

Développement VBA & Excel

2023-2024

Eductive

Campus Eductive – Aix en Provence

ESGI 3

David Palermo

Mail : dpalermo1@myges.fr

Développement VBA & Excel

Semestre 1 : 15 heures

Excel & VBA introduction

- 1 Excel
- 2 VBA
- 3 VBA & Excel Sécurisation

1 - Excel : Sommaire

- 1.1 La notion de tableur
- 1.2 Lancer Excel
- 1.3 L'interface Microsoft Excel 365 MSO
- 1.4 Principe des formules
- 1.5 Création de formules simples
- 1.6 Recopie des formules
- 1.7 Adresse relative
- 1.8 Adresse absolue
- 1.9 Utilisation des fonctions dans les formules
- 1.10 les 10 fonctions consulter le plus souvent
- 1.11 Quelques fonctions
- 1.12 Réalisation d'un graphique
- 1,13 Excel : Introduction - Tableau Croisé
- 1.14 Excel : Introduction – Base de donnée
- 1.15 Lien Utiles

1.1 - Excel : La notion de tableur

Un Tableur :

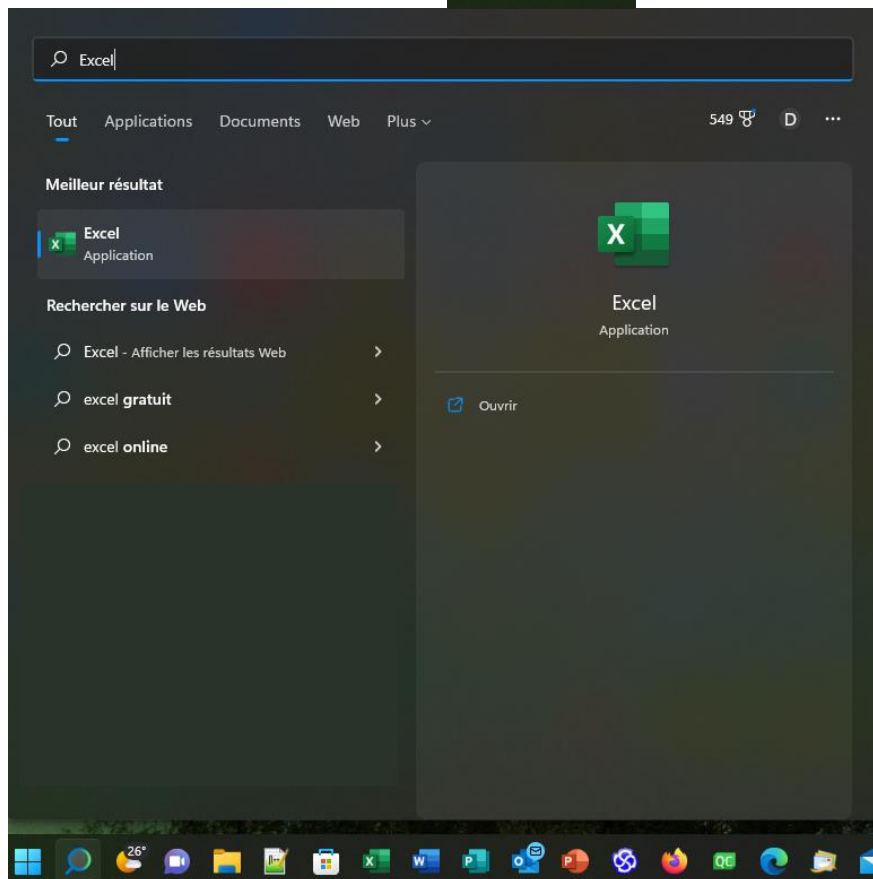
- Permet de manipuler des données sous forme de tableau (lignes, colonnes)
- Contient une ou plusieurs feuilles de calcul : classeur
- Une feuille de calcul est formé de cellules : intersection d'une ligne et d'une colonne
- On désigne une cellule par son adresse : une lettre (pour la colonne) et un chiffre (pour la ligne)

	A	B	C	D
1	A1	B1	C1	D1
2	A2	B2	C2	D2
3	A3	B3	C3	D3
4				

1.2 - Excel : Lancer Excel

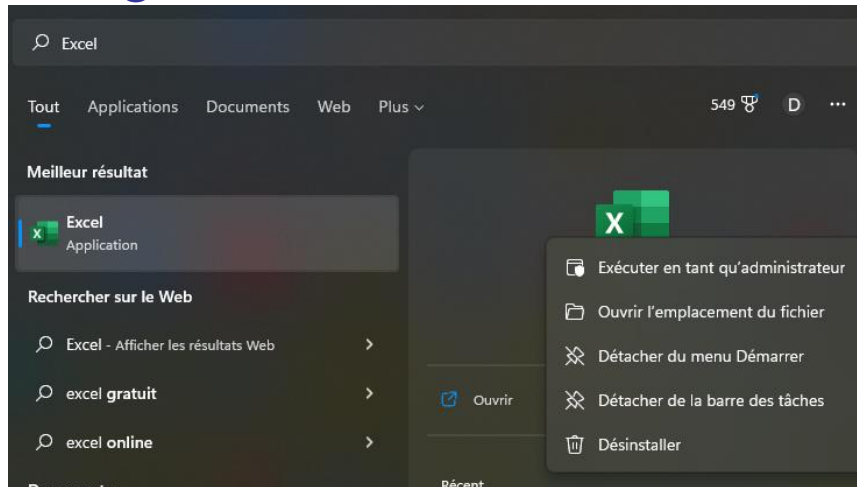
1. Par le menu

Bouton **démarrer** puis ->



1.2 - Excel : Lancer Excel

2. Par l'icône de Excel : mettre un raccourci menu de démarrage et/ou barre de tache



3. Exécuter Excel on peut aussi exécuter Excel en cliquant sur un formats de fichier prise en charge

<https://docs.microsoft.com/fr-fr/deployoffice/compat/office-file-format-reference>

1.3 - Excel : L'interface Microsoft Excel 365 MSO

Excel

Accueil

Nouveau

Ouvrir

Compte

Votre avis

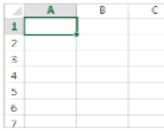
Options

Excel


david palermo DP

Bonjour


Nouveau




Nouveau classeur




Jeu Sudoku



Budget personnel mensuel



Plan d'amortissement



Planning hebdomadaire des...

Autres modèles →

Rechercher

Récent

Épinglé

Partagés avec moi

Épinglez les fichiers de votre choix pour les retrouver facilement plus tard. Cliquez sur l'icône d'épingle qui s'affiche lorsque vous placez le pointeur sur un fichier.

Autres classeurs →

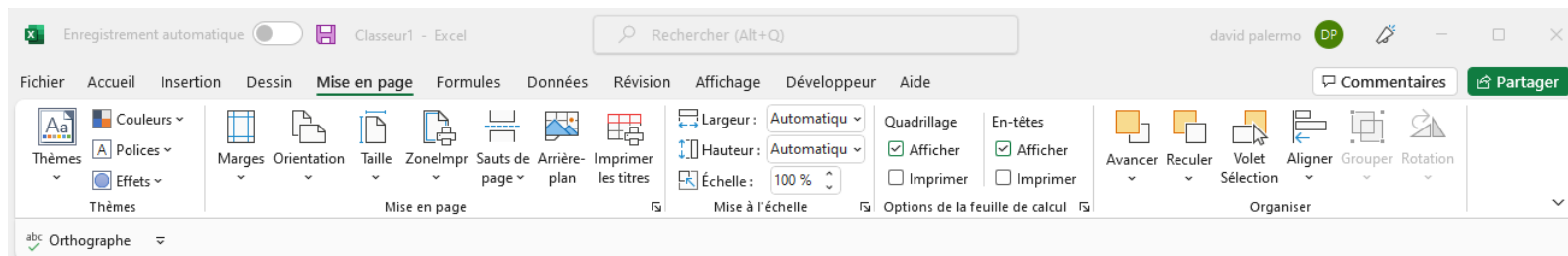
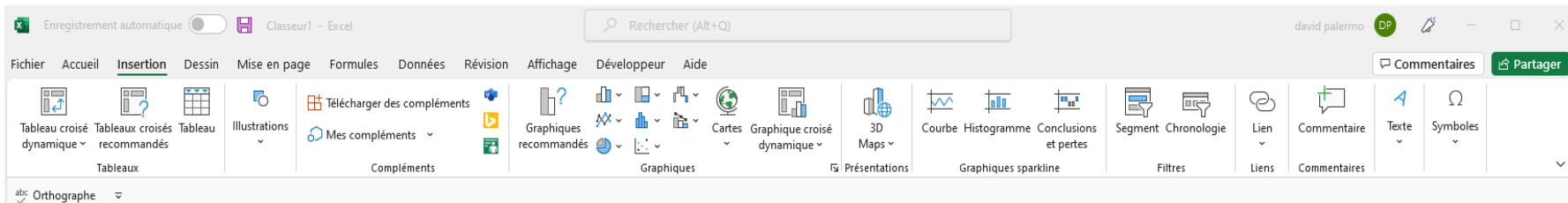
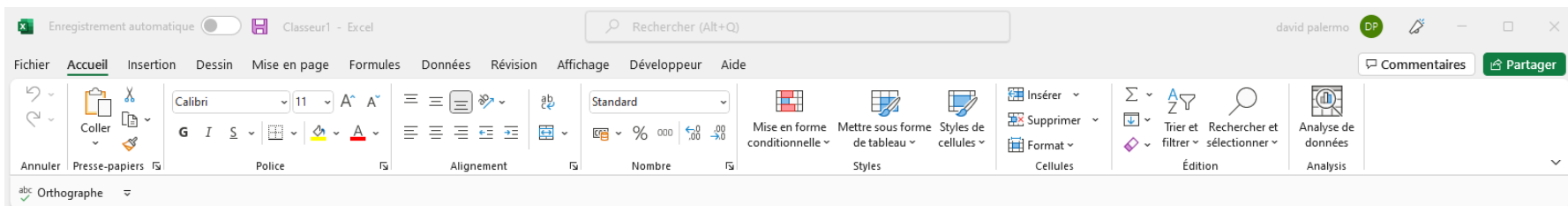


Menu

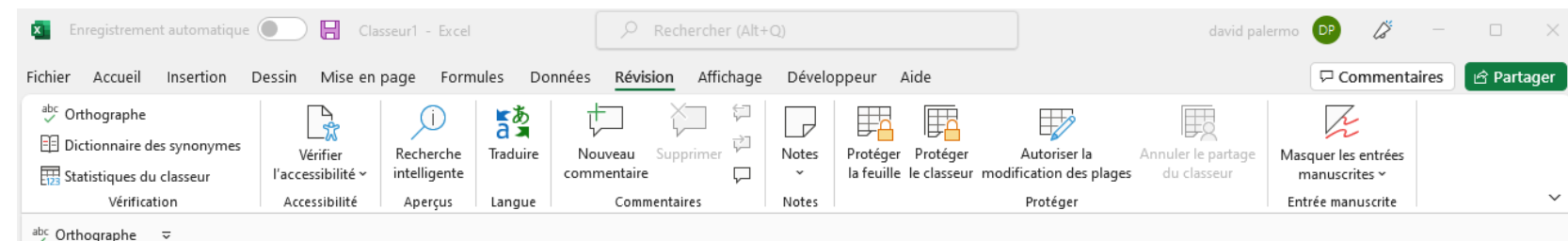
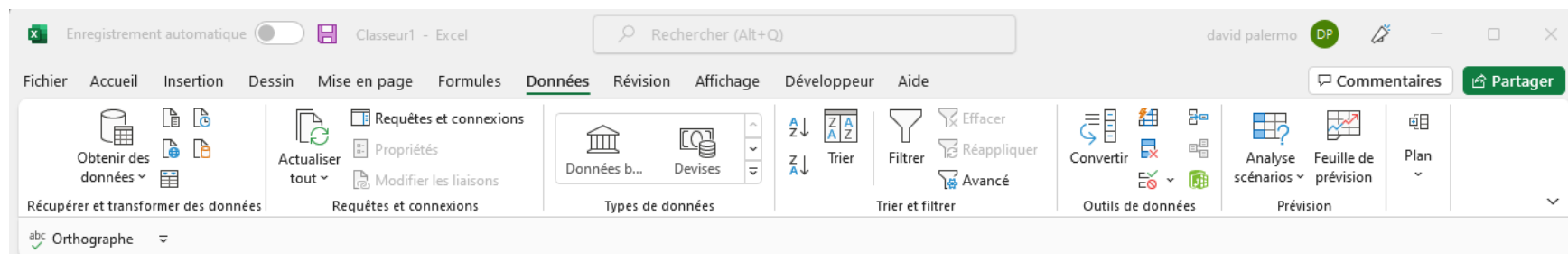
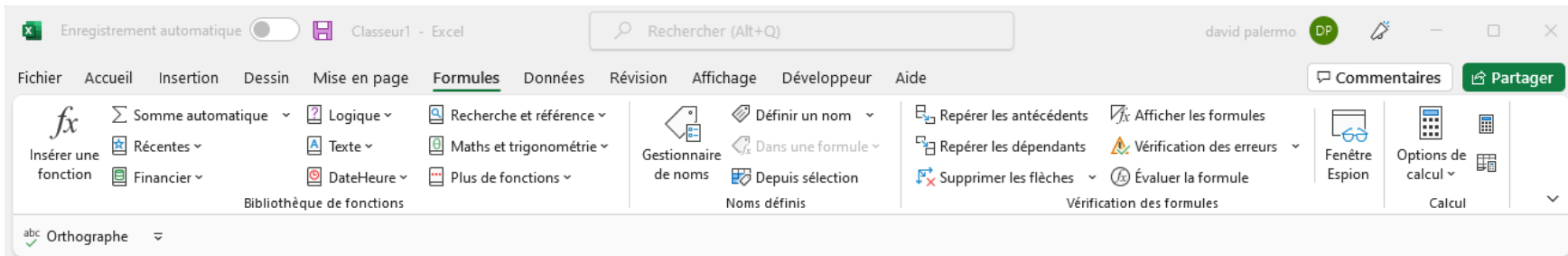


Feuilles de calcul

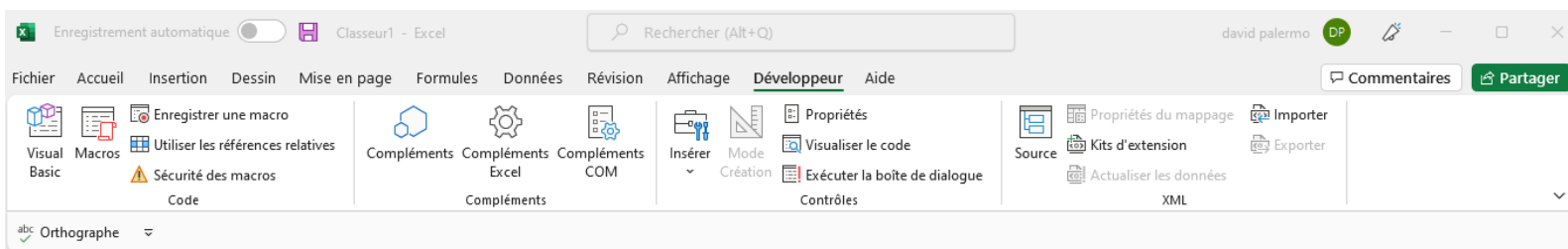
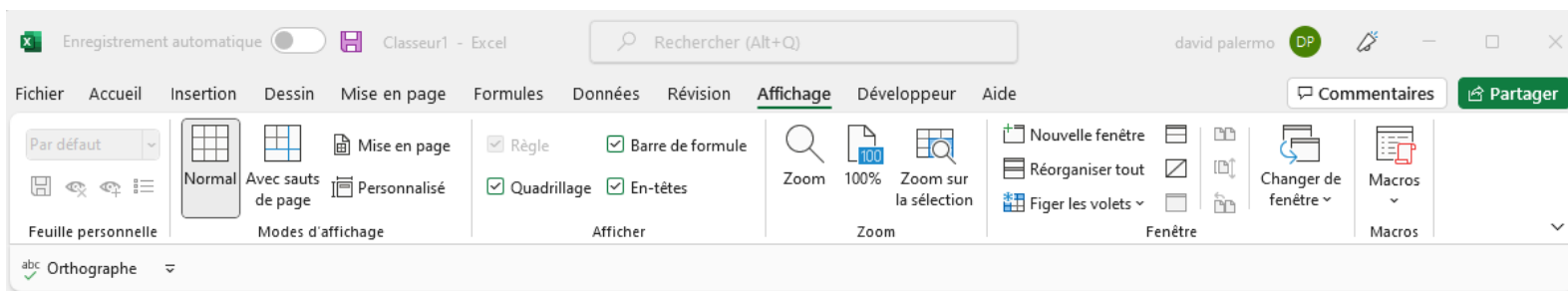
1.3 - Excel : L'interface Microsoft Excel 365 MSO



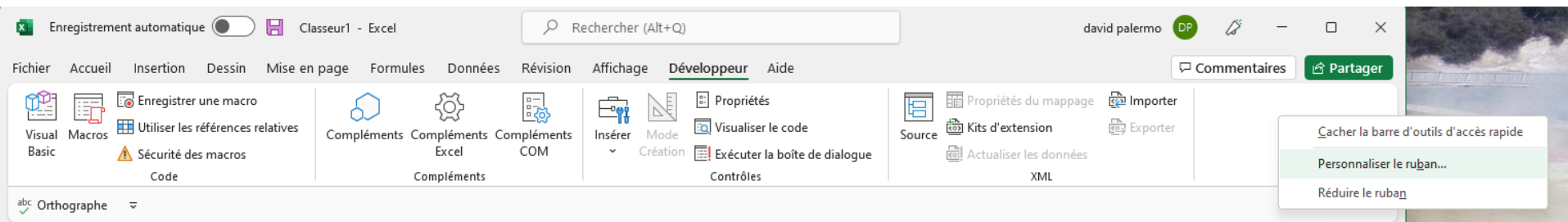
1.3 - Excel : L'interface Microsoft Excel 365 MSO



1.3 - Excel : L'interface Microsoft Excel 365 MSO



1.3 - Excel : L'interface Microsoft Excel 365 MSO



1.4 - Excel : Principe des formules

Formule de calcul : texte qui réalise un calcul et affiche le résultat dans une cellule, qui commence par **=**

On peut trouver après :

- Une adresse de cellule : B4
- Un opérateur arithmétique : +
- Une constante : 50
- Une fonction : SOMME()

Exemples :

$= B4 + C4$ $= E5 * 20$ $= SOMME(B2:D2)$

1.5 - Excel : Création de formules simples

Exercice: Calcul Facturation Total HT et Total TTC

		Facture			
<i>Désignation</i>	<i>Prix HT</i>	<i>Quantité</i>	<i>Total HT</i>	<i>TVA</i>	<i>Total TTC</i>
Clé USB	5,00 €	8		19,6	
bifteck	8,50 €	3		5,6	

		Facture			
<i>Désignation</i>	<i>Prix HT</i>	<i>Quantité</i>	<i>Total HT</i>	<i>TVA</i>	<i>Total TTC</i>
Clé USB	5,00 €	8	40,00 €	19,6	47,84 €
bifteck	8,50 €	3	25,50 €	5,5	26,90 €

1.6 - Excel : Recopie des formules

	A	B	C	D	E	F
1			Facture			
2						
3	Désignation	Prix HT	Quantité	Total HT	TVA	Total TTC
4	Clé USB	5,00 €	8	40,00 €	19,6	47,84 €
5	bifteck	8,50 €	3	25,50 €	5,5	26,90 €
6				0,00 €		
7				0,00 €	+	
8						

= B5 * C5

= B7 * C7

= (100+E5)/100*D5

Clic sur E5,
Coin inférieur droit -> +

1.7 - Excel : Adresse relative

Dans l'exemple précédent

ligne 5 $= B5 * C5$

par recopie on a

ligne 6 $= B6 * C6$

ligne 7 $= B7 * C7$

	A	B	C	D	E	F
1			Facture			
2						
3	Désignati	Prix HT	Quantité	Total HT	TVA	Total TTC
4	Clé USB	5,00 €	8	40,00 €	19,6	47,84 €
5	bifteck	8,50 €	3	25,50 €	5,5	26,90 €
6				0,00 €		
7				0,00 €		
8						
9						

Excel multiplie les 2 cellules situées à gauche.

L'adresse relative à une feuille : maFeuille
maFeuille!B5

1.8 - Excel : Adresse absolue

	A	B	C	D	E
1	Conversion en francs				
2			Taux euro	6,55957	
3		Euros	Francs		
4		1,00 €	6,56		B4*D2
5		2,00 €	0		B5*D3
6		3,00 €	0		B6*D4

Adresse absolue
\$D\$2

	A	B	C	D	E
1	Conversion en francs				
2			Taux euro	6,55957	
3		Euros	Francs		
4		1,00 €	6,56		B4*D2
5		2,00 €	0		B5*D3
6		3,00 €	0		B6*D4
7					

1.9 - Excel : Utilisation des fonctions dans les formules

Plage de cellule

Ensemble de cellules formant un rectangle E19:H30
coin supérieur gauche : coin inférieur droit

Fonction : succession de deux éléments

- Le nom de la fonction
- Un argument encadré par des parenthèses

SOMME(B5:B13)

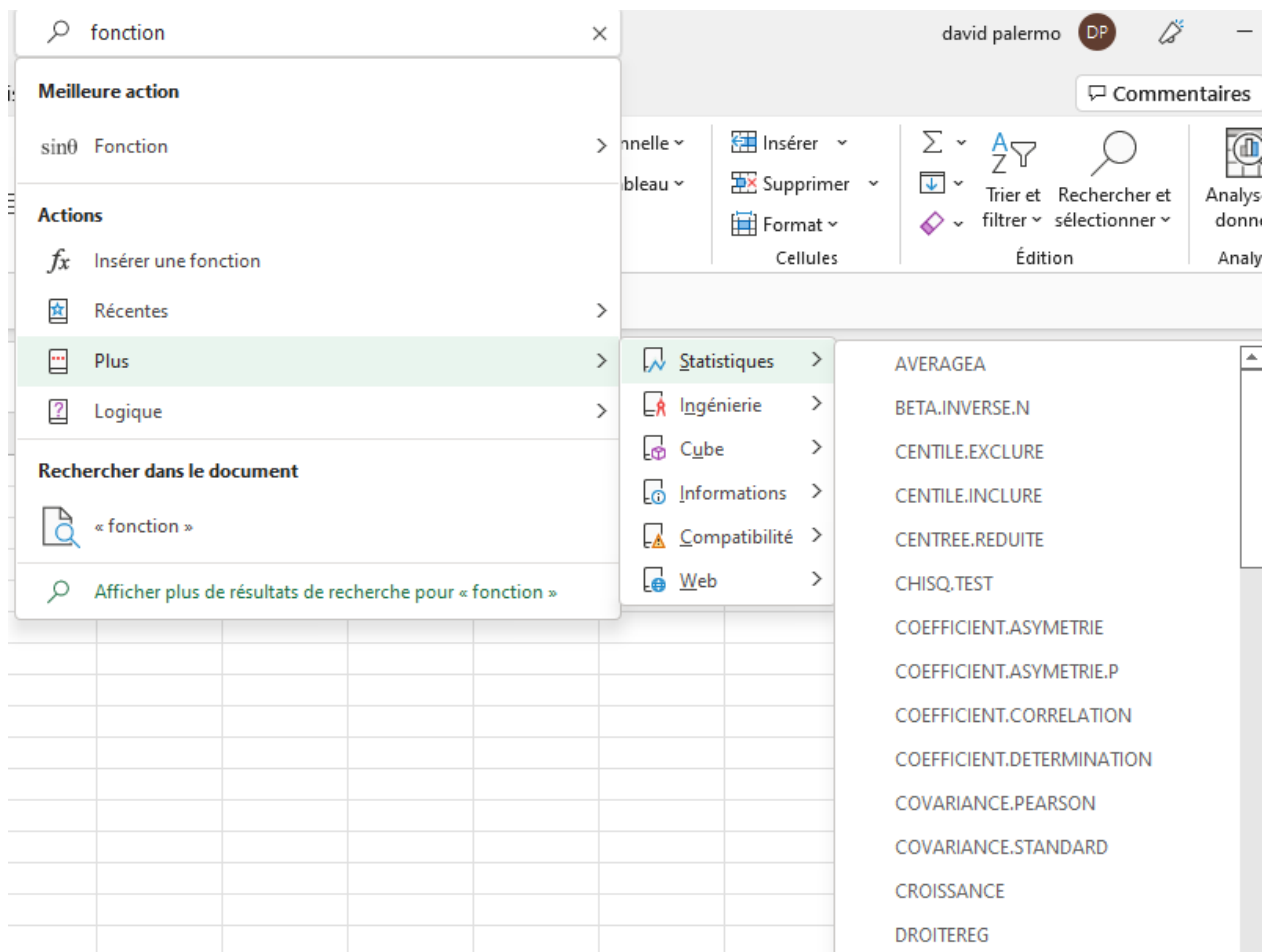
👉 *Pas de blanc entre le nom de la fonction et la parenthèse*

1.10 - Excel : les 10 fonctions consulter le plus souvent.

<https://support.microsoft.com/fr-fr/office/fonctions-excel-par-cat%C3%A9gorie-5f91f4e9-7b42-46d2-9bd1-63f26a86c0eb>

Fonction	Description
SOMME	Utilisez cette fonction pour additionner les valeurs dans une cellule.
SI	Utilisez cette fonction pour renvoyer une valeur si une condition est vraie et une autre valeur si elle est fausse.
RECHERCHE	Utilisez cette fonction pour effectuer une recherche dans une ligne ou une colonne et renvoyer une valeur à partir de la même position dans une seconde ligne ou colonne.
RECHERCHE V	Utilisez cette fonction pour rechercher des éléments dans une table ou une plage par ligne. Par exemple, vous pouvez rechercher le nom d'un employé à l'aide de son matricule ou rechercher son numéro de téléphone à l'aide de son nom (comme dans un annuaire téléphonique). Regardez cette vidéo sur l'utilisation de la fonction RECHERCHEV.
EQUIV	Utilisez cette fonction pour rechercher un élément dans une plage de cellules, puis renvoyer la position relative de cet élément dans la plage. Par exemple, si la plage A1:A3 contient les valeurs 5, 7 et 38, la formule =MATCH(7,A1:A3,0) renvoie le nombre 2, car 7 est le deuxième élément de la plage.
CHOISIR	Utilisez cette fonction pour sélectionner l'une des 254 valeurs possibles à partir du rang d'index. Ainsi, si les arguments valeur1 à valeur7 représentent les jours de la semaine, la fonction CHOISIR renvoie l'un de ces jours lorsqu'un nombre compris entre 1 et 7 est utilisé en tant que no_index.
DATE	Utilisez cette fonction pour renvoyer le numéro de série séquentiel qui représente une date particulière. Cette fonction est particulièrement utile lorsque l'année, le mois et le jour sont présentés par formules ou références de cellules. Par exemple, une feuille de calcul peut contenir des dates dans un format non reconnu par Excel, tel que AAAAMMJJ. Utilisez la fonction DATEDIF pour calculer le nombre de jours, de mois ou d'années qui séparent deux dates.
JOURS	Utilisez cette fonction pour renvoyer le nombre de jours qui séparent deux dates.
TROUVE, TROUVERB	Trouve et TROUVERB localiser une chaîne de texte dans une deuxième chaîne de texte. Ils renvoient le numéro de la position de départ de la première chaîne de texte à partir du premier caractère de la deuxième chaîne de texte.
INDEX	Utilisez cette fonction pour renvoyer une valeur ou une référence à une valeur provenant d'un tableau ou d'une plage.

1.11 - Excel : Quelques fonctions



The screenshot displays the Excel 'fonction' (function) search pane. The search bar at the top contains the text 'fonction'. Below the search bar, there are several sections:

- Meilleure action**: This section shows the top action for the search, which is 'sin()' followed by 'Fonction'.
- Actions**: This section lists various actions available for the search term. The first action is 'Insérer une fonction' (Insert a function), followed by 'Récentes' (Recent).
- Rechercher dans le document**: This section shows the search results within the document. It includes a search bar with the text '« fonction »' and a link to 'Afficher plus de résultats de recherche pour « fonction »' (Show more search results for « fonction »).

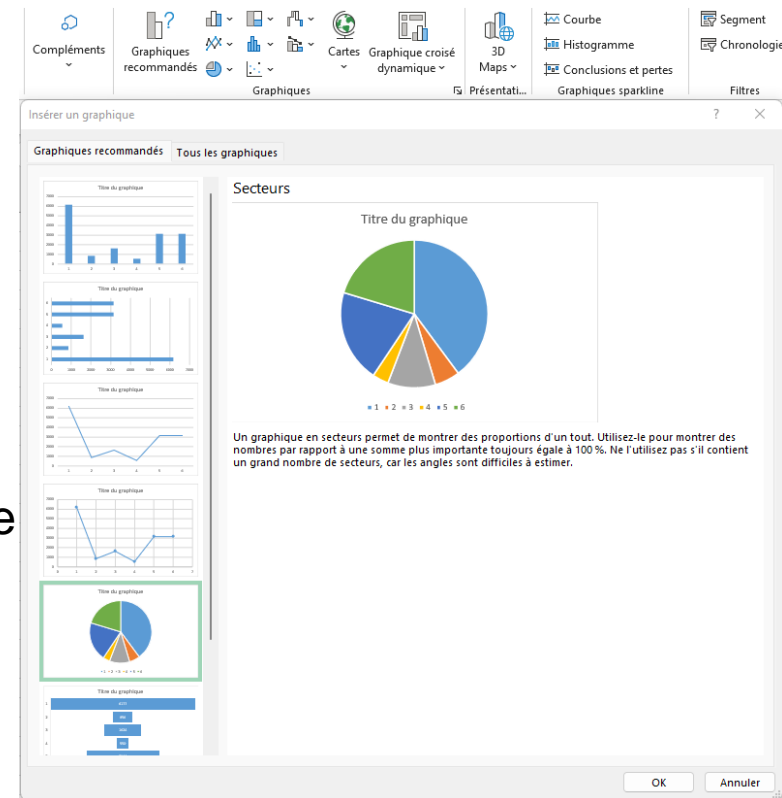
The background of the screenshot shows the Excel ribbon with the 'Formules' (Formulas) tab selected. The ribbon includes the 'Statistiques' (Statistics) group, which is currently expanded, showing a list of statistical functions such as AVERAGEA, BETA.INVERSE.N, CENTILE.EXCLUDE, CENTILE.INCLUDE, CENTREE.REDUITE, CHISQ.TEST, COEFFICIENT.ASYMETRIE, COEFFICIENT.ASYMETRIE.P, COEFFICIENT.CORRELATION, COEFFICIENT.DETERMINATION, COVARIANCE.PEARSON, COVARIANCE.STANDARD, CROISSANCE, and DROITEREG.

1.12 - Excel : Réalisation d'un graphique

Budget d'une copropriété

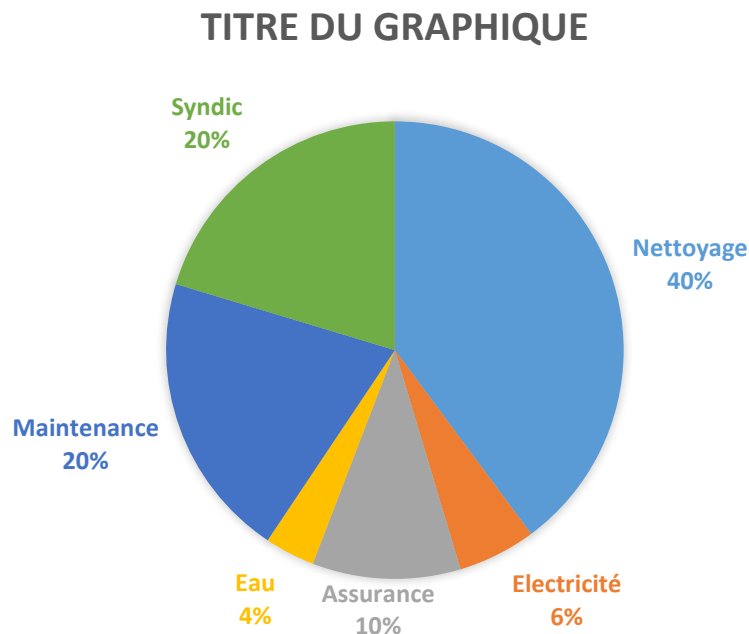
	A	B
1	Nettoyage	6177
2	Electricité	856
3	Assurance	1624
4	Eau	550
5	Maintenance	3144
6	Syndic	3153

1. Sélectionnez les données de la plage A1:B6
2. Exécutez la commande Insertion -> Graphique
3. Type de Graphique



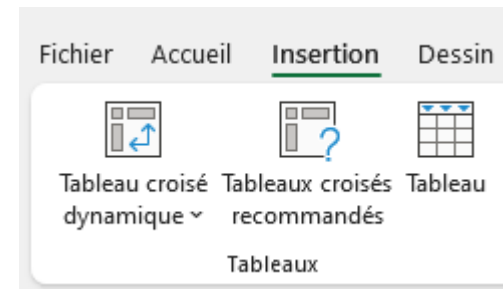
1.12 - Excel : Réalisation d'un graphique

4. Titre : Répartition des charges de l'exercice 2008
5. Etiquettes de données cochez : Nom de Catégorie et pourcentage



1.13 - Excel : Introduction - Tableau Croisé dynamique

Un tableau croisé dynamique est un outil disponible sur Excel (ou sur les autres tableurs) qui **vous sert à analyser vos bases de données** (c'est-à-dire vos données brutes).



<https://ams-training.com/wp-content/uploads/2021/01/Tableaux-et-Graphiques-Crois%C3%A9s-Dynamiques-2010.pdf>
<https://www.morpheus-formation.fr/blog/excel/comment-faire-un-tableau-croise-dynamique-sur-excel/>

1.13 - Excel : Introduction - Tableau Croisé

	A	B	C	D
1	Nom	salaire	performance	bonus
2	Sacha	2000	1	200
3	Dexio	1200	2	120
4	Ondin	1300	3	117
5	Auroro	1400	4	126
6	Kahili	1500	5	105
7	Flora	1600	6	112

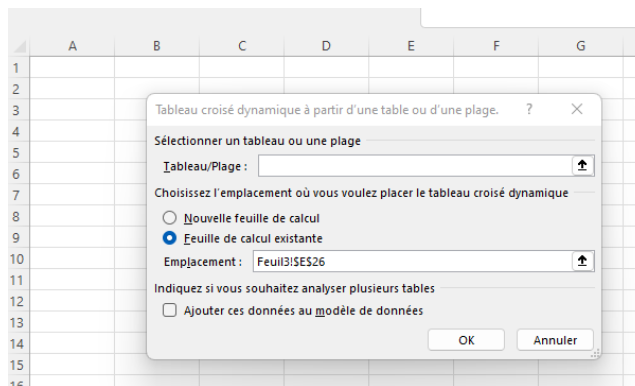


Tableau croisé dynamique à partir d'une table ou d'une plage. ? X

Sélectionner un tableau ou une plage

Tableau/Plage : Feuil2!\$A\$1:\$D\$7

Choisissez l'emplacement où vous voulez placer le tableau croisé dynamique

☐ Nouvelle feuille de calcul

☒ Feuille de calcul existante

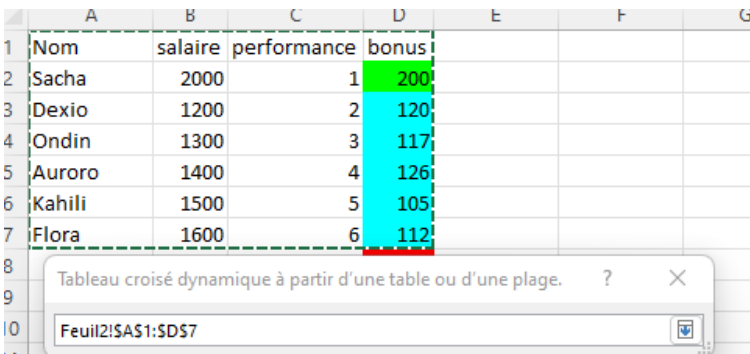
Emplacement : Feuil3!\$E\$26

Indiquez si vous souhaitez analyser plusieurs tables

☐ Ajouter ces données au modèle de données

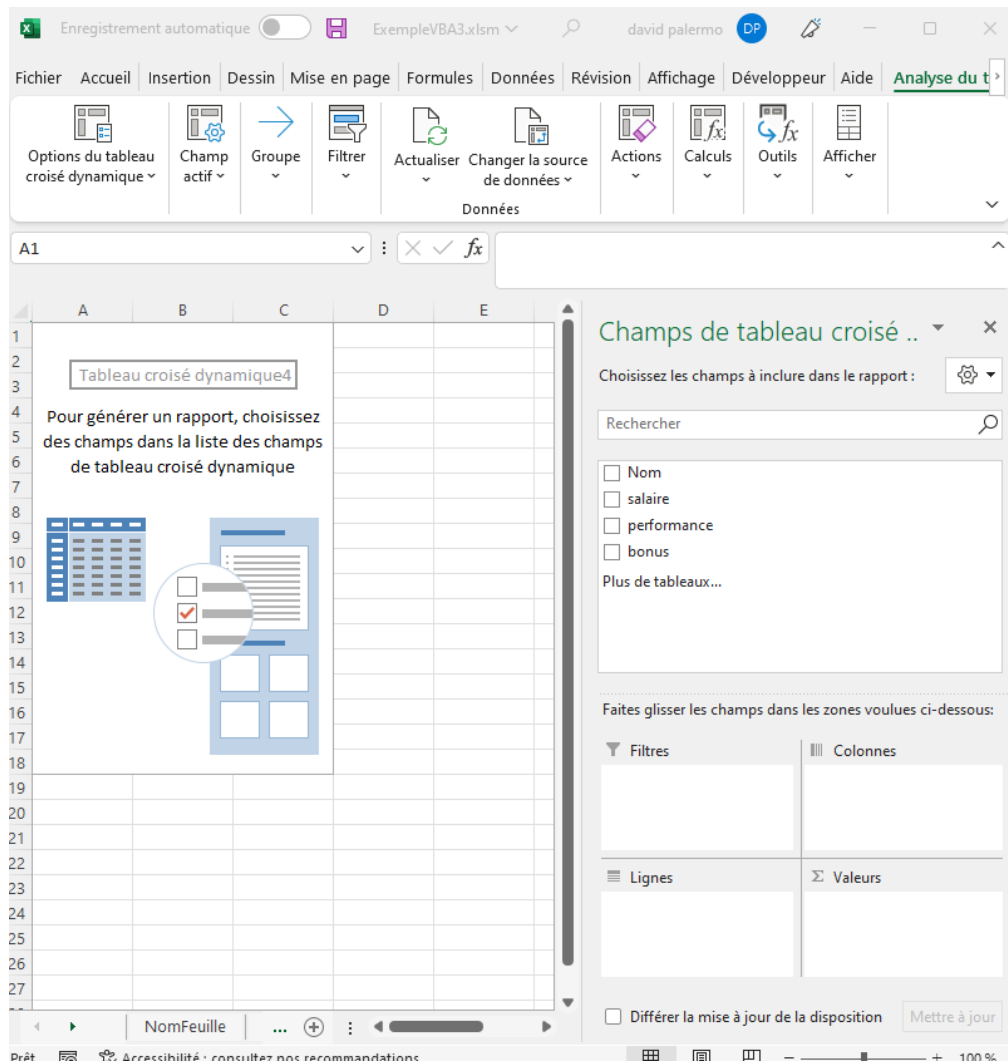
OK

Annuler



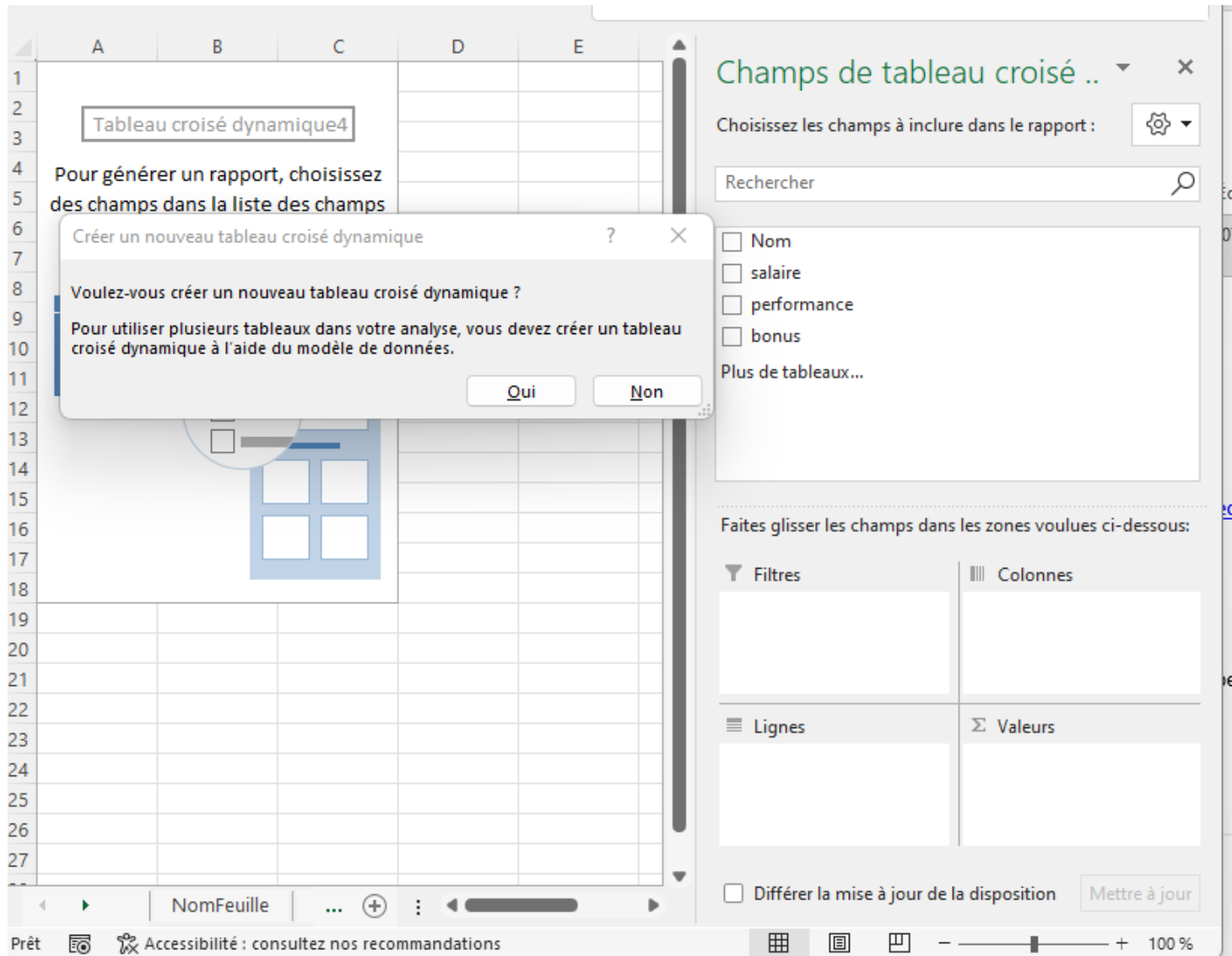
	A	B	C	D	E	F	G
1	Nom	salaire	performance	bonus			
2	Sacha	2000	1	200			
3	Dexio	1200	2	120			
4	Ondin	1300	3	117			
5	Auroro	1400	4	126			
6	Kahili	1500	5	105			
7	Flora	1600	6	112			

1.13 - Excel : Introduction - Tableau Croisé



The screenshot shows the Microsoft Excel interface with the 'Analyse de données' (Data Analysis) ribbon selected. The ribbon includes options like 'Options du tableau croisé dynamique', 'Champ actif', 'Groupe', 'Filtrer', 'Actualiser', 'Changer la source de données', 'Actions', 'Calculs', 'Outils', and 'Afficher'. The 'Tableau croisé dynamique' (PivotTable) task pane is open on the right, titled 'Champs de tableau croisé ..'. It contains a search bar and a list of fields: 'Nom', 'salaire', 'performance', and 'bonus'. Below the list, there are four zones for dragging fields: 'Filtres' (Filters), 'Colonnes' (Columns), 'Lignes' (Rows), and 'Valeurs' (Values). The main worksheet area shows a grid with columns A through E and rows 1 through 27. A text box in the grid reads: 'Tableau croisé dynamique4. Pour générer un rapport, choisissez des champs dans la liste des champs de tableau croisé dynamique'. A small icon of a pivot table is also visible in the grid.

1.13 - Excel : Introduction - Tableau Croisé



The screenshot shows the Excel interface with a PivotTable Fields task pane on the right and a dialog box in the center. The task pane is titled 'Champs de tableau croisé ..' and contains a search bar, a list of fields (Nom, salaire, performance, bonus), and four zones for dragging fields: Filtres, Colonnes, Lignes, and Valeurs. The dialog box is titled 'Créer un nouveau tableau croisé dynamique' and asks 'Voulez-vous créer un nouveau tableau croisé dynamique ?' with 'Oui' and 'Non' buttons.

Champs de tableau croisé ..

Choisissez les champs à inclure dans le rapport :

Rechercher

☐ Nom
☐ salaire
☐ performance
☐ bonus
 Plus de tableaux...

Faites glisser les champs dans les zones voulues ci-dessous:

Filtres	Colonnes
Lignes	Valeurs

☐ Différer la mise à jour de la disposition **Mettre à jour**

Créer un nouveau tableau croisé dynamique

Voulez-vous créer un nouveau tableau croisé dynamique ?

Pour utiliser plusieurs tableaux dans votre analyse, vous devez créer un tableau croisé dynamique à l'aide du modèle de données.

Oui **Non**

1.13 - Excel : Introduction - Tableau Croisé

Options du tableau croisé dynamique ▾ Champ actif ▾ Groupe ▾

Insérer un segment
Insérer une chronologie
Connexions de filtre

Actualiser
Changer la source de données ▾

Actions
Calculs
Outils
Afficher

A3 : Étiquettes de lignes

	A	B	C	D	E	F
1						
2						
3	Étiquettes de lignes ▾	Somme de salaire	Somme de performance			
4	Auroro	1400	4			
5	Dexio	1200	2			
6	Flora	1600	6			
7	Kahili	1500	5			
8	Ondin	1300	3			
9	Sacha	2000	1			
10	Total général	9000	21			
11						
12						
13						
14						
15						
16						
17						
18						
19						
20						
21						
22						
23						
24						
25						
26						
27						

Champs de tableau croisé ..

Actif Tous

Choisissez les champs à inclure dans le rapport :

Rechercher

Plage

- ☒ Nom
- ☒ salaire
- ☒ performance
- ☐ bonus

Faites glisser les champs dans les zones voulues ci-dessous:

Filtres

Colonnes

Σ Valeurs

Lignes

Nom

Σ Valeurs

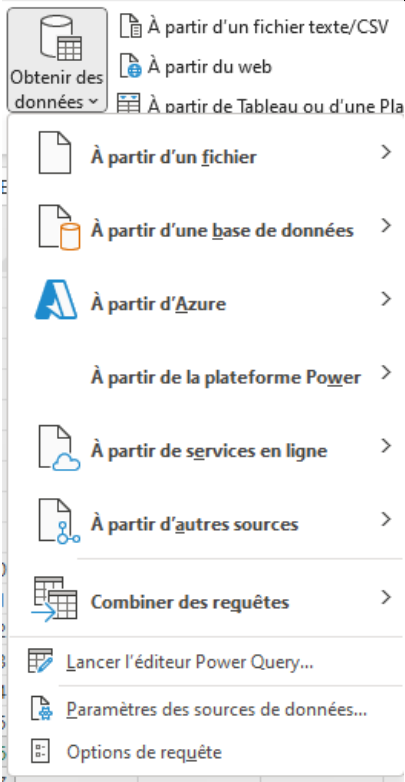
Somme de salaire

Somme de performance

☐ Différer la mise à jour de la disposition Mettre à jour

Prêt Accessibilité : consultez nos recommandations

1.14 - Excel : Introduction – Base de donnée



Excel peut permettre de

- Créer une base de données
- Récupérer des données d'une base de données externe

<https://support.microsoft.com/fr-fr/office/fonctions-de-base-de-donn%C3%A9es-r%C3%A9f%C3%A9rence-ad87e69b-fc20-4d3d-9d52-d7dc023f5c23>

<https://blog.hubspot.fr/marketing/base-de-donnees-excel>

<https://support.microsoft.com/fr-fr/office/cr%C3%A9er-un-mod%C3%A8le-de-donn%C3%A9es-dans-excel-87e7a54c-87dc-488e-9410-5c75dbcb0f7b>

1.15 - Excel : Bibliographies et Lien Utiles

- <https://www.coursinfo.fr/excel/>
- <https://www.excel-pratique.com/fr/vba/>



2 - VBA : Sommaire

- 2.1 L'onglet Développeur
- 2.2 L'enregistreur de macro
- 2.3 Editeur de macro
- 2.4 Ecriture du code VBA
- 2.5 Classeur / Feuille/ Cellule
- 2.6 Les feuilles excel
- 2.7 Les propriétés
- 2.8 Instruction With
- 2.9 les variables
- 2.10 Tableaux Statiques
- 2.11 Tableaux Dynamiques
- 2.12 Introduction de la portée et de la visibilité - Private & Public
- 2.13 Fonction MsgBox
- 2.14 Fonction InputBox
- 2.15 Gestion erreur
- 2.16 les boucles : do...Loop, For...Next, For Each...Next
- 2.17 If...Then...Else
- 2.18 Select Case
- 2.19 Les types de procédure
- 2.20 Procédure Sub
- 2.21 Procédure Function
- 2.22 Exemple Function & Sub
- 2.23 Passage d'argument des procédures
- 2.24 UserForme

Visual Basic pour Applications est le langage de programmation des applications de Microsoft Office.

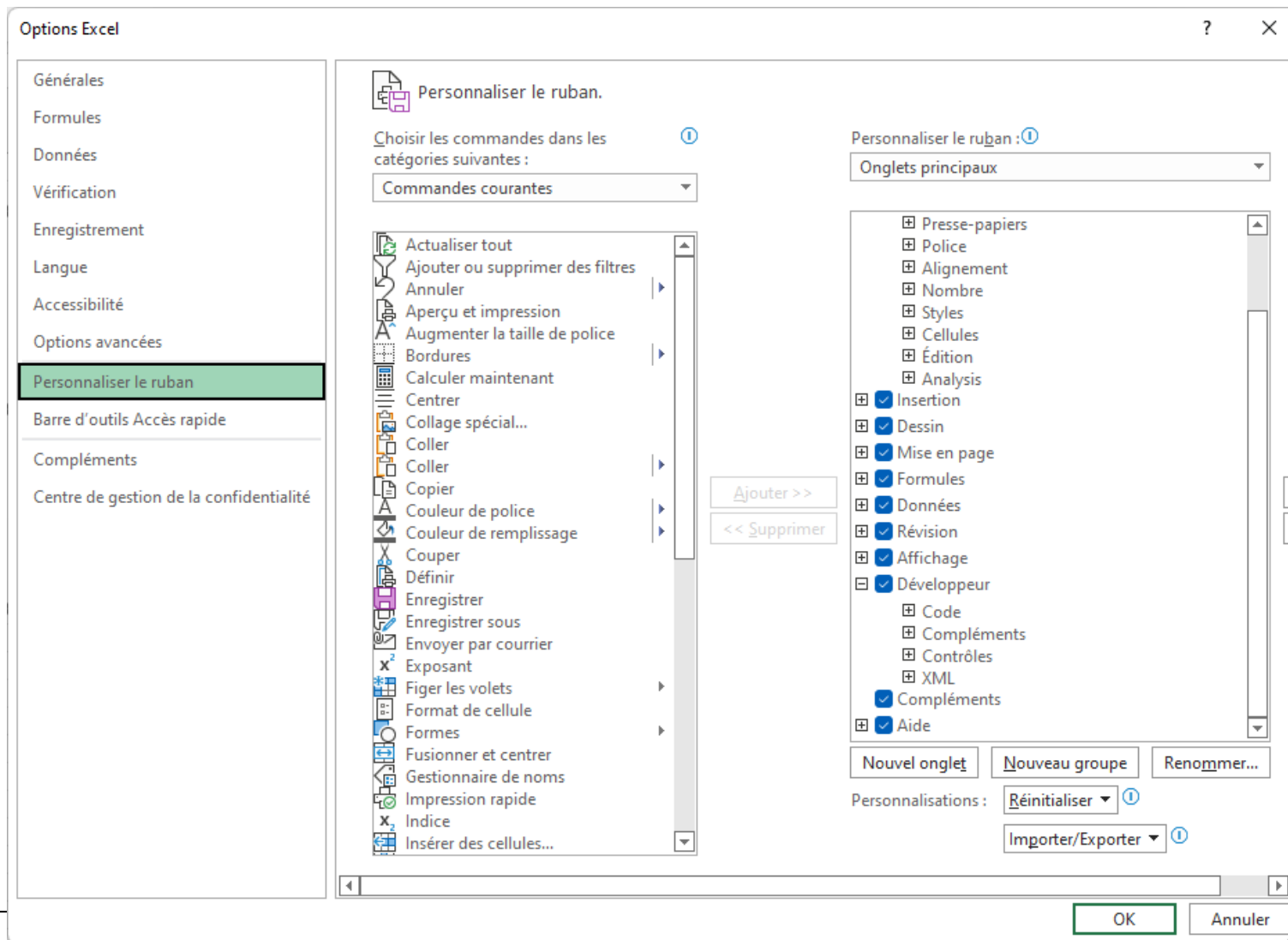
VBA sert a :

- Automatisation et répétition
- Extensions de l'interaction utilisateur

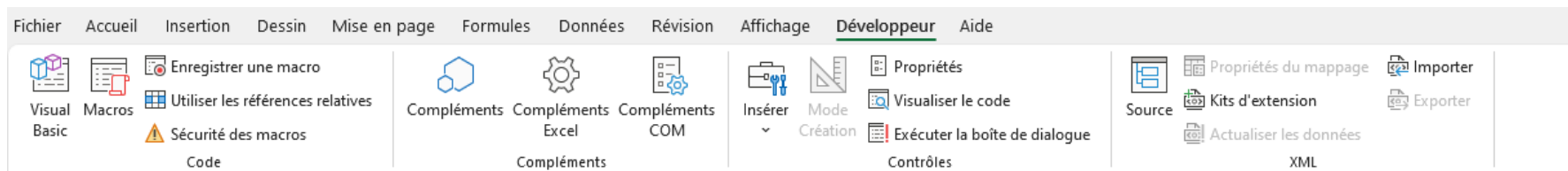
<https://docs.microsoft.com/fr-fr/office/vba/library-reference/concepts/getting-started-with-vba-in-office>

2.1 - VBA : L'onglet Développeur

Faire apparaître l'onglet développeur : Fichier -> option -> personnaliser le ruban

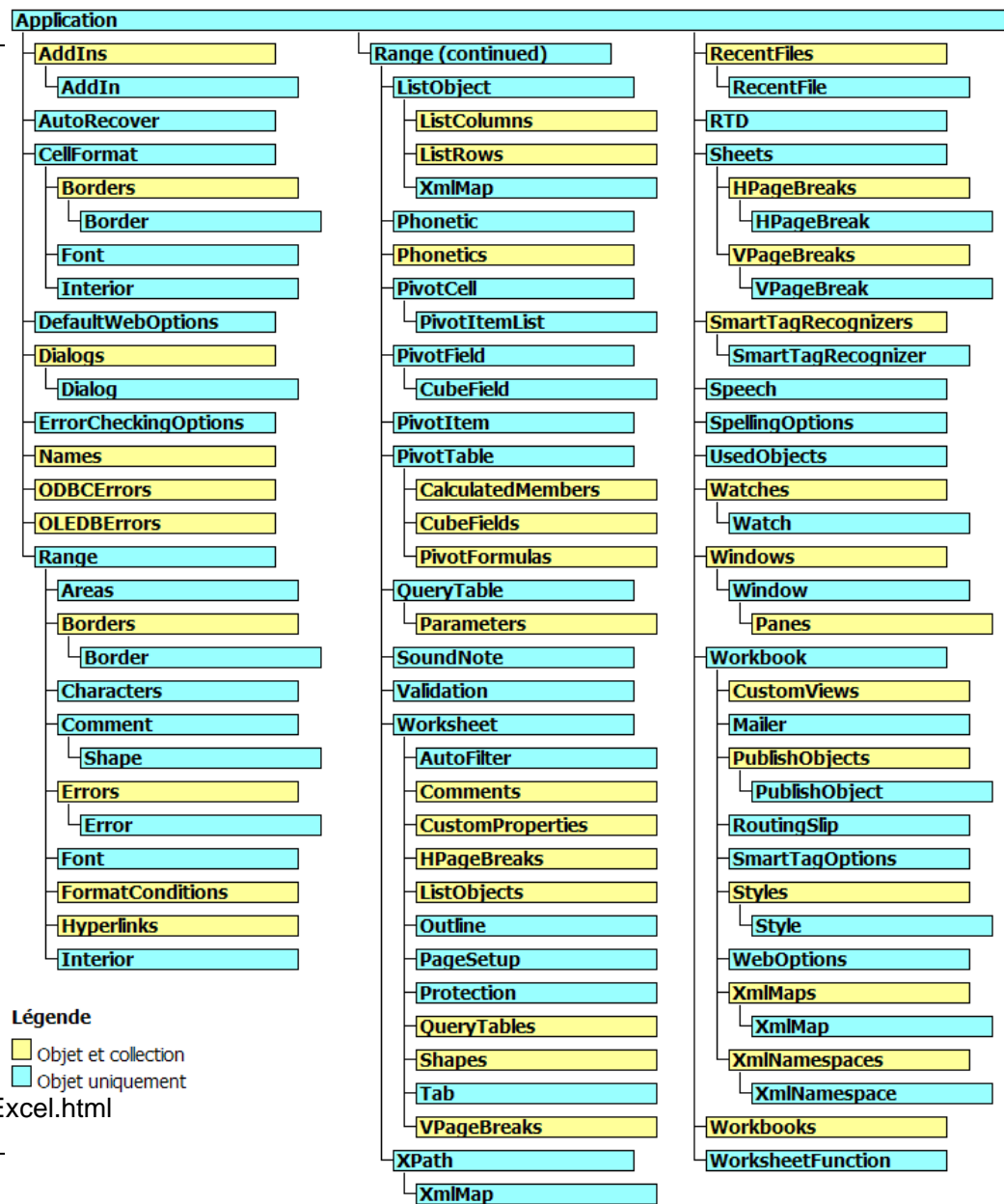


2.1 - VBA : L'onglet Développeur



2.1 - VBA : Les Objets Excel

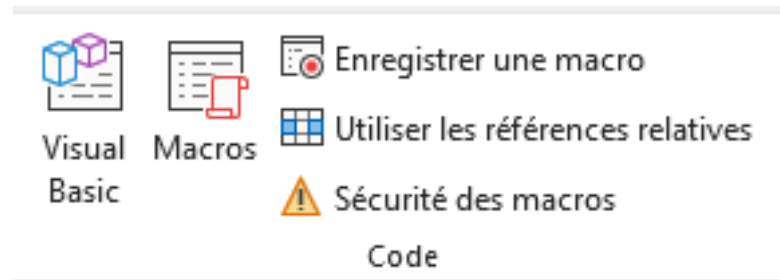
Modèle objet Microsoft Excel



https://tutoriel-vba.espaceweb.usherbrooke.ca/vba_objetsExcel.html

2.2 - VBA : L'enregistreur de macro

L'enregistreur de macro sert pour automatiser une tâche répétitive, vous pouvez enregistrer une macro avec l'enregistreur de macro dans Microsoft Excel.



2.2 - VBA : L'enregistreur de macro

	A	B	C	D
1	Nom	Prénom	Age	Majeur/Mineur
2	Barbe	Julie	15	Mineur
3	Dupont	Paul	18	Majeur
4	Maathai	Wangari	61	Majeur
5	Pankhurst	Emmeline	70	Majeur
6	Curie	Marie	67	Majeur
7	Hamilton	Margaret	85	Majeur
8				
9				

 Enregistrer une macro

Enregistrer une macro

?

×

Nom de la macro :

MiseEnForme

Touche de raccourci :

Ctrl+

Enregistrer la macro dans :


Ce classeur

Description :

OK


Annuler

 Visual Basic

 Macros

☐ Arrêter l'enregistrement

☒ Utiliser les références relatives

 Sécurité des macros

Code

2.2 - VBA : L'enregistreur de macro

Visual Basic

Enregistrer une macro

Utiliser les références relatives

Sécurité des macros

Code

Orthographe

C32

	A	B	C	D
1	Nom	Prénom	Age	Majeur/Mineur
2	Barbe	Julie	15	Mineur
3	Dupont	Paul	18	Majeur
4	Maathai	Wangari	61	Majeur
5	Pankhurst	Emmeline	70	Majeur
6	Curie	Marie	67	Majeur
7	Hamilton	Margaret	85	Majeur
8				



Macro

Nom de la macro :

MiseEnForme

MiseEnForme

Exécuter

Pas à pas détaillé

Modifier

Créer

Supprimer

Options...

Macros dans : Tous les classeurs ouverts

Description

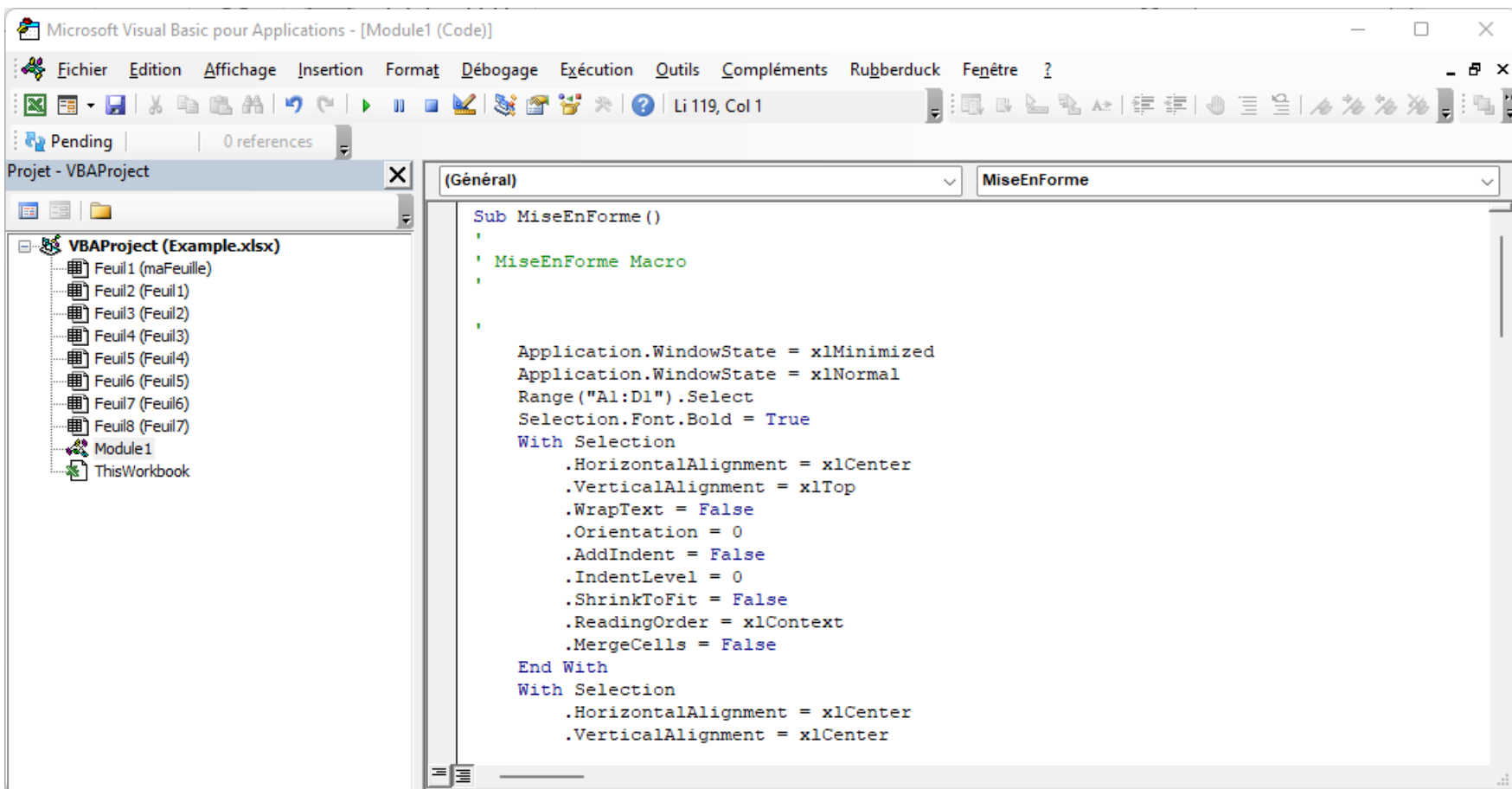
Annuler

	A	B	C	D	E
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					

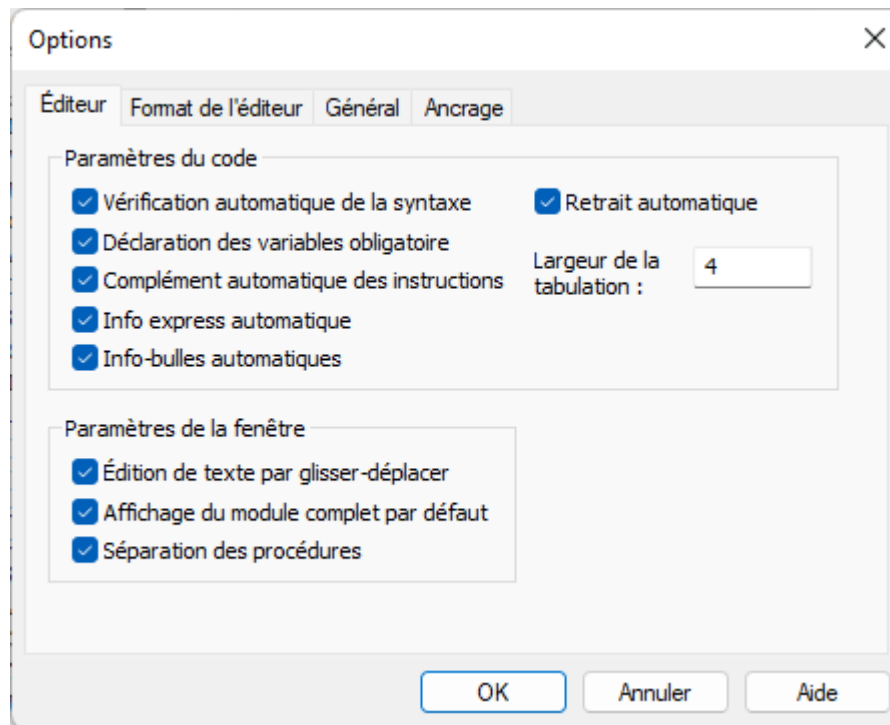
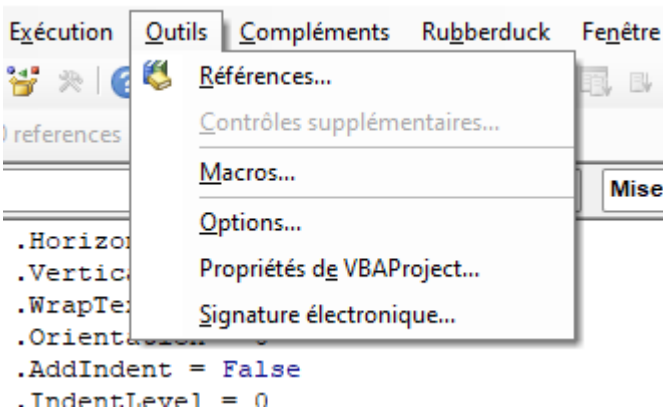
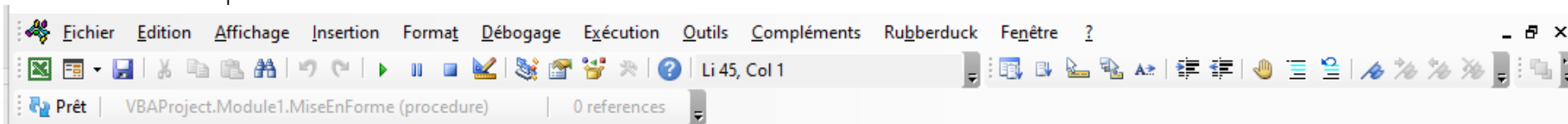
2.3 - VBA : Editeur de macro



Visual
Basic



2.3 - VBA : Editeur de macro



2.4 - VBA : Ecriture du code VBA

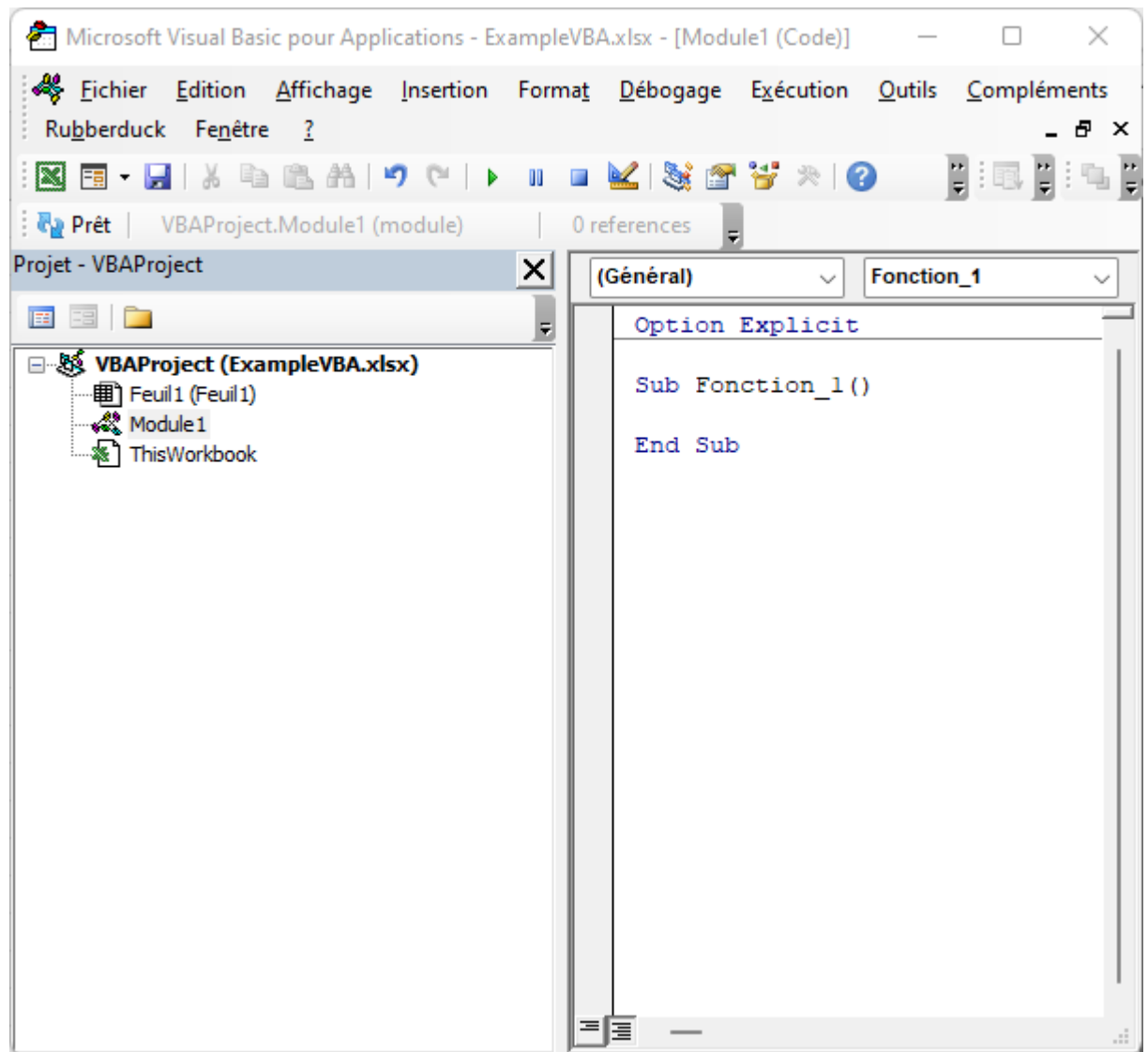
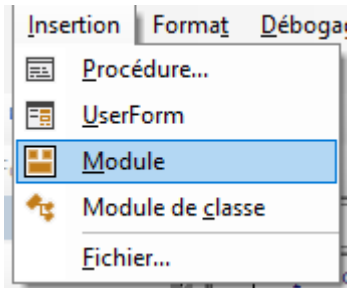
Le code VBA s'écrit :

- dans les modules à l'intérieur de procédures ou de fonctions.
- Dans une feuille de saisie pour chaque classeur et pour chaque feuille du classeur.
- Dans ThisWorkbook: Cet objet représente un classeur Microsoft Excel.

2.5 - VBA : Classeur / Feuille/ Cellule

- Les classeurs sont désignés par le mot « Workbook ».
 - En VBA ils peuvent être ouverts, fermés, enregistrés, activés, masqués supprimés
 - <https://docs.microsoft.com/fr-fr/office/vba/api/excel.workbook>
- Les feuilles de calcul sont désignées par le mot « Worksheet ».
 - En VBA ils peuvent être créer, supprimer, activés, masqués supprimés ...
 - <https://docs.microsoft.com/fr-fr/office/vba/api/excel.worksheet>
- Une plage de cellules est désignée par le mot « Range » , une cellule a la fois par « Range » ou « Cells » et une colonne par « Range » ou « Columns »
 - [https://docs.microsoft.com/fr-fr/office/vba/api/excel.range\(object\)](https://docs.microsoft.com/fr-fr/office/vba/api/excel.range(object))
 - <https://docs.microsoft.com/fr-fr/office/vba/api/excel.worksheet.cells>
 - <https://docs.microsoft.com/fr-fr/office/vba/api/excel.worksheet.columns>

2.6 - VBA : Les feuilles excel





2.6 - VBA : Les feuilles excel

Enregistrement automatique ExampleVBA.xlsx • Enregistré ✓ david palermo DP

Fichier Accueil Insertion Dessin Mise en page Formules Données Révision Affichage **Développeur** Aide

Visual Basic Macros Compléments Compléments Excel Compléments COM Insérer Mode Création Source Propriétés du mappage Kits d'extension Actualiser les données XML Importer Exporter

Code Compléments

abc Orthographe

C4

A B C D E H I J K

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

Affecter une macro

Nom de la macro :

Fonction_1
Fonction_1

Modifier
Enregistrer...

Macros dans : Tous les classeurs ouverts

Description

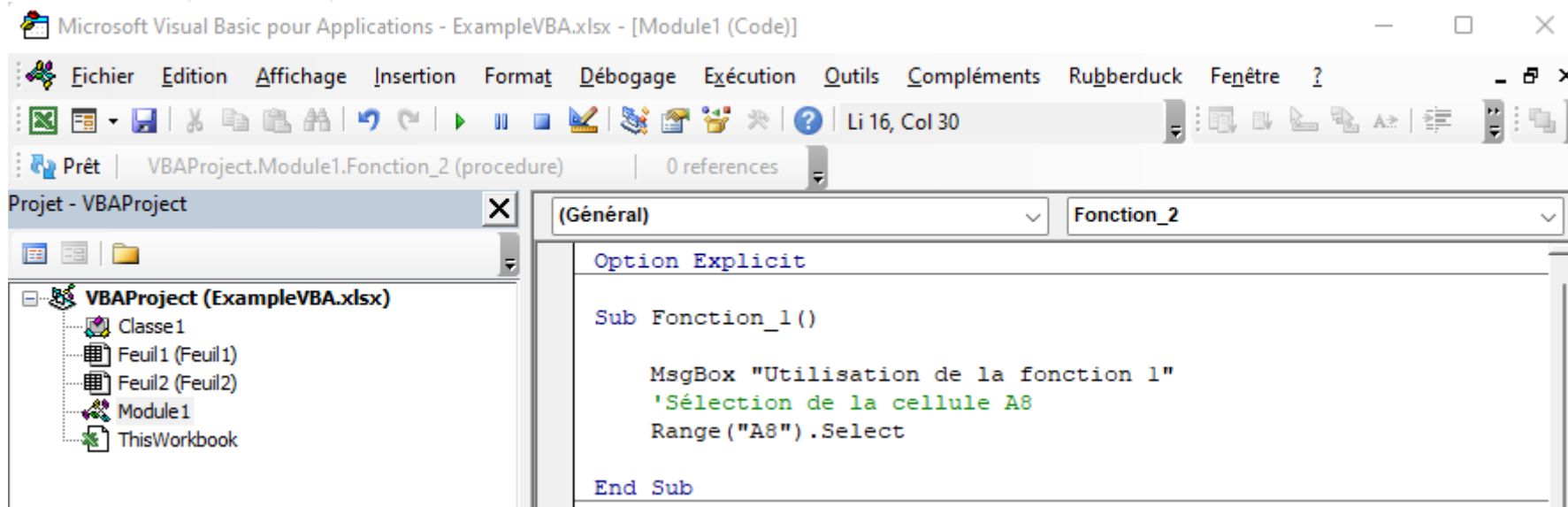
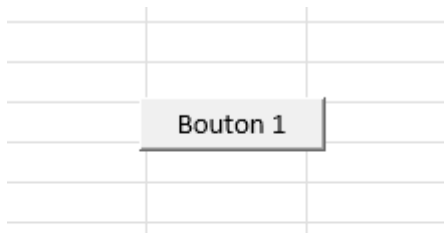
OK Annuler

Feuil1

Prêt Accessibilité : vérification terminée

100%

2.6 - VBA : Les feuilles excel



2.6 - VBA : Les feuilles excel

Microsoft Visual Basic pour Applications - ExampleVBA.xlsx - [Module1 (Code)]

Fichier Edition Affichage Insertion Format Débogage Exécution Outils Compléments Rubberduck Fenêtre ?

Li 16, Col 30

Prêt VBAProject.Module1.Fonction_2 (procedure) 0 references

Projet - VBAProject

VBAProject (ExampleVBA.xlsx)

- Classe1
- Feuil1 (Feuil1)
- Feuil2 (Feuil2)
- Module1
- ThisWorkbook

(Général) Fonction_2

Option Explicit

```

Sub Fonction_1()

    MsgBox "Utilisation de la fonction 1"
    'Sélection de la cellule A8
    Range("A8").Select

End Sub

Sub Fonction_2()

    MsgBox "Utilisation de la fonction 2"

    'Activation de la feuille 2
    Sheets("Feuil2").Activate

    'Sélection de la cellule A8
    Range("A1:A8").Select

End Sub

```

Bouton 1

Bouton 2

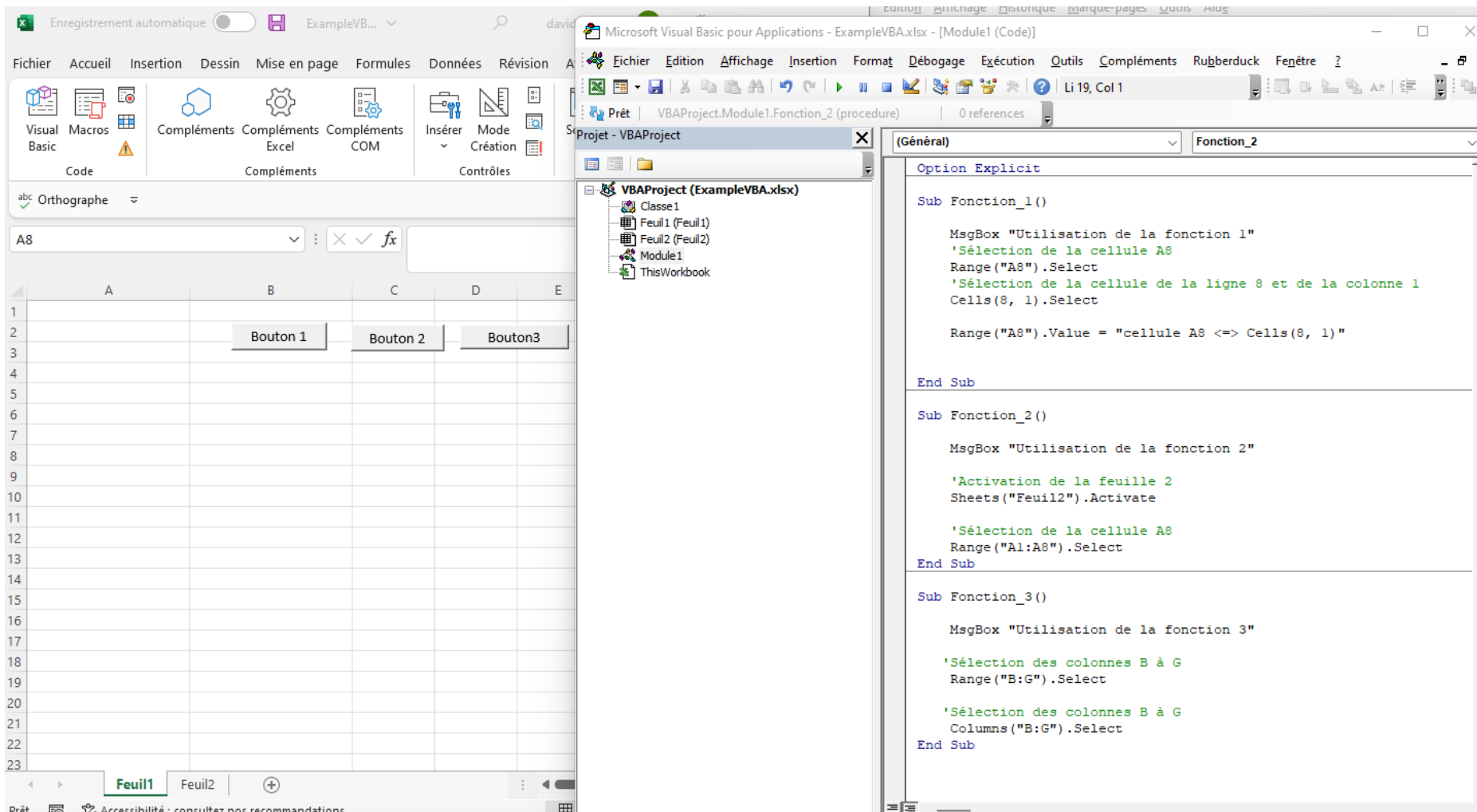
5

6

Feuil1 Feuil2

Prêt Accessibilité : consultez nos reco

2.6 - VBA : Les feuilles excel



Microsoft Visual Basic pour Applications - ExampleVBA.xlsx - [Module1 (Code)]

Fichier Edition Affichage Insertion Format Débogage Exécution Outils Compléments Rubberduck Fenêtre ?

Prêt | VBAProject.Module1.Fonction_2 (procedure) | 0 references

Projet - VBAProject

- VBAProject (ExampleVBA.xlsx)
 - Classe1
 - Feuil1 (Feuil1)
 - Feuil2 (Feuil2)
 - Module1
 - ThisWorkbook

(Général) | Fonction_2

Option Explicit

```

Sub Fonction_1()

    MsgBox "Utilisation de la fonction 1"
    'Sélection de la cellule A8
    Range("A8").Select
    'Sélection de la cellule de la ligne 8 et de la colonne 1
    Cells(8, 1).Select

    Range("A8").Value = "cellule A8 <=> Cells(8, 1)"

End Sub

Sub Fonction_2()

    MsgBox "Utilisation de la fonction 2"

    'Activation de la feuille 2
    Sheets("Feuil2").Activate

    'Sélection de la cellule A8
    Range("A1:A8").Select

End Sub

Sub Fonction_3()

    MsgBox "Utilisation de la fonction 3"

    'Sélection des colonnes B à G
    Range("B:G").Select

    'Sélection des colonnes B à G
    Columns("B:G").Select

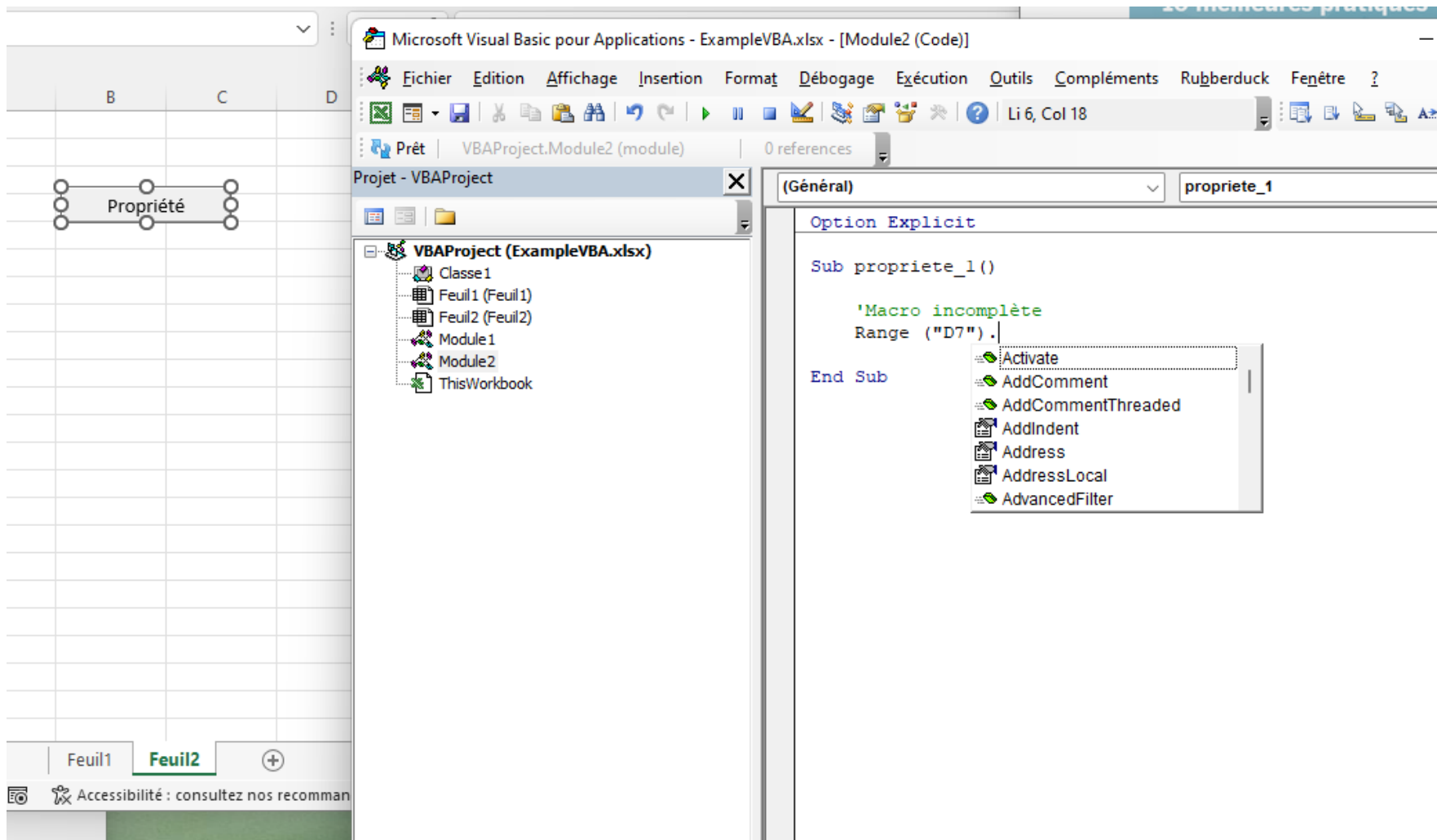
End Sub

```


2.7 - VBA : Les propriétés

Agir sur le contenu et l'apparence des cellules et des feuilles.

2.7 - VBA : Les propriétés



Microsoft Visual Basic pour Applications - ExampleVBA.xlsx - [Module2 (Code)]

Fichier Edition Affichage Insertion Format Débogage Exécution Outils Compléments Rubberduck Fenêtre ?

Prêt | VBAProject.Module2 (module) | 0 references

Projet - VBAProject

VBAProject (ExampleVBA.xlsx)

- Classe1
- Feuil1 (Feuil1)
- Feuil2 (Feuil2)
- Module1
- Module2
- ThisWorkbook

(Général) | propriete_1

Option Explicit

```
Sub propriete_1()
    'Macro incomplète
    Range ("D7").
End Sub
```

- Activate
- AddComment
- AddCommentThreaded
- AddIndent
- Address
- AddressLocal
- AdvancedFilter

Feuil1 Feuil2 (+)

Accessibilité : consultez nos recommandations

2.7 - VBA : Les propriétés

Option Explicit

```
Sub propriete_1()  
  
    'Modifier la valeur de la case D7  
    Range("D7").Value = 74  
  
    Range("D8").Value = 84  
  
    Sheets("Feuil2").Range("D9").Value = "Case D9"  
  
    Sheets("Feuil2").Range("D10").Value = "Case D10"  
  
    Workbooks("ExampleVBA.xlsx").Sheets("Feuil2").Range("D11").Value = "Case D11"  
  
    'Modifier la taille du texte des cellules D1 à D8  
    Range("D1:D8").Font.Size = 18  
  
    'Mettre en gras les cellules D1 à D7  
    Range("D1:A7").Font.Bold = True  
  
    'Mettre en italique la cellule D9  
    Range("D9").Font.Italic = True  
  
    'Souligner la cellules D10  
    Range("D10").Font.Underline = True  
  
    'Modifier la police de caractères la cellules D11  
    Range("D11").Font.Name = "Arial"  
  
End Sub
```

2.8 - VBA : Instruction With

```
Sub proprietes_21()
```

```
    Sheets("Feuil2").Range("E1").Font.Bold = True  
    Sheets("Feuil2").Range("E1").Font.Size = 15  
    Sheets("Feuil2").Range("E1").Font.Italic = True  
    Sheets("Feuil2").Range("E1").Font.Name = "Arial"  
    Sheets("Feuil2").Range("E1").Value = 21
```

```
End Sub
```

```
Sub proprietes_22()
```

```
    With Sheets("Feuil2").Range("E2")  
        .Font.Bold = True  
        .Font.Size = 15  
        .Font.Italic = True  
        .Font.Name = "Arial"  
        .Value = "22"
```

```
    End With
```

```
End Sub
```

2.8 - VBA : Instruction With

```
Sub proprietes_22()
```

```
    With Sheets("Feuil2").Range("E2")  
        .Font.Bold = True  
        .Font.Size = 15  
        .Font.Italic = True  
        .Font.Name = "Arial"  
        .Value = "22"  
    End With
```

```
End Sub
```

```
Sub proprietes_23()
```

```
    With Sheets("Feuil2").Range("G2")  
        .Font.Color = RGB(0, 255, 0)  
        .Font.Size = 15  
        .Font.Italic = True  
        .Font.Name = "Arial"  
        .Value = "23"  
    End With
```

```
    With Sheets("Feuil2").Range("G3")  
        .Font.ColorIndex = 4  
        .Borders.Color = RGB(255, 0, 0)  
        .Value = "23"  
        .Interior.Color = RGB(0, 0, 0)
```

```
End With
```

```
Sheets("Feuil2").Tab.Color = RGB(0, 255, 255)
```

```
End Sub
```

2.9 - VBA : les variables

Dim ma_variable **As** String

- **Dim** : déclaration de la variable
- **ma_variable** : nom choisi pour cette variable
- **As** : déclaration du type de la variable
- **String** : type de la variable

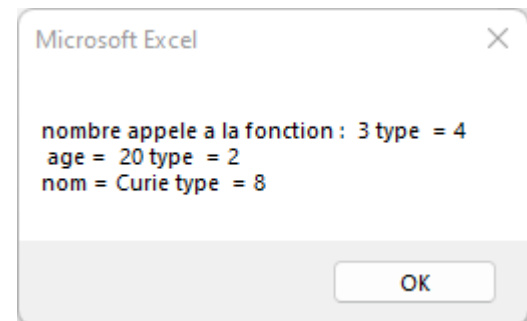
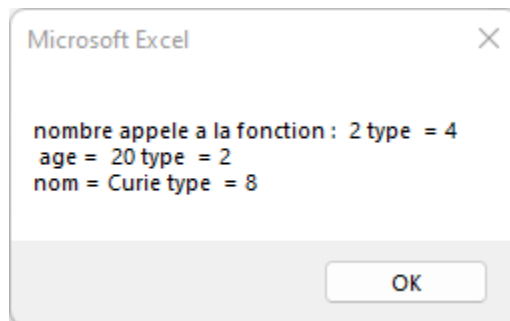
2.9 - VBA : les variables

Nom	Type	Détails	Numero
Variant	Tous	Tout type de données (type par défaut si la variable n'est pas déclarée).	0
Null	aucun		1
Integer	Numérique	Nombre entier de -32 768 à 32 767	2
Long	Numérique	Nombre entier de -2 147 483 648 à 2 147 483 647	3
Single	Numérique	-3,402823E38 à -1,401298E-45 (pour les valeurs négatives); 1.401298E - 45 à 3.402823E38 (pour les valeurs positives)	4
Double	Numérique	-1,79769313486232E308 à -4,94065645841247E-324 (valeurs négatives); 4,94065645841247E-324 à 1,79769313486232E308 (pour les valeurs positives)	5
Currency	Numérique	Nombre à décimale fixe de -922 337 203 685 477,5808 à 922 337 203 685 477,5807	6
Date	Date	Date et heure.	7
String	Texte	Texte.	8
Boolean	Boolean	True (vrai) ou False (faux).	11
Byte	Numérique	Nombre entier de 0 à 255.	17
Object	Objet	Objet Microsoft.	

<https://excelcorpo.com/types-de-variables-dans-excel-vba/>

2.9 - VBA : les variables

```
Sub Variable_1()  
    Static Cpt As Single  
    Cpt = Cpt + 1  
  
    Dim age As Integer  
    age = 20  
  
    Dim nom As String  
    nom = "Curie"  
  
    MsgBox ("nombre appele a la fonction : " & Str(Cpt) & " type = " & VarType(Cpt) & Chr(13)  
        " age = " & Str(20) & " type = " & VarType(age) & Chr(13) & Chr(10) & _  
        "nom = " & nom & " type = " & VarType(nom))  
End Sub
```



2.9 - VBA : les variables

```
Option Explicit
```

```
Public Type Contacts
```

```
    Nom As String
```

```
    Age As Integer
```

```
End Type
```

```
Sub Variable_2()
```

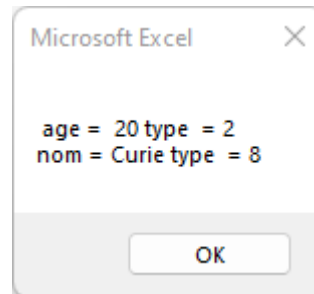
```
    Dim aContacts As Contacts
```

```
    aContacts.Age = 20
```

```
    aContacts.Nom = "Curie"
```

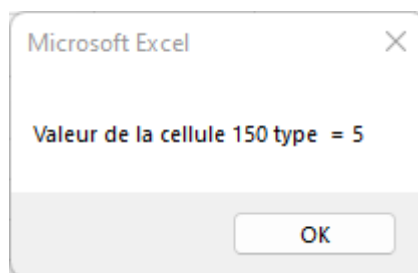
```
    MsgBox (" age = " & Str(20) & " type = " & VarType(aContacts.Age) & Chr(13) & Chr(10) & _  
           "nom = " & aContacts.Nom & " type = " & VarType(aContacts.Nom) )
```

```
End Sub
```



2.9 - VBA : les variables

```
Sub Variable_3()  
  
    Dim Cellule As Range  
    Set Cellule = Range("A1")  
    Cellule.Value = 150  
  
    MsgBox ("Valeur de la cellule " & Cellule.Value & " type  = " & VarType(Cellule.Value))  
End Sub
```



2.10 - VBA : Tableaux Statiques

Un tableau statique est déclaré avec sa taille spécifiée lorsque vous déclarez initialement le tableau.

Option Explicit

```
Sub tableau_1()  
  
    'Tableau de 5 élément de 0 à 4  
    Dim intA(4) As Integer  
    intA(0) = 0  
    intA(1) = 1  
    intA(2) = 2  
    intA(3) = 3  
    intA(4) = 4  
  
    Debug.Print intA(2)  
    'Tableau de 5 élément de 2 à 5  
    Dim intB(2 To 5) As Integer  
    intB(2) = 2  
    intB(3) = 3  
    intB(4) = 4  
    intB(5) = 5  
    Debug.Print intB(2)  
End Sub
```

<https://www.automateexcel.com/fr/vba/declarer-dim-cree-initialiser-tableau>

Une variable Array Dynamique est un tableau dont la taille peut être modifiée au moment de l'exécution.

```
Sub tableau_2()
```

```
'Tableau de 5 élément de 0 à 4
```

```
Dim intA() As Integer
```

```
ReDim intA(4)
```

```
intA(0) = 0
```

```
intA(1) = 1
```

```
intA(2) = 2
```

```
intA(3) = 3
```

```
intA(4) = 4
```

```
ReDim intA(4)
```

```
intA(0) = 0
```

```
intA(1) = 1
```

```
intA(2) = 2
```

```
intA(3) = 3
```

```
intA(4) = 4
```

```
Debug.Print intA(2)
```

```
'Tableau de 5 élément de 2 à 5
```

```
Dim intB() As Integer
```

```
ReDim intB(2 To 5)
```

```
intB(2) = 2
```

```
intB(3) = 3
```

```
intB(4) = 4
```

```
intB(5) = 5
```

```
'Preserve garde les valeurs
```

```
ReDim Preserve intB(2 To 5)
```

```
intA(0) = 0
```

```
intA(1) = 1
```

```
intA(2) = 2
```

```
intA(3) = 3
```

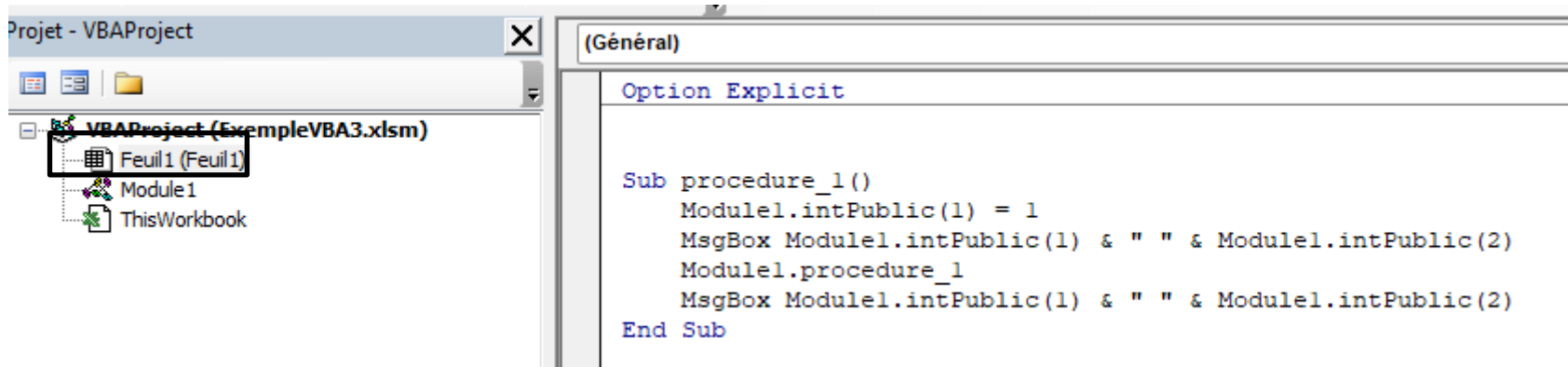
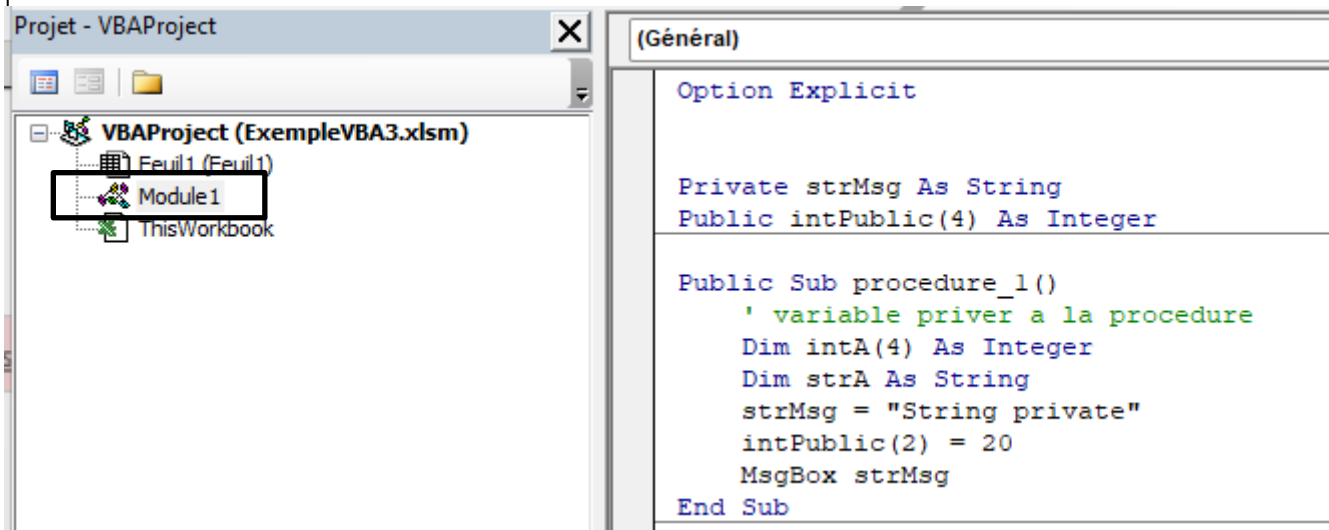
```
intA(4) = 4
```

```
Debug.Print intB(2)
```

```
End Sub
```

- Une variable ou une constante définie dans une procédure n'est pas visible hors de cette procédure
- Les variables au niveau d'un module peuvent être publiques ou privées

2.12 - VBA : Introduction de la portée et de la visibilité - Private & Public

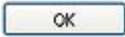





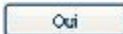
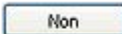


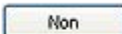
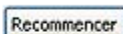







2.13 - VBA : Fonction MsgBox

```
Sub MsgBox_1()  
    Dim Msg, Style, Title, Help, Ctxt, Response, MyString  
    Msg = "Voulez vous continuer ?"           ' Define message.  
    Style = vbYesNo Or vbCritical Or vbDefaultButton2 ' Define buttons.  
    Title = "MsgBox Demonstration"           ' Define title.  
    Help = "DEMO.HLP"                        ' Define Help file.  
    Ctxt = 1000                              ' Define topic context.  
  
    ' Display message.  
    Response = MsgBox(Msg, Style, Title, Help, Ctxt)  
    If Response = vbYes Then                 ' User chose Yes.  
        MyString = "Yes"                   ' Perform some action.  
    Else                                    ' User chose No.  
        MyString = "No"                   ' Perform some action.  
    End If  
End Sub
```

<https://docs.microsoft.com/fr-fr/office/vba/language/reference/user-interface-help/msgbox-function>

2.13 - VBA : Fonction MsgBox

Constantes	Valeurs	Descriptions
vbOKOnly	0	• Affiche le bouton  uniquement.
vbOKCancel	1	• Affiche les boutons  
vbAbortRetryIgnore	2	• Affiche les boutons   
vbYesNoCancel	3	• Affiche les boutons   
vbYesNo	4	• Affiche les boutons  
vbRetryCancel	5	• Affiche les boutons  
vbCritical	16	•  Affiche l'icône Message critique
vbQuestion	32	•  Affiche l'icône Question
vbExclamation	48	•  Affiche l'icône Point d'exclamation
vbInformation	64	•  Affiche l'icône Information
vbDefaultButton1	0	• Le premier bouton est le bouton par défaut.
vbDefaultButton2	256	• Le deuxième bouton est le bouton par défaut.
vbDefaultButton3	512	• Le troisième bouton est le bouton par défaut.
vbDefaultButton4	768	• Le quatrième bouton est le bouton par défaut.
vbApplicationModal	0	• Boîte de dialogue modale. L'utilisateur doit répondre au message affiché dans la zone de message avant de pouvoir continuer de travailler dans l'application en cours.
vbSystemModal	4096	• Modal système. Toutes les applications sont interrompues jusqu'à ce que l'utilisateur réponde au message affiché dans la zone de message.
vbMsgBoxHelpButton	16384	• Ajoute le bouton Aide à la zone de message.
VbMsgBoxSetForeground	65536	• Indique la fenêtre de zone de message comme fenêtre de premier plan.
vbMsgBoxRight	524288	• Le texte est aligné à droite.
vbMsgBoxRtlReading	1048576	• Indique que le texte doit apparaître de droite à gauche sur les systèmes hébraïques et arabes.

2.14 - VBA : Fonction InputBox

InputBox (*Prompt, Title, Default, Left, Top, HelpFile, HelpContextID, Type*)

Nom	Requis/Facultatif	Type de données	Description
<i>Prompt</i>	Requis	String	Message à afficher dans la boîte de dialogue. Il peut s'agir d'une chaîne de caractères, d'un nombre, d'une date ou d'une valeur booléenne (Microsoft Excel contraint la valeur à avoir le type de données String avant de l'afficher). La longueur maximale est de 255 caractères, sans quoi il n'y a pas d'invite, et la méthode de l'application renvoie immédiatement Erreur 2015.
<i>Titre</i>	Facultatif	Variant	Titre de la zone d'entrée. Si vous ne spécifiez pas cet argument, le titre par défaut est « Entrée ».
<i>Par défaut</i>	Facultatif	Variant	Spécifie une valeur qui apparaît dans la zone de texte quand la boîte de dialogue s'affiche. Si cet argument est omis, la zone de texte reste vide. Cette valeur peut être un objet Range .
<i>Left</i>	Facultatif	Variant	Spécifie une position <i>x</i> pour la boîte de dialogue, exprimée en points et calculée à partir du coin supérieur gauche de l'écran.
<i>Top</i>	Facultatif	Variant	Spécifie une position <i>y</i> pour la boîte de dialogue, exprimée en points et calculée à partir du coin supérieur gauche de l'écran.
<i>HelpFile</i>	Facultatif	Variant	Nom du fichier d'aide pour cette zone d'entrée. Si les arguments <i>HelpFile</i> et <i>HelpContextID</i> sont présents, un bouton d'aide apparaît dans la boîte de dialogue.
<i>HelpContextID</i>	Facultatif	Variant	Numéro d'identification du contexte de la rubrique d'aide dans <i>HelpFile</i> .
<i>Type</i>	Facultatif	Variant	Spécifie le type des données renvoyées. Si vous ne spécifiez pas cet argument, la boîte de dialogue renvoie du texte.

<https://docs.microsoft.com/fr-fr/office/vba/api/excel.application.inputbox>

2.14 - VBA : Fonction InputBox

Valeur	Description
0	Une formule.
1	Un nombre.
2	Texte (une chaîne).
4	Valeur logique (True ou False)
8	Référence de cellule, sous la forme d'un objet Range.
16	Une valeur d'erreur, telle que #N/A.
64	Un tableau de valeurs.

Option Explicit

```
Sub InputBox_1()  
  
Dim myNum As Integer  
myNum = Application.InputBox("Enter a number")  
  
Dim myRange As Range  
Set myRange = Application.InputBox(prompt:="Choisissez un champ", Type:=8)  
  
End Sub
```

2.15 - VBA : Gestion erreur

On Error GoTo *line* On Error Resume Next On Error GoTo 0

Statement	Description
On Error GoTo <i>line</i>	<p>Active la routine de gestion des erreurs qui commence à la <i>ligne</i> spécifiée dans l'argument <i>de ligne requis</i>.</p> <p>L'argument de ligne est une <i>étiquette de ligne</i> ou un <i>numéro de ligne</i>.</p> <p>Si une <i>erreur d'run-time</i> se produit, le contrôle se branche en <i>ligne</i>, ce qui active le handler d'erreurs.</p> <p>La ligne <i>spécifiée</i> doit se trouver dans la même procédure que l'instruction On Error ; Dans le cas contraire, une <i>erreur de compilation</i> se produit.</p>
On Error Resume Next	<p>Spécifie qu'en cas d'erreur d'exécution, le contrôle passe à l'instruction qui suit immédiatement l'instruction dans laquelle l'erreur s'est produite et l'exécution se poursuit. Il est recommandé d'utiliser cette formulation plutôt que l'instruction On Error GoTo pour accéder à des objets.</p>
On Error GoTo 0	<p>Invalide dans la procédure actuelle tout gestionnaire d'erreurs validé.</p>

<https://docs.microsoft.com/fr-fr/office/vba/language/reference/user-interface-help/on-error-statement>

2.15 - VBA : Gestion erreur

Option Explicit

```
Sub InputBox_1()  
  
On Error Resume Next  
Dim myNum As Integer  
myNum = Application.InputBox("Enter a number")  
  
On Error GoTo Line  
Dim myRange As Range  
Set myRange = Application.InputBox(prompt:="Choisissez un champ", Type:=8)  
Exit Sub  
  
Line:  
MsgBox "Erreur de champs", vbYesNo + vbCritical  
On Error GoTo 0  
  
End Sub
```

https://www.automateexcel.com/fr/vba/gestion-erreurs/#Declaration_VBA_On_Error

- Do...Loop : exécution en boucle pendant ou jusqu'à ce qu'une condition a la valeur **True**
- For...Next : utilisation d'un compteur pour exécuter des instructions un nombre de fois spécifié
- For Each...Next : répétition d'un groupe d'instructions pour chaque objet d'une collection

<https://docs.microsoft.com/fr-fr/office/vba/language/concepts/getting-started/looping-through-code>

2.16 - VBA : les boucles -> Do...Loop

```
Sub DOLOOP()
    Dim max As Integer
    Dim ligne As Integer
    Dim msg As String

    ligne = 0
    max = 9
    Do
        ligne = ligne + 1
        msg = "A" + CStr(ligne)
        Worksheets("NomFeuille").Range(msg).Value = ligne * 11
    Loop While ligne < max

End Sub
```

	A	B	C
1	11	Remplir A	
2	22		
3	33		
4	44		
5	55		
6	66		
7	77		
8	88		
9	99		
10			

2.16 - VBA : les boucles -> For...Next

```
Sub FORNEXT()
    Dim max As Integer
    Dim ligne As Integer
    Dim msg As String

    max = 9
    For ligne = 1 To max
        msg = "A" + CStr(ligne)
        Worksheets("NomFeuille").Range(msg).Value = 0
    Next ligne
End Sub
```

	A	B	C	D
1	0	Remplir A	Effacer A	
2	0			
3	0			
4	0			
5	0			
6	0			
7	0			
8	0			
9	0			

2.16 - VBA : les boucles -> For Each...Next

```
Sub FORNEXT ()
    Dim max As Integer
    Dim ligne As Integer
    Dim msg As String

    max = 9
    For ligne = 1 To max
        msg = "A" + CStr(ligne)
        Worksheets("NomFeuille").Range(msg).Value = 0
    Next ligne
End Sub
```

	A	B	C	D	E
1	11	Remplir A	Effacer A	Somme	
2	22				
3	33				
4	44				
5	55				
6	66				
7	77				
8	88				
9	99				
10	495				
11					

2.17 - VBA : If...Then...Else

Option Explicit

```
Sub CouleurBonus (rg As Range)
    If rg.value = 0 Then
        rg.Interior.ColorIndex = 3
    Else
        If rg.value > 0 And rg.value <= 100 Then
            rg.Interior.ColorIndex = 45
        Else
            If rg.value >= 100 And rg.value <= 135 Then
                rg.Interior.ColorIndex = 8
            Else
                rg.Interior.ColorIndex = 4
            End If
        End If
    End If
End Sub
```

2.18 - VBA : Select Case

```
Sub Bonus()  
    Dim Bonus As Range  
    Dim performance As Range  
    Dim salary As Range  
    Dim ligne As Integer  
    Dim val As Integer  
  
    For ligne = 2 To 11  
        Set salary = Range("B" + CStr(ligne))  
        Set performance = Range("C" + CStr(ligne))  
        Set Bonus = Range("D" + CStr(ligne))  
        val = performance.value  
  
        Select Case val  
            Case 1  
                Bonus.value = salary.value * 0.1  
            Case 2, 3  
                Bonus.value = salary.value * 0.09  
            Case 4 To 6  
                Bonus.value = salary.value * 0.07  
            Case Is >= 7  
                Bonus.value = 100  
            Case Else  
                Bonus.value = 0  
        End Select  
  
        CouleurBonus Bonus  
    Next ligne
```

2.18 - VBA : Select Case

A	B	C	D	E	F	G
	salaire	performance	bonus		Bonus	
	1500	0				
	1500	1				
	1500	2				
	1500	3				
	1500	4				
	1500	5				
	1500	6				
	1500	7				
	1500	8				
	1500	9				

	A	B	C	D	E	F	
1		salaire	performance	bonus		Bonus	
2		1500	0	0			
3		1500	1	150			
4		1500	2	135			
5		1500	3	135			
6		1500	4	105			
7		1500	5	105			
8		1500	6	105			
9		1500	7	100			
10		1500	8	100			
11		1500	9	100			
12							

2.19 - VBA : les types de procédure

Visual Basic utilise plusieurs types de procédures :

- Les procédures Sub exécutent des actions mais ne retournent aucune valeur au code appelant.
- Les procédures de gestion d'événements sont des procédures Sub qui s'exécutent en réponse à un événement déclenché par une action de l'utilisateur ou une occurrence dans un programme.
- Les procédures Function retournent une valeur au code appelant. Elles peuvent effectuer d'autres actions avant de retourner une valeur.
- Les procédures Property renvoient et attribuent des valeurs de propriétés à des objets ou modules.
- Les procédures Operator définissent le comportement d'un opérateur standard lorsqu'un ou les deux opérandes représentent une structure ou classe qui vient d'être définie.
- Les procédures génériques dans Visual Basic définissent un ou plusieurs paramètres de type en plus de leurs paramètres normaux, et le code appelant peut donc passer des types de données spécifiques chaque fois qu'il effectue un appel.

2.20 - VBA : Procédure Sub

Une procédure est une forme d'instruction composée dont l'exécution dépend de paramètres mais qui ne retourne rien (contrairement `a une fonction) :

Sub Identificateur
(DeclarationDeParametres)
Instructions
Exit Sub
End Sub

2.21 - VBA : Procédure Function

Une fonction est une forme d'instruction composée dont l'exécution dépends de paramètres (Les fonctions sont des procédures qui renvoient une valeur)

*Function Identificateur (Declaration Parametres)
As TypeDeRetour
Instructions
Affectation d'Identificateur
Exit Function
End Function*

2.22 - VBA : Exemple Function & Sub

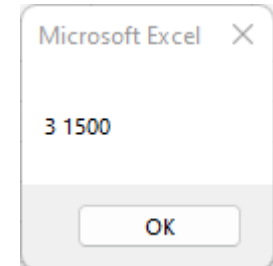
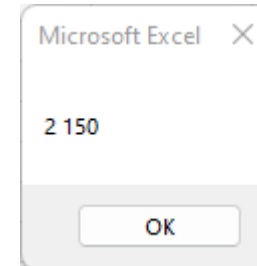
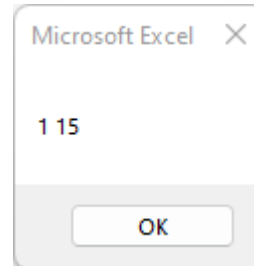
Option Explicit

```
Private Sub SomProcedure_1(ByVal x As Integer, ByVal y As Integer, ByRef som As Integer)
    som = x + y
End Sub
```

```
Private Sub SomProcedure_2(ByVal x As Integer, ByVal y As Integer, som As Integer)
    som = x + y
End Sub
```

```
Private Function SomFonction_1(ByVal x As Integer, ByVal y As Integer) As Integer
    SomFonction_1 = x + y
End Function
```

```
Public Sub Exemple()
    Dim Somme As Integer, A As Integer, B As Integer
    A = 10
    B = 5
    Somme = 0
    SomProcedure_1 A, B, Somme
    MsgBox "1 " & Somme
    A = 100
    B = 50
    Somme = 0
    SomProcedure_2 A, B, Somme
    MsgBox "2 " & Somme
    A = 1000
    B = 500
    Somme = SomFonction_1(A, B)
    MsgBox "3 " & Somme
End Sub
```



2.23 - VBA : Passage d'argument des procédure

ByRef permet de passer à une procédure l'adresse d'un argument (pas de copie) => La valeur réelle de cette argument peut être modifiée par la procédure à laquelle elle a été passée

(Par défaut, les arguments sont passés par référence)

ByVal permet de passer à une procédure la valeur d'un argument plutôt que son adresse. La procédure accède à une copie de la variable => La valeur réelle de cette argument n'est donc pas modifiée par la procédure à laquelle elle est passée.

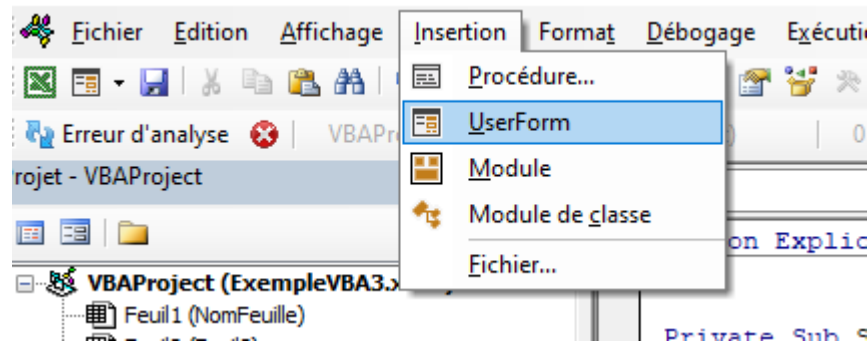
Remarque : L'utilisation de ByVal implique un temps de calcul plus long et nécessite un espace mémoire plus important.

<https://docs.microsoft.com/fr-fr/dotnet/visual-basic/programming-guide/language-features/procedures/procedure-parameters-and-arguments>

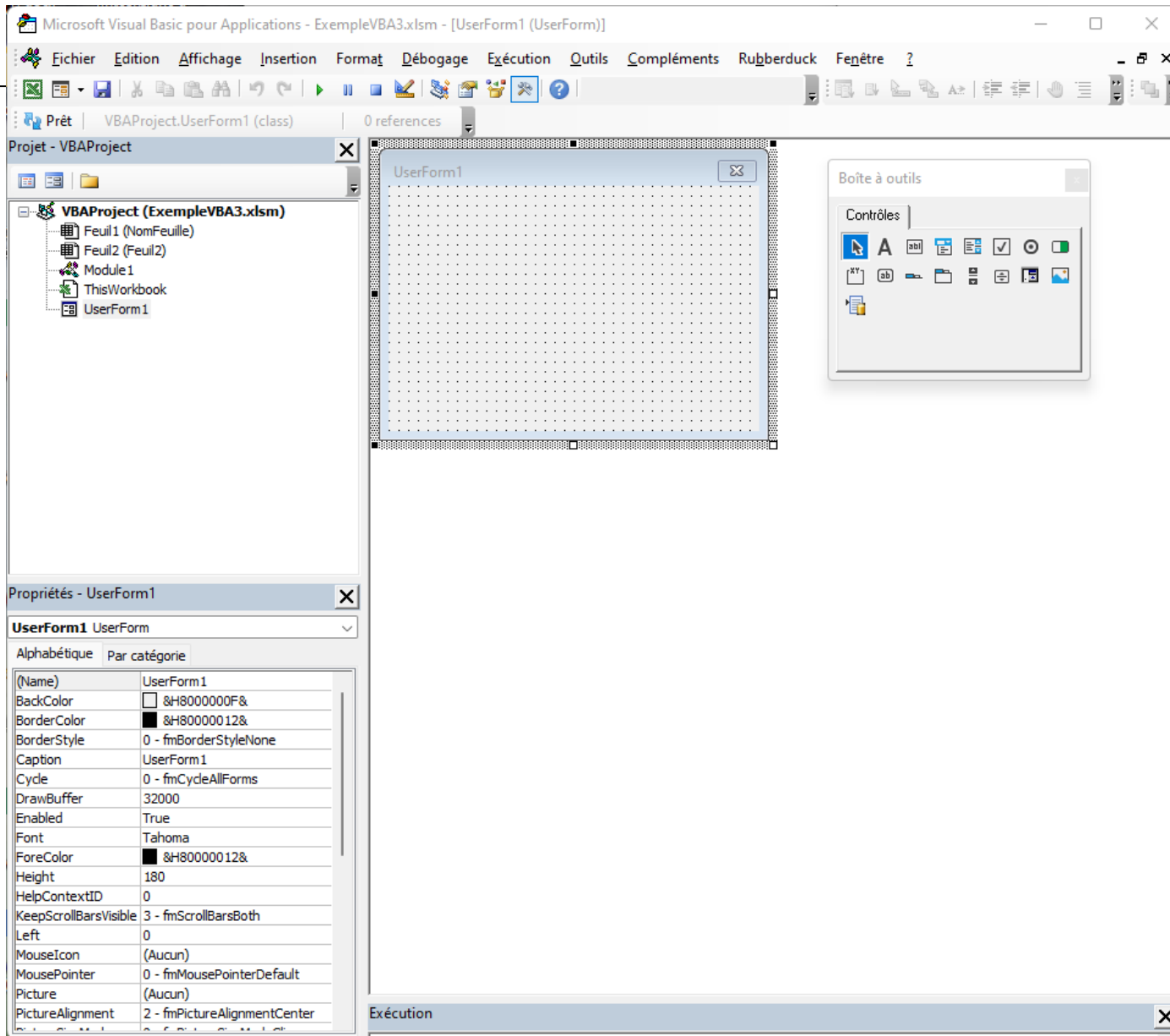
2.24 - VBA : UserForm

Les UserForm sont des boîtes de dialogues personnalisées, ou des fenêtres qui font partie de l'interface utilisateur d'une application.

Sous VBE, les UserForm sont créés par le menu "Insertion -> UserForm".



2.24 - VBA : UserForme

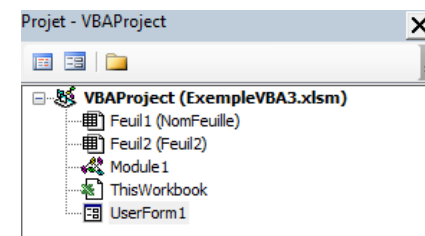
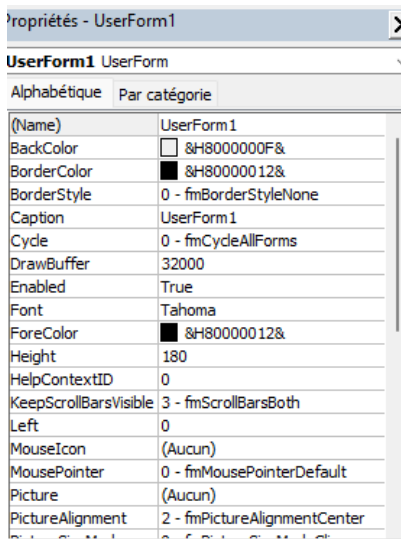
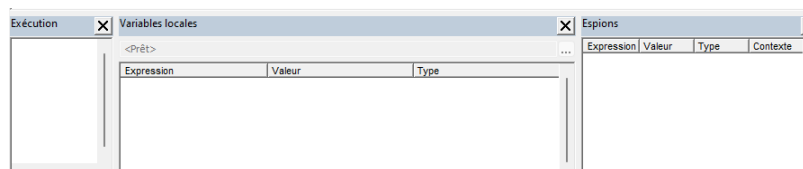
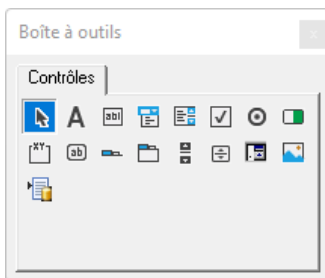
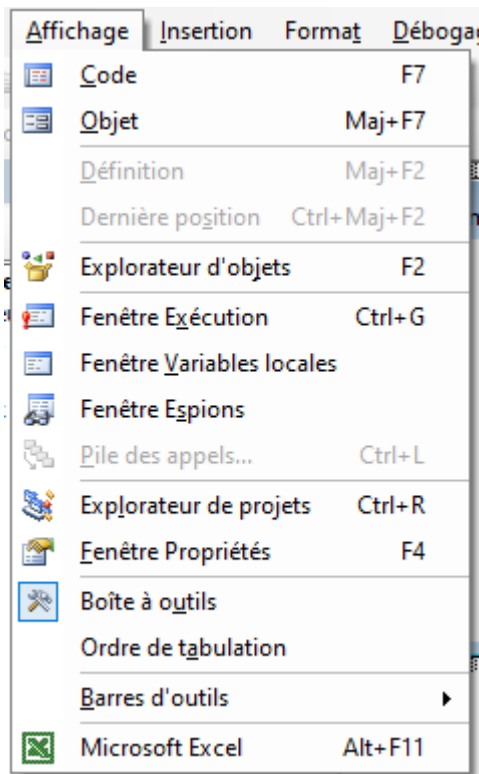


The screenshot displays the VBA editor interface for a project named 'ExempleVBA3.xlsm'. The main window shows the 'UserForm1' design, which is a blank grid. The 'Boîte à outils' (Toolbox) on the right contains various controls for building the user form. The 'Propriétés - UserForm1' (Properties) window at the bottom left lists the properties of the selected user form.

Propriétés - UserForm1

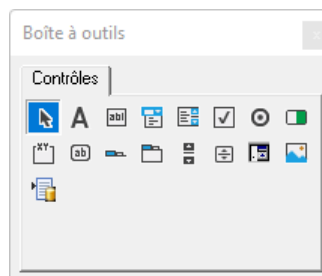
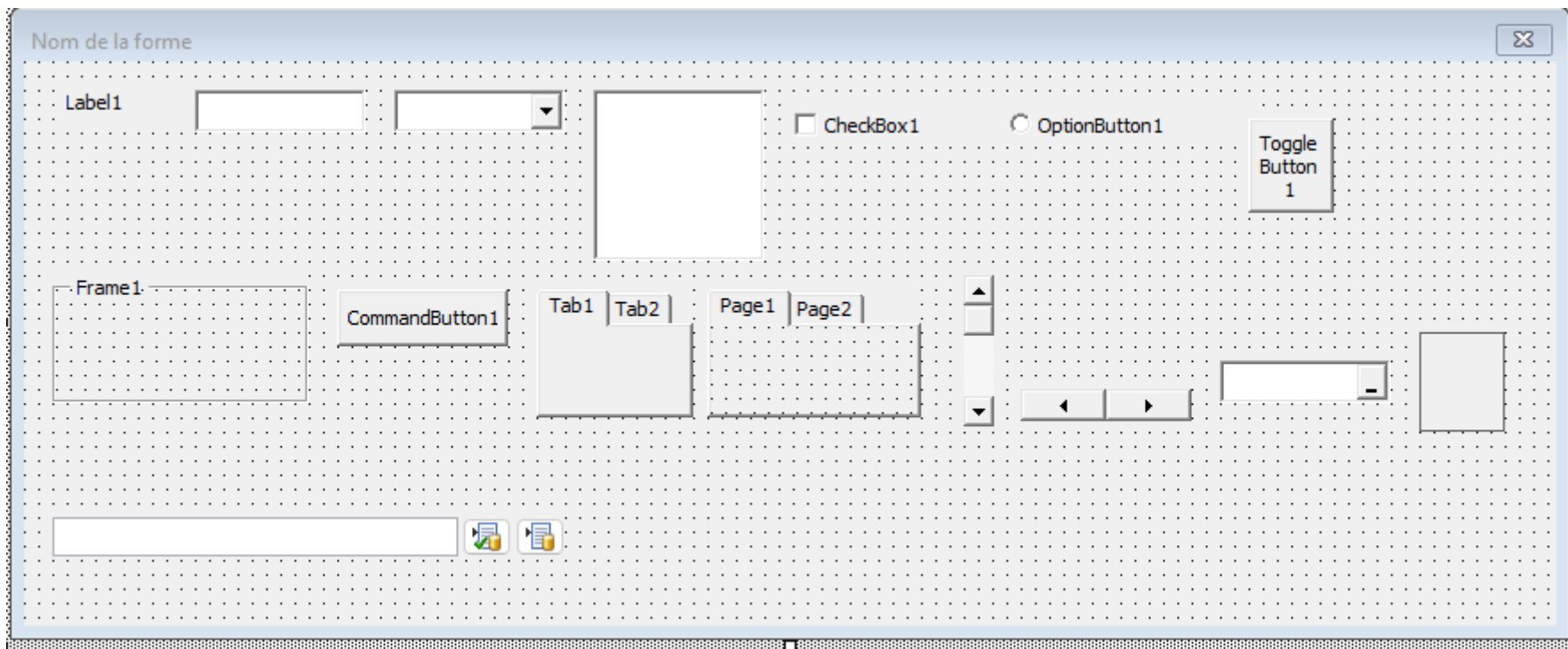
Propriété	Valeur
(Name)	UserForm1
BackColor	&H8000000F&
BorderColor	&H80000012&
BorderStyle	0 - fmBorderStyleNone
Caption	UserForm1
Cycle	0 - fmCycleAllForms
DrawBuffer	32000
Enabled	True
Font	Tahoma
ForeColor	&H80000012&
Height	180
HelpContextID	0
KeepScrollBarsVisible	3 - fmScrollBarsBoth
Left	0
MouseIcon	(Aucun)
MousePointer	0 - fmMousePointerDefault
Picture	(Aucun)
PictureAlignment	2 - fmPictureAlignmentCenter

2.24 - VBA : UserForm



<https://silkyroad.developpez.com/VBA/UserForm>

2.24 - VBA : UserForm



2.24 - VBA : UserForme -> Exemple d'application

Contact

Nom

Prenom

Adresse

Valider Annuler

UserForm1 UserForm	
Alphabétique	Par catégorie
(Name)	UserForm1
BackColor	<input type="checkbox"/> &H8000000F&
BorderColor	<input checked="" type="checkbox"/> &H80000012&
BorderStyle	0 - fmBorderStyleNone
Caption	Contact
Cycle	2 - fmCycleCurrentForm
DrawBuffer	32000
Enabled	True
Font	Tahoma
ForeColor	<input checked="" type="checkbox"/> &H80000012&
Height	291,75
HelpContextID	0
KeepScrollBarsVisible	3 - fmScrollBarsBoth
Left	0
MouseIcon	(Aucun)
MousePointer	0 - fmMousePointerDefault
Picture	(Aucun)
PictureAlignment	2 - fmPictureAlignmentCenter
PictureSizeMode	0 - fmPictureSizeModeClip
PictureTiling	False
RightToLeft	False
ScrollBars	0 - fmScrollBarsNone
ScrollHeight	0
ScrollLeft	0
ScrollTop	0
ScrollWidth	0
ShowModal	True
SpecialEffect	0 - fmSpecialEffectFlat
StartupPosition	1 - CenterOwner
Tag	
Top	0
WhatsThisButton	False
WhatsThisHelp	False
Width	396
Zoom	100

2.24 - VBA : UserForme -> Exemple d'application

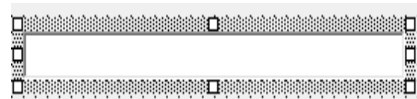


Propriétés - Label1

Label1 Label

Alphabétique Par catégorie

(Name)	Label1
Accelerator	
AutoSize	False
BackColor	&H8000000F&
BackStyle	1 - fmBackStyleOpaque
BorderColor	&H80000006&
BorderStyle	0 - fmBorderStyleNone
Caption	Nom
ControlTipText	
Enabled	True
Font	Tahoma
ForeColor	&H80000012&
Height	18
HelpContextID	0
Left	18
MouseIcon	(Aucun)
MousePointer	0 - fmMousePointerDefault
Picture	(Aucun)
PicturePosition	7 - fmPicturePositionAboveCenter
SpecialEffect	0 - fmSpecialEffectFlat
TabIndex	0
TabStop	False
Tag	
TextAlign	1 - fmTextAlignLeft
Top	36
Visible	True
Width	72
WordWrap	True



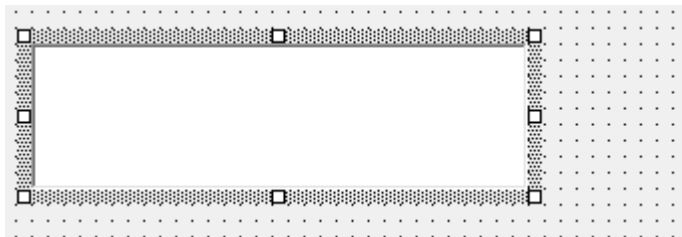
Propriétés - TextBox1

TextBox1 TextBox

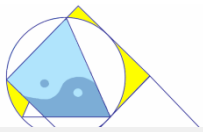
Alphabétique Par catégorie

(Name)	TextBox1
AutoSize	False
AutoTab	False
AutoWordSelect	True
BackColor	&H80000005&
BackStyle	1 - fmBackStyleOpaque
BorderColor	&H80000006&
BorderStyle	0 - fmBorderStyleNone
ControlSource	
ControlTipText	
DragBehavior	0 - fmDragBehaviorDisabled
Enabled	True
EnterFieldBehavior	0 - fmEnterFieldBehaviorSelectAll
EnterKeyBehavior	False
Font	Tahoma
ForeColor	&H80000008&
Height	18
HelpContextID	0
HideSelection	True
IMEMode	0 - fmIMEModeNoControl
IntegralHeight	True
Left	102
Locked	False
MaxLength	0
MouseIcon	(Aucun)
MousePointer	0 - fmMousePointerDefault
MultiLine	False
PasswordChar	
ScrollBars	0 - fmScrollBarsNone
SelectionMargin	True
SpecialEffect	2 - fmSpecialEffectSunken
TabIndex	3
TabKeyBehavior	False
TabStop	True
Tag	
Text	
TextAlign	1 - fmTextAlignLeft
Top	36
Value	
Visible	True
Width	192
WordWrap	True

2.24 - VBA : UserForme -> Exemple d'application



Propriétés - TextBox3	
TextBox3 TextBox	
Alphabétique	Par catégorie
(Name)	TextBox3
AutoSize	False
AutoTab	False
AutoWordSelect	True
BackColor	<input type="checkbox"/> &H80000005&
BackStyle	1 - fmBackStyleOpaque
BorderColor	<input checked="" type="checkbox"/> &H80000006&
BorderStyle	0 - fmBorderStyleNone
ControlSource	
ControlTipText	
DragBehavior	0 - fmDragBehaviorDisabled
Enabled	True
EnterFieldBehavior	0 - fmEnterFieldBehaviorSelectAll
EnterKeyBehavior	False
Font	Tahoma
ForeColor	<input checked="" type="checkbox"/> &H80000008&
Height	54,75
HelpContextID	0
HideSelection	True
IMEMode	0 - fmIMEModeNoControl
IntegralHeight	True
Left	102
Locked	False
MaxLength	0
MouseIcon	(Aucun)
MousePointer	0 - fmMousePointerDefault
MultiLine	True
PasswordChar	
ScrollBars	3 - fmScrollBarsBoth
SelectionMargin	True
SpecialEffect	2 - fmSpecialEffectSunken
TabIndex	5
TabKeyBehavior	False
TabStop	True
Tag	
Text	
TextAlign	1 - fmTextAlignLeft
Top	102
Value	
Visible	True
Width	186
WordWrap	False



2.24 - VBA : UserForme -> Exemple d'application

Enregistrement automatique ☐ ExempleVBA4.xlsm

Rechercher (Alt+Q)

david palermo

Fichier Accueil Insertion Dessin Mise en page Formules Données Révision Affichage **Développeur** Aide Format de forme

Visual Basic Macros Utiliser les références relatives Sécurité des macros

Compléments Excel COM Insérer Mode Création Propriétés Visualiser le code Exécuter la boîte de dialogue

Source Propriétés du mappage Kits d'extension Actualiser les données Importer Exporter XML

abc Orthographe

Bouton 1

Nom Prenom Adresse

Nouveau contact

Microsoft Visual Basic pour Applications - ExempleVBA4.xlsm [arrêt] - [Module1 (Code)]

Fichier Edition Affichage Insertion Format Débogage Exécution Outils Compléments Rubberduck Fenêtre ?

Prêt VBAProject.Module1 (module) 0 references

Projet - VBAProject

VBAPProject (ExempleVBA4.xlsm)

Feuil1 (Feuil1)

Module1

ThisWorkbook

UserForm1

Propriétés - Module1

Module1 Module

Alphabétique Par catégorie

(Name) Module1

(Général) RentrerContact

Option Explicit

Sub RentrerContact

Us

End Sub

Dans ce cas l'affichage de la forme sera Modal
Sinon faire :

- UserForm1.Show 0
- UserForm1.Show False
- UserForm1.Show vbModeless

Exécution Variables locales Espions

<Arrêt>

Expression Valeur Type

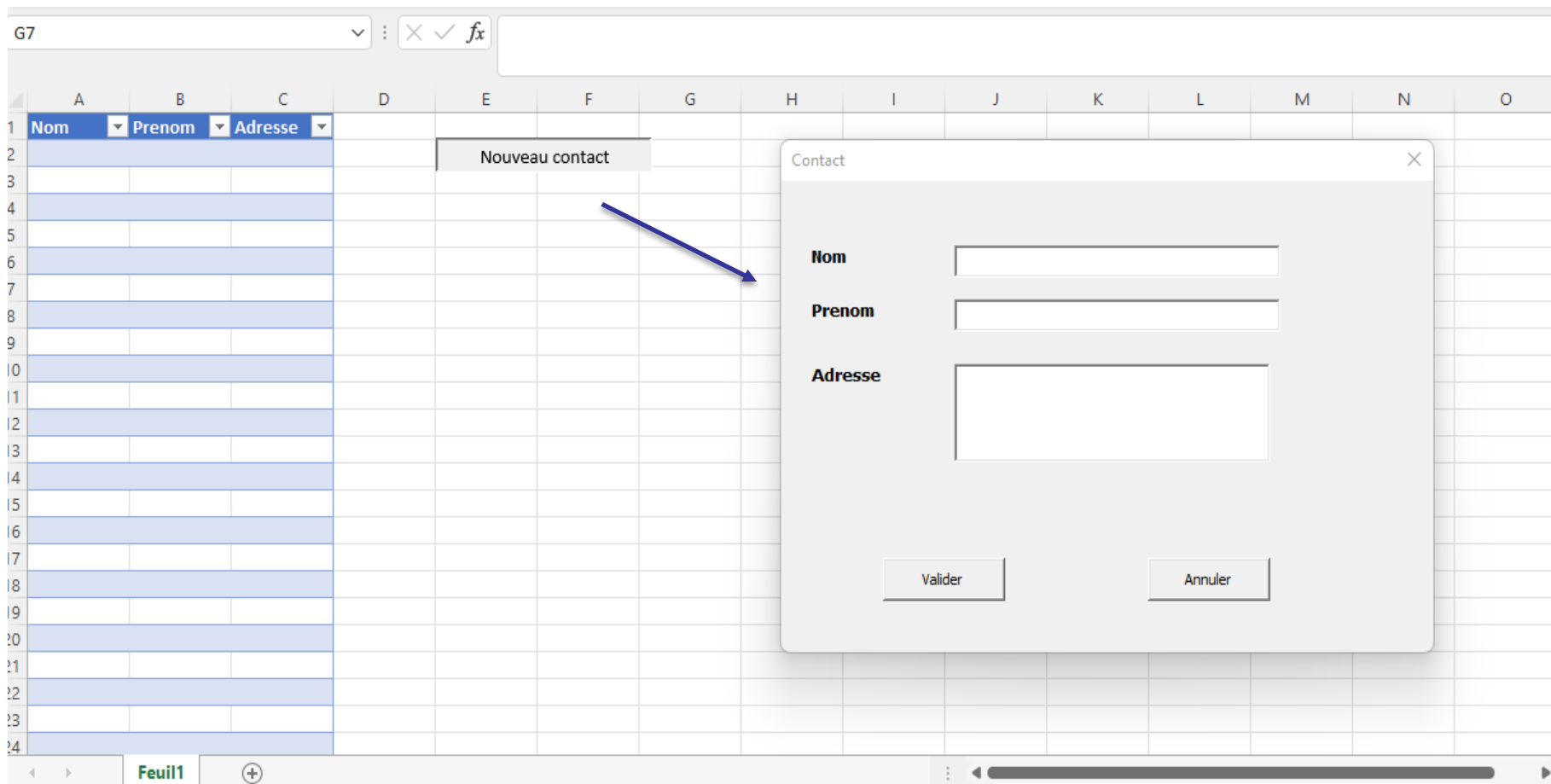
Feuil1

Accessibilité : vérification terminée

Version 1,1 - 08/2023

Copyright : Yantra Technologies 2004-2023

2.24 - VBA : UserForme -> Exemple d'application



The screenshot shows an Excel spreadsheet with a table structure. The first row (row 1) contains headers: "Nom", "Prenom", and "Adresse". The subsequent rows (rows 2 to 24) are empty. A blue arrow points from the "Nouveau contact" button in the spreadsheet to the "Contact" UserForm. The UserForm has three input fields labeled "Nom", "Prenom", and "Adresse", and two buttons at the bottom: "Valider" and "Annuler".

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	Nom	Prenom	Adresse												
2															
3															
4															
5															
6															
7															
8															
9															
10															
11															
12															
13															
14															
15															
16															
17															
18															
19															
20															
21															
22															
23															
24															

2.24 - VBA : UserForme -> Exemple d'application

Option Explicit

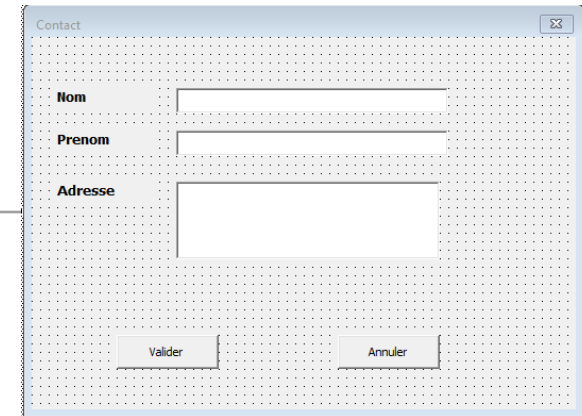
'permet de définir les propriétés des objets et les valeurs par défaut des variables,
'lors du lancement du Userform.

```
Private Sub UserForm_Initialize()
    MsgBox "Initialiser"
    Me.TextBox1 = ""
    Me.TextBox2 = ""
    Me.TextBox3 = ""
End Sub
```

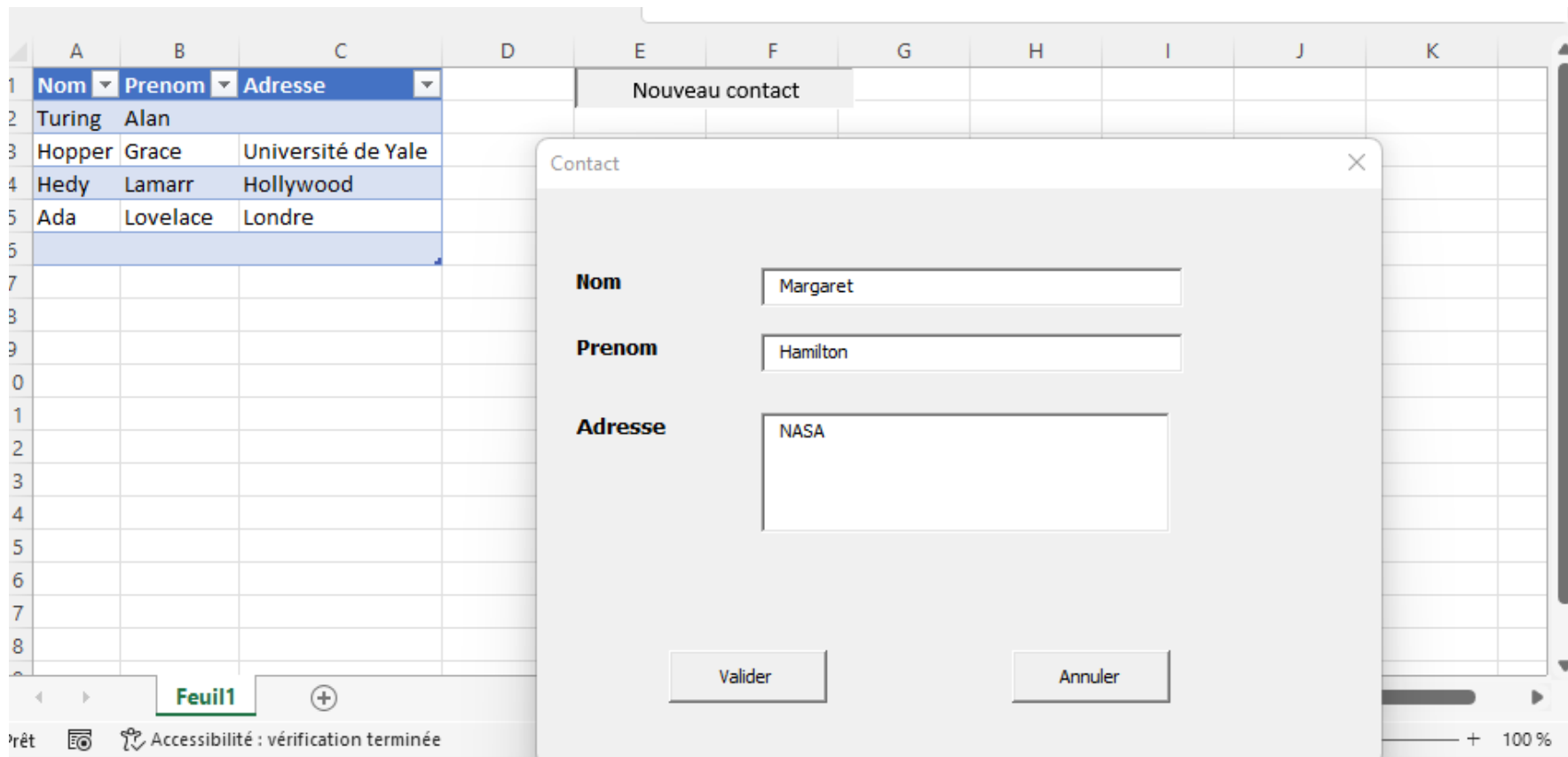
```
Private Sub CommandButton1_Click()
    MsgBox "Valider"
    'https://www.automateexcel.com/fr/vba/inserer-ligne-ou-colonne/
    Worksheets("Feuill").Range("A2:A2").EntireRow.Insert
    Worksheets("Feuill").Range("A2").Value = Me.TextBox1.Value
    Worksheets("Feuill").Range("B2").Value = Me.TextBox2.Value
    Worksheets("Feuill").Range("C2").Value = Me.TextBox3.Value
    CommandButton2_Click
End Sub
```

```
Private Sub CommandButton2_Click()
    MsgBox "Annuler"
    Me.TextBox1 = ""
    Me.TextBox2 = ""
    Me.TextBox3 = ""
    'https://www.automateexcel.com/vba/show-hide-sheets/
    UserForm1.Hide ' : masque la UserForme sans la decharger de la memoire et les valeurs associés

    'Unload UserForm1 : ferme la UserForme et la decharge de la memoire
    'Unload Me ' : ferme la UserForme courante et la decharge de la memoire
End Sub
```



2.24 - VBA : UserForme -> Exemple d'application



The screenshot shows an Excel spreadsheet with a VBA UserForm titled "Contact" overlaid. The spreadsheet has columns A, B, and C, and rows 1 through 8. The data in the spreadsheet is as follows:

Nom	Prenom	Adresse
Turing	Alan	
Hopper	Grace	Université de Yale
Hedy	Lamarr	Hollywood
Ada	Lovelace	Londre

The UserForm "Contact" has three text boxes for "Nom", "Prenom", and "Adresse". The "Nom" box contains "Margaret", the "Prenom" box contains "Hamilton", and the "Adresse" box contains "NASA". There are two buttons at the bottom: "Valider" and "Annuler".

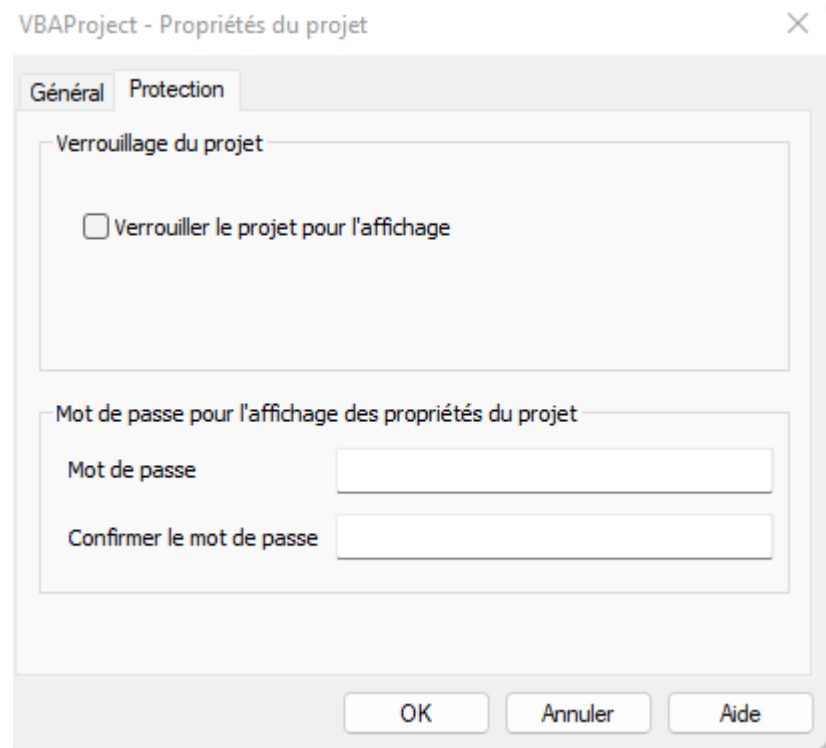
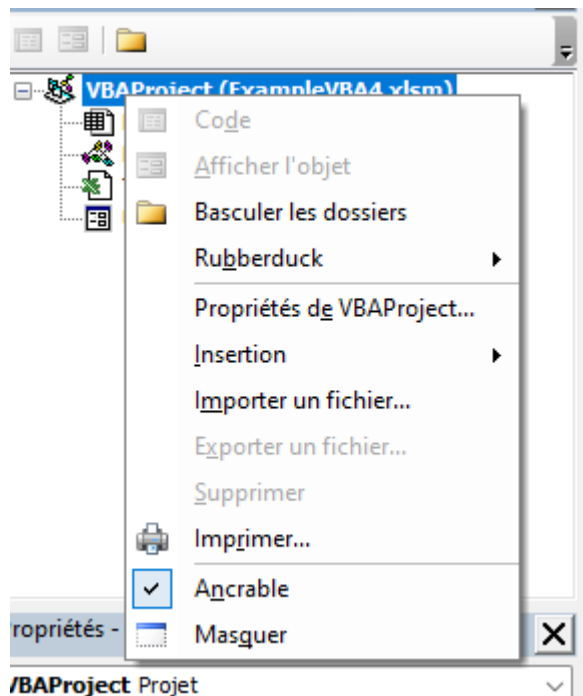
3 - VBA & Excel Sécurisation : Sommaire

3.1 Protéger feuille Excel

3.2 Protéger le classeur

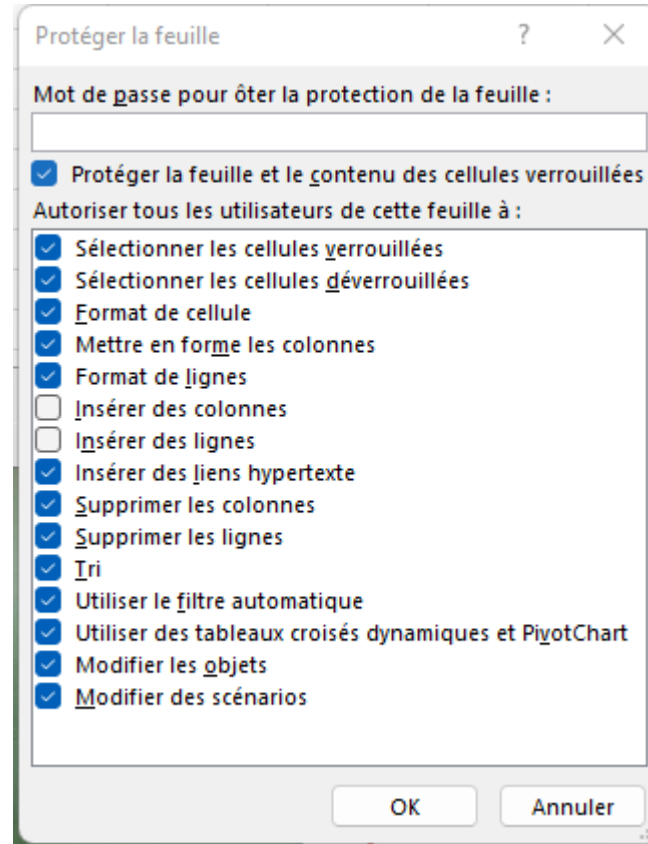
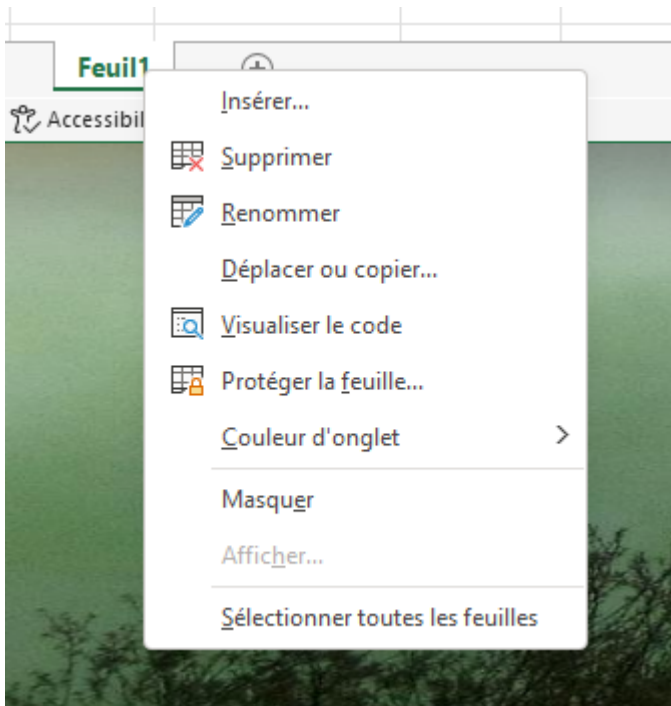
3 - VBA & Excel : Sécurisation

- Protéger le projet VBA : ajouter mots de passe



3.1 - VBA & Excel : Protéger feuille Excel

- Protéger feuille Excel : choisir les autorisation et ajouter mots de passe



<https://formations-excel.blogspot.com/2015/11/comment-proteger-une-feuille-classeur.html>

3.1 - VBA & Excel : Protéger feuille Excel

- **Déprotéger une Feuille de Calcul Excel Avec un Mot de Passe**

`Worksheets("Feuil1").Unprotect "MotDePasse "`

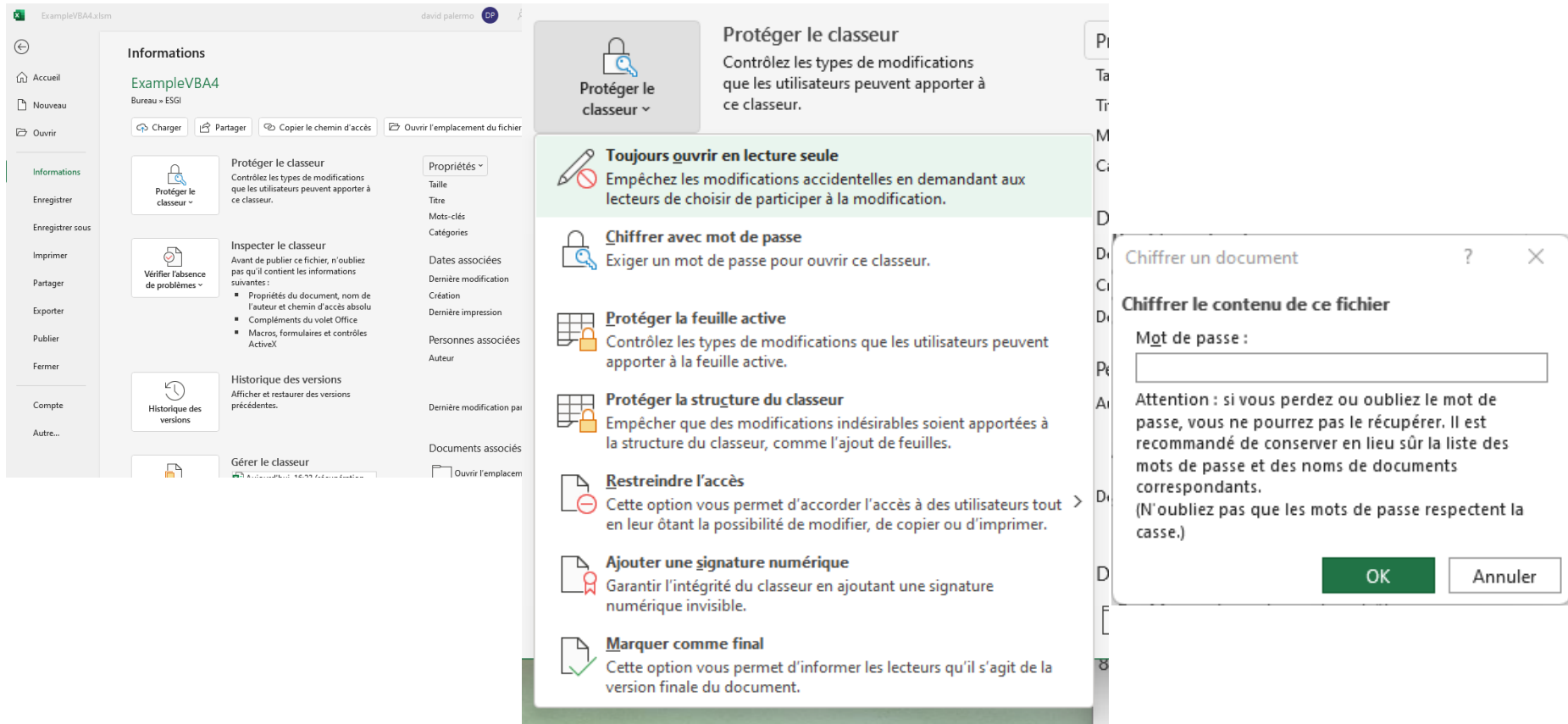
- **Protection d'une Feuille de Calcul – Protection par Mot de Passe**

`Worksheets("Feuil1").Protect "MotDePasse"`

<https://www.automateexcel.com/fr/vba/proteger-deproteger-feuilles-calcul>

3.2 - VBA & Excel : Protéger le classeur

Fichier-> information -> protégé le classeur



Informations

ExampleVBA4
Bureau > ESGI

Charger Partager Copier le chemin d'accès Ouvrir l'emplacement du fichier

Protéger le classeur
Contrôlez les types de modifications que les utilisateurs peuvent apporter à ce classeur.

Inspecter le classeur
Avant de publier ce fichier, n'oubliez pas qu'il contient les informations suivantes :

- Propriétés du document, nom de l'auteur et chemin d'accès absolu
- Compléments du volet Office
- Macros, formulaires et contrôles ActiveX

Historique des versions
Afficher et restaurer des versions précédentes.

Gérer le classeur
4/27 11/11/2023 14:00:00 14:00:00

Protéger le classeur
Contrôlez les types de modifications que les utilisateurs peuvent apporter à ce classeur.

Toujours ouvrir en lecture seule
Empêchez les modifications accidentelles en demandant aux lecteurs de choisir de participer à la modification.

Chiffrer avec mot de passe
Exiger un mot de passe pour ouvrir ce classeur.

Protéger la feuille active
Contrôlez les types de modifications que les utilisateurs peuvent apporter à la feuille active.

Protéger la structure du classeur
Empêcher que des modifications indésirables soient apportées à la structure du classeur, comme l'ajout de feuilles.

Restreindre l'accès
Cette option vous permet d'accorder l'accès à des utilisateurs tout en leur ôtant la possibilité de modifier, de copier ou d'imprimer.

Ajouter une signature numérique
Garantir l'intégrité du classeur en ajoutant une signature numérique invisible.

Marquer comme final
Cette option vous permet d'informer les lecteurs qu'il s'agit de la version finale du document.

Chiffrer un document

Chiffrer le contenu de ce fichier

Mot de passe :

Attention : si vous perdez ou oubliez le mot de passe, vous ne pourrez pas le récupérer. Il est recommandé de conserver en lieu sûr la liste des mots de passe et des noms de documents correspondants. (N'oubliez pas que les mots de passe respectent la casse.)

OK Annuler

3.2 - VBA & Excel : Protéger le classeur

- **Déprotéger un classeur Excel Avec un Mot de Passe**

```
Workbooks("Classeur1").Unprotect Password:="Mot de Passe"  
Workbooks("Classeur1").Unprotect Mot de Passe"
```

- **Protection un classeur Excel – Protection par Mot de Passe**

```
Workbooks("Classeur1").Protect "MotDePasse"
```

<https://www.automateexcel.com/fr/vba/protection-classeurs-mot-de-passe/>

4 -Bibliographies

- <https://www.excel-pratique.com/fr/vba>
- https://tutoriel-vba.espaceweb.usherbrooke.ca/vba_accueil.html

