



# TP 11 – Initiation aux Macros VBA

SKL3011

Niveau : 3<sup>ème</sup> Année

Année : 2024/2025

Filière : Toutes les filières

Session : Automne 2024

## Objectif du TP

Le **VBA** (Visual Basic for Applications) est un langage de programmation intégré à Excel. Il permet d'automatiser les tâches répétitives et de créer des applications interactives.

Les **macros**, écrites en VBA, sont des **scripts** qui enregistrent et exécutent ces tâches, rendant Excel plus puissant et adapté aux besoins spécifiques des utilisateurs.

Dans ce TP, vous utiliserez la programmation VBA pour :

- Créer et exécuter des macros simples.
- Interagir avec l'utilisateur via MsgBox et InputBox.
- Manipuler des données et cellules Excel.
- Appliquer des conditions pour automatiser des tâches.
- Gérer les erreurs et les entrées invalides.

## Macro 1 : Création d'une Macro pour Saluer l'Utilisateur

Dans cet exercice, vous allez apprendre à utiliser les fonctions **MsgBox** et **InputBox** pour interagir avec l'utilisateur, et à manipuler des variables.

❖ Ce que votre macro doit faire :

Cette macro demandera votre nom à l'aide d'une boîte de dialogue, puis affichera un message de salutation personnalisé sous la forme : "**Bonjour [nom saisi]**".

❖ Étapes à suivre :

1- Créer un bouton pour exécuter la macro :

- Ouvrez un classeur Excel.
- Allez dans l'onglet Développeur et insérez un bouton.
- Nommez le bouton **Saluer**.
- Pour l'instant, laissez le bouton sans action. Nous y affecterons notre macro plus tard.

2- Afficher un message de base :

- Ouvrez l'éditeur VBA puis créez un nouveau module.
- Créez une procédure **Sub** appelée **DireBonjour**.
- Ajoutez dans cette procédure une boîte de message contenant le texte : "**Bonjour le monde !**" en utilisant la fonction **MsgBox**.
- Exécutez votre macro manuellement pour tester le résultat.

### 3- Modifier le style de la MsgBox :

- Ajoutez un deuxième argument à la fonction `MsgBox` pour changer le style de la fenêtre.
- Affichez le message avec le style `vbExclamation`, puis changez-le en `vbInformation`.

### 4- Utiliser une variable pour personnaliser le message :

- Créez une variable que nous appellerons `nom`.
- Affectez à cette variable votre propre nom (*par exemple* : `nom = "Anir"`).
- Modifiez votre code pour afficher : "`Bonjour [Nom]`", en utilisant la variable `nom` dans la fonction `MsgBox`.

### 5- Demander le nom à l'utilisateur :

- Remplacez l'affectation fixe de la variable `nom` par une saisie utilisateur, à l'aide de la fonction `InputBox`.
- Le message de la boîte de saisie doit être : "`Quel est votre nom ?`".
- Testez votre macro et vérifiez que le message affiché s'adapte au nom saisi.

### 6- Déclarer explicitement la variable :

- Ajoutez une ligne au début de votre macro pour déclarer la variable `nom` en tant que chaîne de caractères (`String`).
- Remarquez que la déclaration explicite des variables permet d'éviter les erreurs liées à des types inattendus ou à des fautes de frappe dans les noms des variables.

### 7- Utiliser Option Explicit :

- Ajoutez `Option Explicit` en haut du module, au-dessus de votre macro.
- Mettez en commentaire la ligne où vous déclarez explicitement la variable `nom` (*en ajoutant une apostrophe au début de la ligne*), puis testez votre code.
- Corrigez le code en rétablissant la déclaration de la variable.

### 8- Affecter la macro au bouton :

- Retournez dans votre classeur Excel.
- Affectez la macro `DireBonjour` au bouton créé à l'étape 1.
- Cliquez sur le bouton pour tester le fonctionnement complet de votre macro.

### 9- Ajouter l'âge de l'utilisateur :

- Modifiez la macro pour qu'elle demande également l'âge de l'utilisateur, puis affiche le message suivant : "`Bonjour [nom saisi], vous avez [âge saisi] ans.`"
- Saisissez un texte au lieu d'un nombre dans la boîte de dialogue pour l'âge, et observez l'erreur que cela génère.

## Macro 2 : Création d'une Macro pour Additionner deux Valeurs

---

Dans cet exercice, vous allez apprendre à manipuler des variables, effectuer une opération arithmétique simple, puis interagir avec des cellules Excel.

### ❖ Ce que votre macro doit faire :

Cette macro sera capable de lire les valeurs des cellules **B3** et **B4**, de calculer leur somme, et d'afficher le résultat directement dans la cellule **B5**.

## Volet 1 : Affectation et Affichage du résultat basique

1- Créer une procédure **Sub** et insérez un bouton :

- Ouvrez l'éditeur VBA.
- Créez une procédure **Sub** appelée **Addition**.
- Revenez à la feuille de calcul, insérez un bouton, et nommez-le **Add**.
- Affectez la macro **Addition** au bouton Add.

2- Déclarer trois variables :

- Dans votre macro, déclarez trois variables nommées **a**, **b**, et **res**, en leur attribuant le type **Double** (*pour manipuler des valeurs réelles*).

3- Affecter des valeurs aux variables **a** et **b** :

- Assignez une valeur de votre choix à la variable **a** (*par exemple 5*) et une autre à la variable **b** (*par exemple 3*).

4- Calculer la somme :

- Ajoutez une instruction pour calculer la somme de **a** et **b**, et stockez le résultat dans la variable **res**.

5- Afficher le résultat dans un **MsgBox** :

- Insérez une instruction pour afficher la valeur de **res** dans une boîte de message **MsgBox**.
- Testez votre macro en cliquant sur le bouton Add.

## Volet 2 : Manipulation des Cellules Excel

6- Modifier le **MsgBox** pour afficher le résultat dans une cellule :

- Remplacez l'instruction **MsgBox** par une ligne de code qui assigne la valeur de **res** à la cellule **B5** dans votre feuille Excel (pour cela utiliser l'objet **Range**).
- Testez à nouveau votre macro pour vérifier que le résultat apparaît dans la cellule **B5**.

7- Changer l'affectation manuelle des variables **a** et **b** :

- Modifiez votre code pour que la variable **a** reçoive la valeur de la cellule **B3**, et **b** celle de la cellule **B4**.
- Rentrez des valeurs dans les cellules **B3** et **B4**.
- Testez votre macro en vérifiant que le résultat s'affiche dans **B5**.

## Macro 3 : Création d'une Macro pour Vérifier et Classifier une Note

---

Dans cet exercice, vous allez apprendre à utiliser des **conditions** en VBA pour vérifier et traiter des valeurs saisies dans une cellule Excel. Vous allez créer une macro capable de :

- Vérifier une note saisie dans une cellule Excel.
- Appliquer une couleur spécifique à la cellule en fonction de la valeur de la note.
- Gérer les cas où la note est invalide.

❖ Ce que votre macro doit faire :

Cette macro vérifiera la note saisie dans **C3** : si elle est invalide (hors 0-20), un message d'erreur s'affichera et la cellule sera colorée en gris. Pour les notes valides, la cellule sera colorée en vert (si  $note \geq 14$ ), jaune (si  $14 > note \geq 10$ ), ou rouge (si  $note < 10$ ).

### Volet 1 : Vérification Simple des notes

1- Créer une procédure et insérer un bouton :

- Ouvrez l'éditeur VBA et créez une procédure **Sub** appelée **VerifNote**.
- Revenez à la feuille de calcul, insérez un bouton, et nommez-le **Vérifier**.
- Affectez la macro **VerifNote** au bouton **Vérifier**.

2- Déclarer une variable :

- Déclarez une variable appelée **note** de type **Double**.

3- Stockez le contenu de la cellule gardant la note dans la variable :

- Affectez la valeur de la cellule **C3** à la variable **note**.

4- Vérifier si la note est supérieure ou inférieure à 10 :

- Utilisez une condition **If** pour vérifier si la variable **note** est supérieure ou égale à 10.
- Si la **note** est supérieure ou égale à **10**, appliquez la couleur verte à la cellule **C3** (utilisez la propriété **ColorIndex**).
- Sinon, appliquez la couleur rouge.
- Utilisez les indices de couleur suivants :

Numéro de couleur (ColorIndex)	Couleur
3	Rouge
4	Vert
6	Jaune
15	Gris
-4142	Sans remplissage (par défaut)

5- Tester la macro :

- Rentrez une valeur dans la cellule **C3**.
- Exécutez la macro et vérifiez que la cellule change de couleur en fonction de sa valeur.

### Volet 2 : Ajouter une troisième Condition pour les bonnes notes

6- Modifier les conditions :

- Ajoutez une nouvelle condition dans votre code pour gérer les notes entre 10 et 14.
- Les règles de couleurs deviennent les suivantes :
  - Vert pour une **note** supérieure ou égale à **14**.
  - Jaune pour une **note** comprise entre **10** et **14**.
  - Rouge pour une **note** inférieure à **10**.

7- Tester la macro :

- Rentrez différentes valeurs dans **C3** (par exemple 15, 11, et 8).
- Exécutez la macro et vérifiez que les couleurs appliquées correspondent aux règles définies.

### Volet 3 : Gestion des notes Invalides

8- Ajouter une vérification pour les notes invalides :

- Avant de vérifier la valeur de la note, ajoutez une condition pour vérifier tout d'abord si la **note** est inférieure à 0 ou supérieure à 20 (note invalide).
- Si la note est invalide :
  - Affichez un message d'erreur avec **MsgBox**.
  - Colorez la cellule **C3** en gris.
  - Ajoutez une instruction **Exit Sub** pour arrêter l'exécution de la macro.

9- Tester la macro :

- Rentrez une valeur invalide (par exemple -5 ou 25) dans **C3**.
- Exécutez la macro et vérifiez que le message d'erreur s'affiche et que la cellule est colorée en gris.

10- Ajoutez le cas où la cellule reste vide :

- Modifiez la condition établie dans la question 8 pour inclure également le cas où la cellule **C3** est vide.

### Macro 4 : Création d'une Macro pour Classifier un Poids

---

Cette macro a pour objectif de classer automatiquement un animal dans une catégorie de poids (léger, moyen ou lourd) tout en gérant les cas de saisie invalide.

#### Étape 1 : Afficher la catégorie selon le poids de l'animal

En se basant sur la feuille **Classification\_Poids**, écrire un script VBA qui permet de lire le poids de l'animal, puis de le classer dans l'une des trois catégories suivantes en fonction de son poids :

- **Lourd** : Si le poids est supérieur à 200 kg.
- **Moyen** : Si le poids est compris entre 50 kg et 200 kg inclus.
- **Léger** : Si le poids est inférieur à 50 kg.

La catégorie de l'animal sera automatiquement affichée dans la cellule **Catégorie** après avoir saisi le poids dans la cellule **Poids de l'Animal**. Le script se déclenche en cliquant sur un bouton.

#### Étape 2 : Traiter le cas où le poids est invalide

Le script doit également vérifier les cas où le poids saisi est invalide :

- Si aucune valeur n'est saisie dans la cellule.
- Si le poids est inférieur ou égal à 0.
- Si la valeur saisie n'est pas un nombre.

Dans ces cas, un message d'erreur doit être affiché, et le contenu de la cellule du poids doit être effacé.