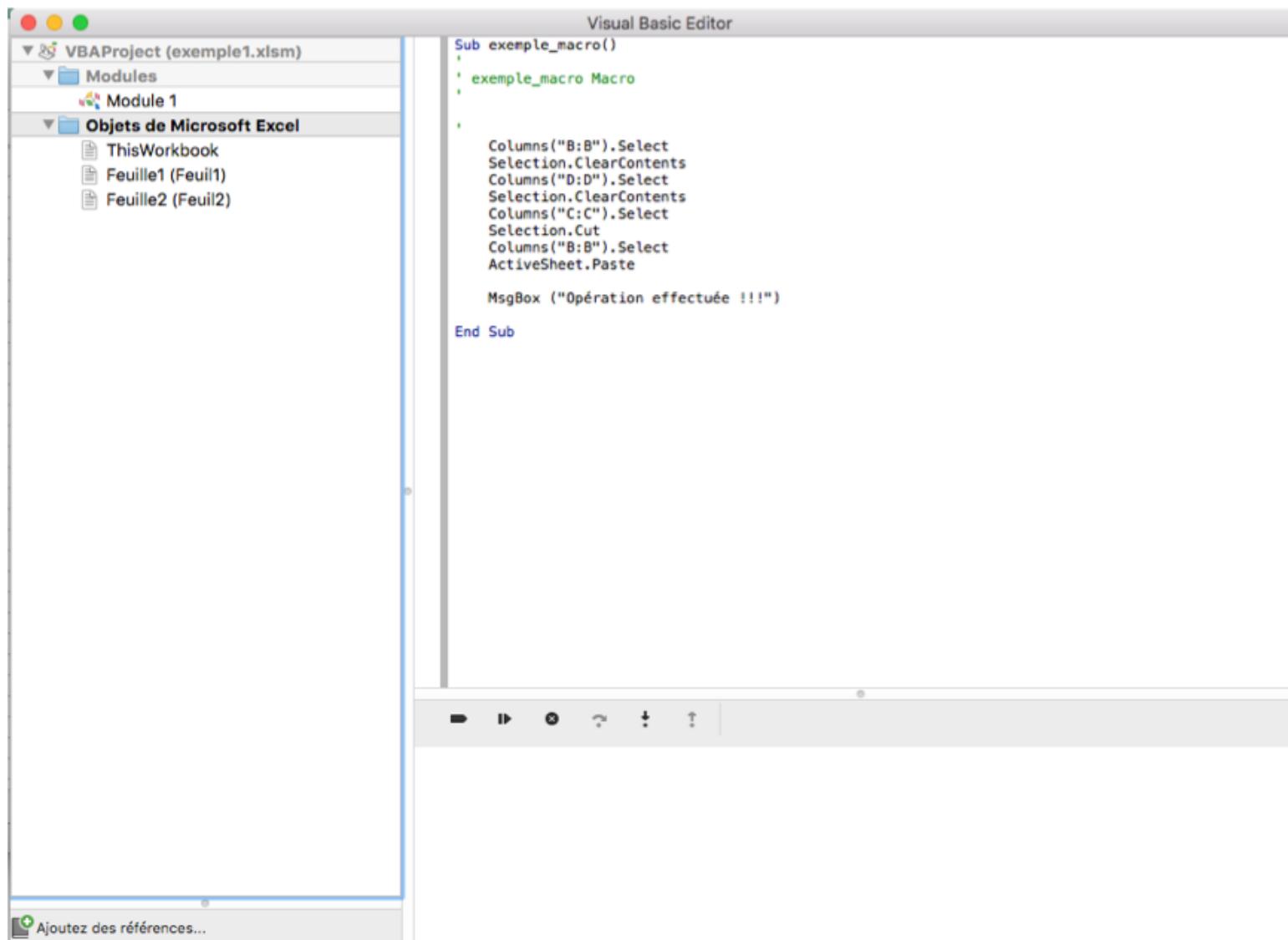
End Sub
```

Résultat

- En plus de la modification précédente, on obtient la boîte de message suivante :



Remarque : l'environnement VBE



VBE

- C'est l'environnement dans lequel vous pouvez saisir, modifier et tester votre code VBA.
- Cet environnement est également appelé IDE (Integrated Development Environment) ou éditeur VBA.
- Il intègre de nombreux outils pour faciliter la programmation et la mise au point du code VBA : outils de déboggage, assistance à la saisie, explorateur d'objets, etc.

→ On y reviendra à la prochaine séance

AFFECTER UNE MACRO

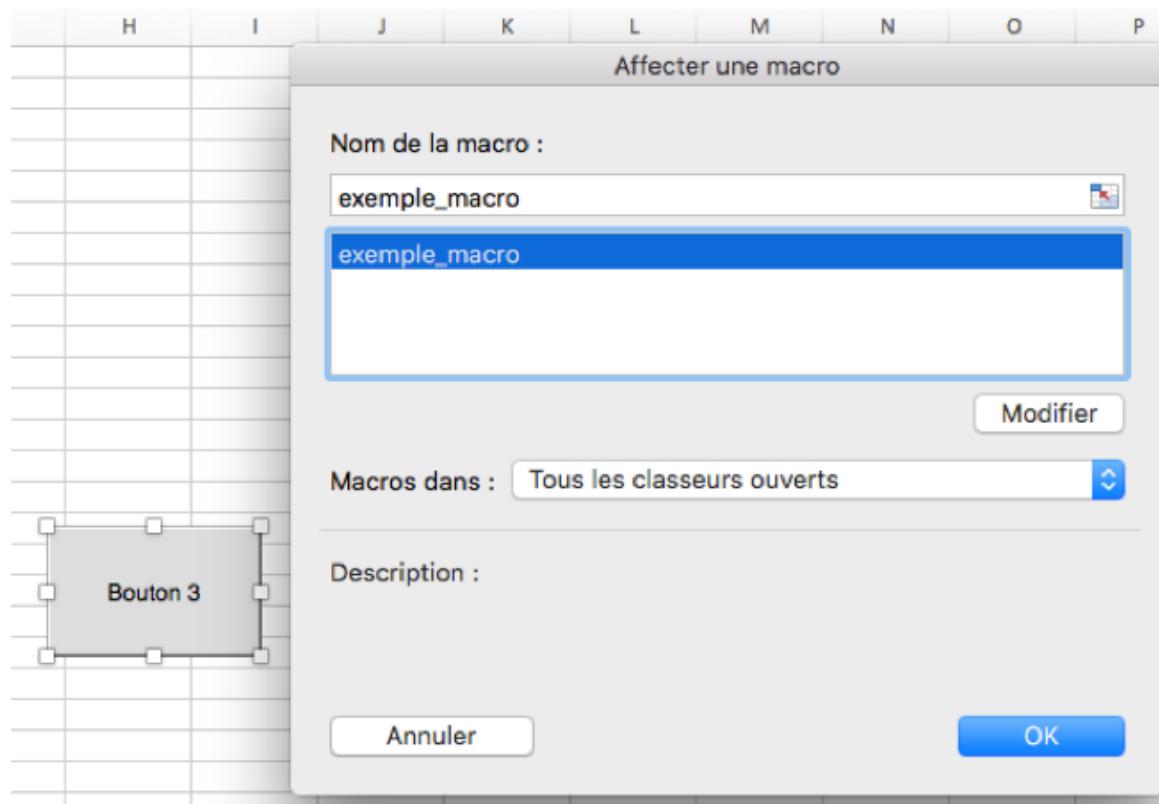
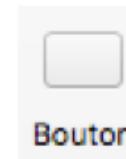
Affecter une macro

- Il est possible d'affecter une macro aux différents éléments suivants :
 - Commandes personnalisées du ruban
 - Bouton de commande de la barre d'outils Accès rapide
 - Contrôles ou objets insérés dans des feuilles de calcul Excel (bouton de commande, image, *etc.*)

Dans le cadre de ce cours

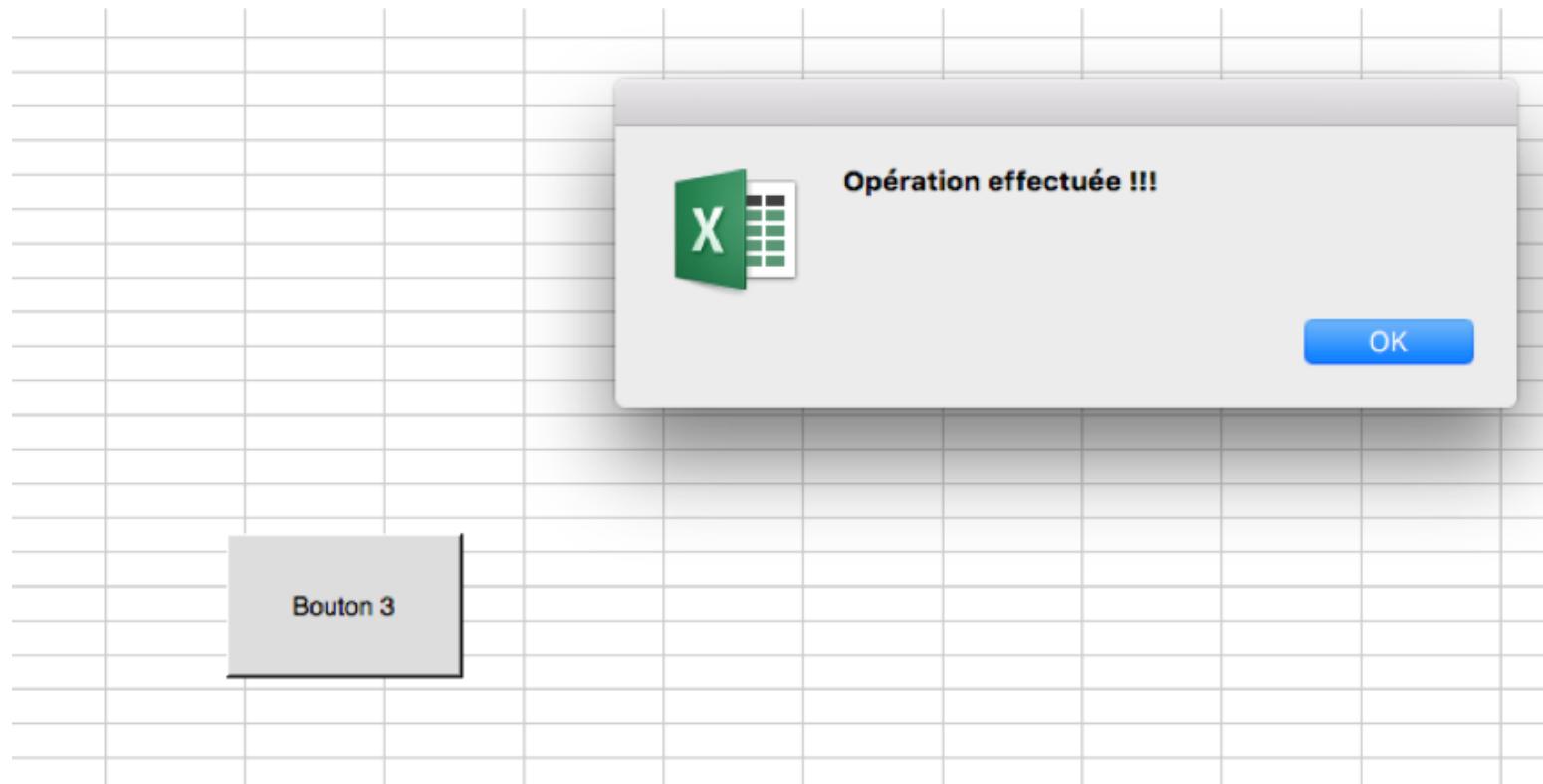
Affecter une macro à un bouton de commande

- Insérer un bouton sur la feuille
- Sélectionner la macro



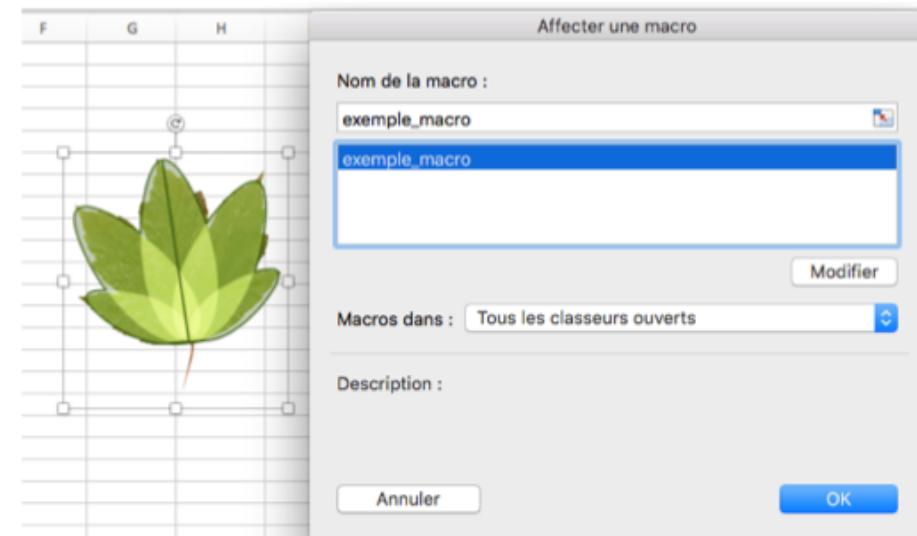
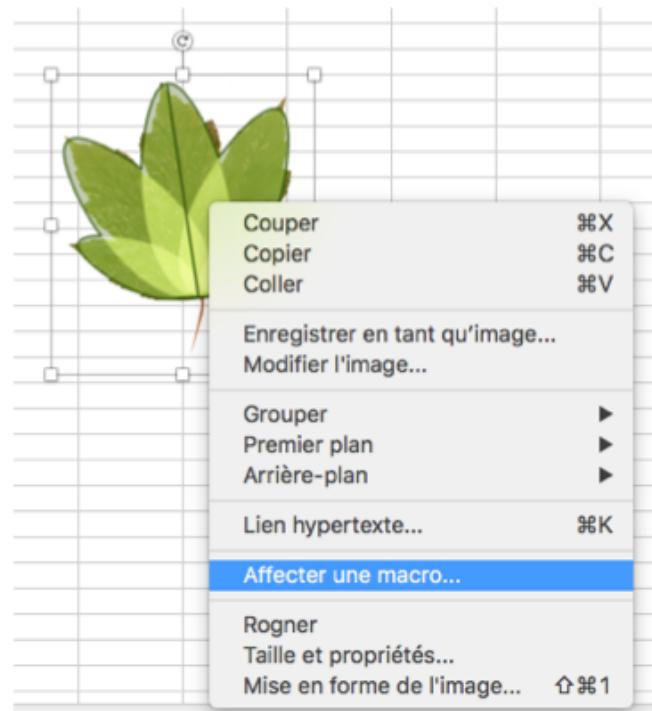
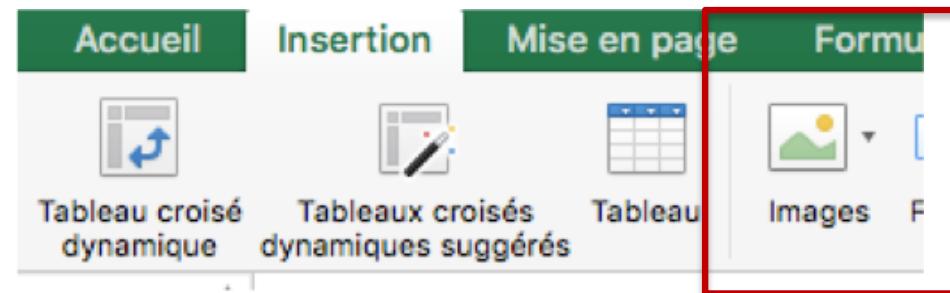
Exécuter la macro à partir du bouton

- Il suffit de cliquer sur le bouton !



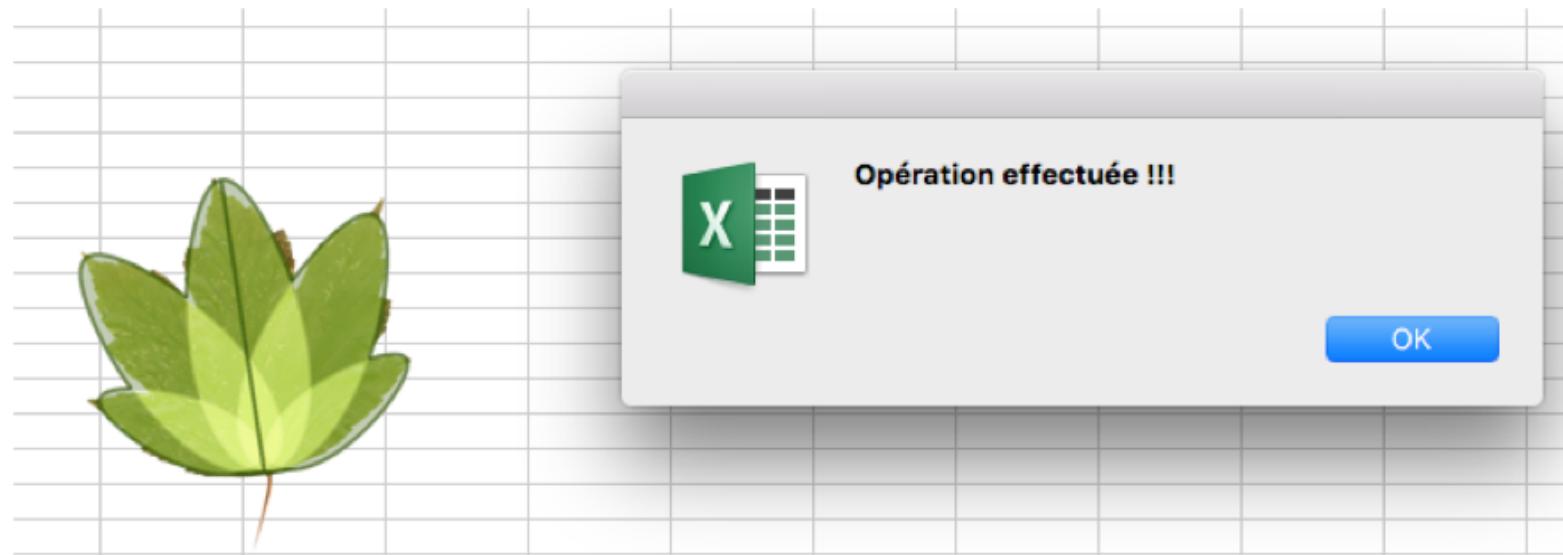
Associer une macro à une image

- Insérer une image



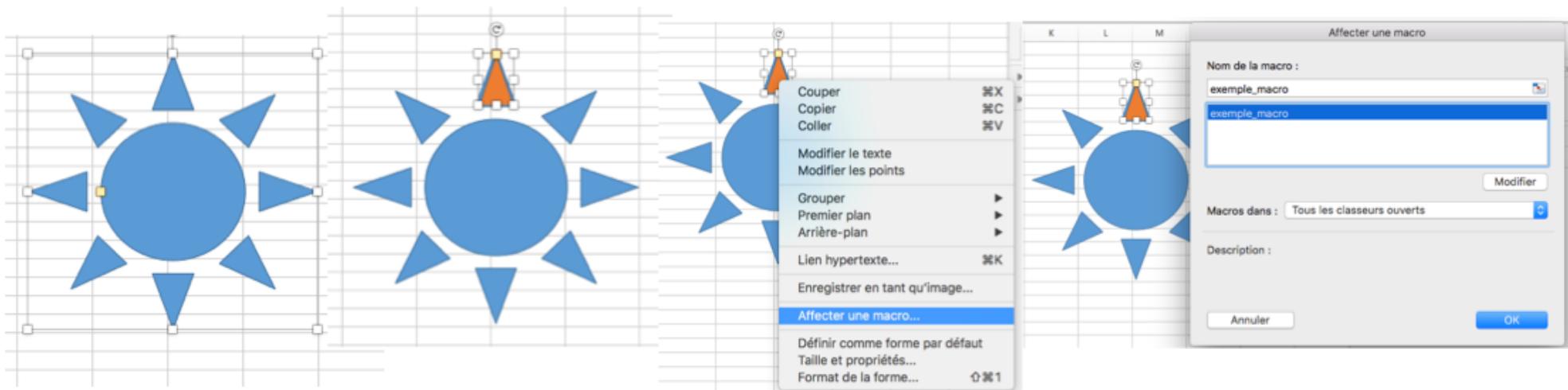
Exécuter la macro à partir de l'image

- L'image est cliquable et un clic sur l'image déclenche l'exécution de la macro



Associer une macro à une zone d'un objet graphique

- Insérer un objet graphique (image, image clipart, forme, SmartArt)
- Créer une zone réactive dans l'objet existant à l'aide des formes
- Associer cette zone à une macro



Exécuter une macro à partir d'une zone d'un objet graphique

- Vous pouvez masquer la zone : sans remplissage et sans contour !

