

# Cours Programmation VBA

Excel + Access Avancés

Samir DELIMI

## Rappel de cours et TD2+3

### Les principes de base de la programmation

#### A. Le langage VBA, la programmation orientée « objet »

##### 1. La programmation pour dépasser les limites de l'enregistrement des macro-commandes

La création de **macro-commandes** ou macros avec l'enregistreur présente des limites :

- Elle ne permet pas de créer des boîtes de dialogue qui guident l'utilisateur et donc de créer une réelle interaction ;
- Elle ne permet pas d'entrer des données en cours d'exécution ou d'effectuer des actions en fonction de besoins spécifiques, d'événements (ex. : ajout ou suppression de lignes) ;
- Elle ne peut être utilisée pour créer des **fonctions** personnalisées.

Il peut donc s'avérer nécessaire d'écrire un programme dans un langage adapté au tableur.

#### Définition

Un **programme** est un ensemble d'opérations écrites dans un langage de programmation sous forme de d'instructions.

##### 2. Le langage de programmation d'Excel : Visual Basic Application (VBA)

#### Définition

Le VBA (Visual Basic for Applications) est le langage développé pour les progiciels Microsoft. Il ne peut s'exécuter sans une application hôte (ex. : Excel, Access).

VBA est un langage de programmation structuré orienté « objet » :

- Il est formé d'une suite de phrases appelées instructions, chacune décrit une tâche.
- Un ensemble d'instructions regroupées est une procédure, le programme peut en contenir plusieurs.
- Les instructions comprennent divers éléments, notamment des objets qui seront manipulés par celles-ci.

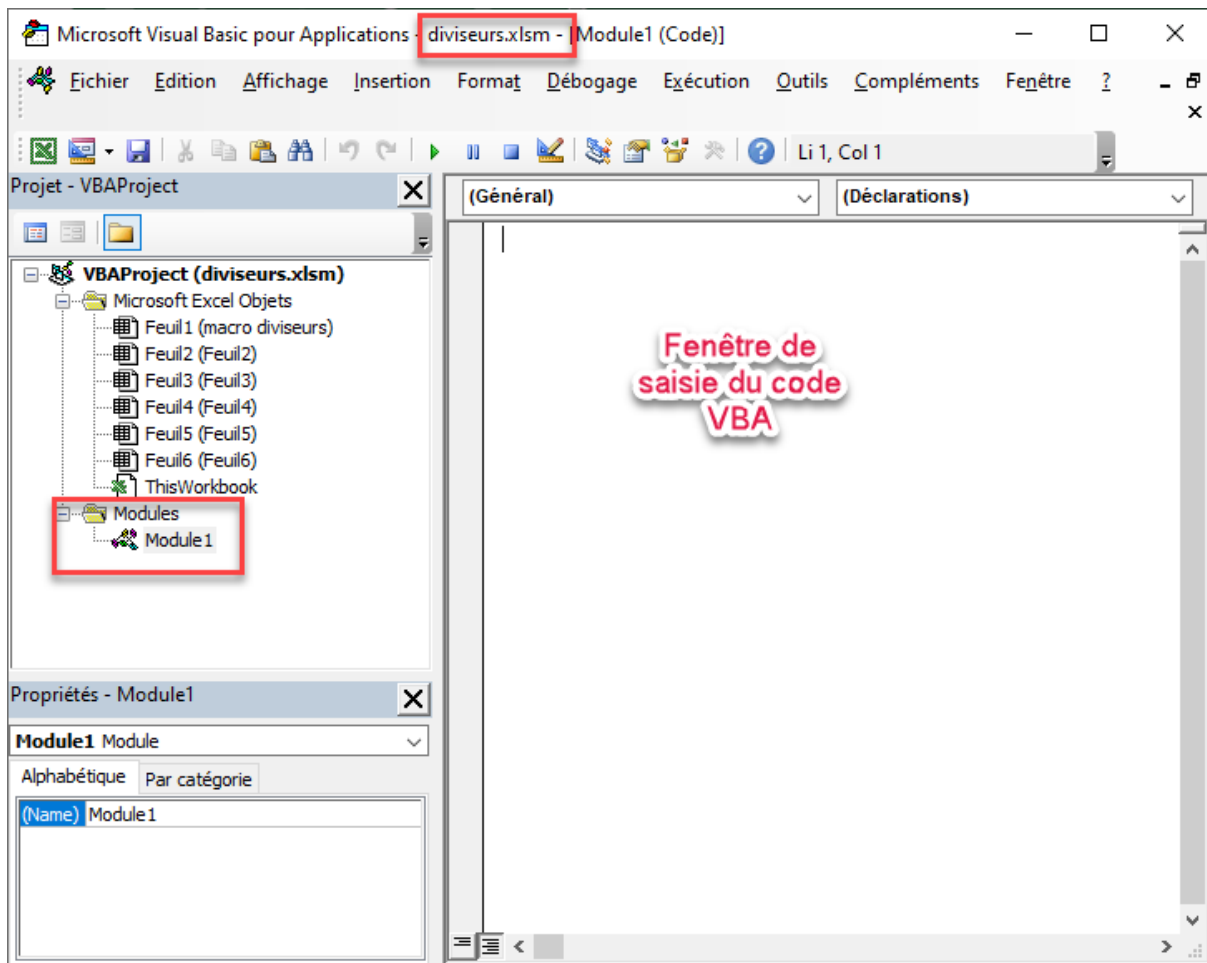
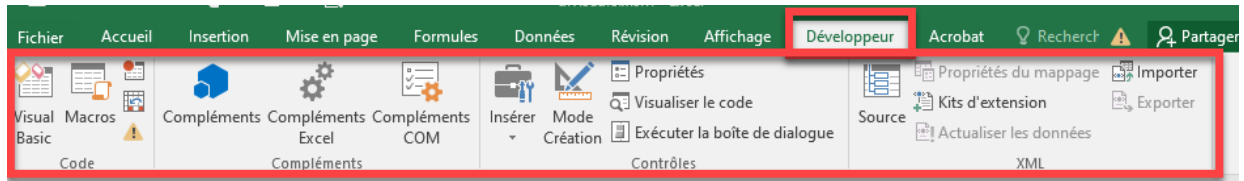
Tout **tableur comprend des objets** qui permettent de réaliser des modèles pour résoudre des problèmes de gestion comme le classeur, la feuille, la colonne, la ligne, la plage.

#### Exemple

Pour utiliser un modèle de facturation implanté dans une feuille, il faut ouvrir le classeur, la feuille pertinente, renseigner les données dans les cellules adaptées et enregistrer le résultat.

VBA est accessible à partir de l'onglet développeur ( Alt+F11 ).

Un projet VBA est associé à chaque classeur Excel ; il regroupe différents modules dans lesquels seront implantés les codes VBA. Les modules peuvent contenir un ou plusieurs programmes VBA. Ils permettent l'organisation de ces programmes.



## B. L'algorithme, préalable indispensable à l'écriture d'un programme

Un programme est structuré et la manipulation du langage peut s'avérer complexe. Un travail préalable de réflexion préside ; **il conduit à l'écriture d'un algorithme**

### 3. La structure générale d'un algorithme

#### Définition

Un **algorithme** est l'énoncé d'un ensemble fini d'instructions de traitement élémentaires, classées portant sur des données. Son exécution résout un problème.

Un algorithme ne dépend pas d'un langage informatique particulier.

#### 4. La structure générale d'un algorithme :

<b>En-tête : déclaration des variables et des constantes</b>	<b>Nom (titre) de l'algorithme</b> <b>Variables</b> : Déclaration (énumération) des variables et de leur type <b>Constantes</b> : Déclaration des constantes (valeurs)
<b>Corps de l'algorithme</b>	Début : début des instructions donc de l'algorithme Instruction 1 Instruction 2 .... Fin (de l'algorithme)

#### 5. Les variables et les constantes (paramètres) dans l'algorithme

##### Définition

Une **variable** est la représentation symbolique d'une donnée.

La valeur d'une variable peut changer au fur et à mesure de l'exécution de l'algorithme.

La variable peut recevoir une valeur :

- par affectation directe (généralement notée  $\leftarrow$ ) ;
- par acquisition (saisie/entrée provenant d'un utilisateur ou lecture sur un fichier).

##### Exemple

```
A ← 23 ; B ← "Salaire"  
Lire C
```

Les variables peuvent être de type :

- Numérique (entier ou réel),
- Alphanumérique (caractère ou chaîne de caractères, texte)
- Ou logique (variable booléenne, vrai/faux ou oui/ non).

Par assimilation de langages de programmation, on distingue parfois d'autres types de variable (date, pourcentage...).

Déclarer une variable

La déclaration de type se formule ainsi :

**<Nom variable> : <Type>.**

### Exemple

**Nom client** : chaîne de caractères ;  
**Total compte** : réel ;  
**Soldé** : logique

### Définition

Une **constante** (ou paramètre) est une grandeur qui ne change pas de valeur tout au long du déroulement de l'algorithme.

La première instruction de l'algorithme consiste généralement à **lire (acquérir, entrer, saisir...)** les informations de départ dans des variables, la dernière à **écrire (sortir, afficher, éditer...)** et le résultat obtenu.

### Exemple

**Début**  
**A, B, C** : entier  
**Lire A, B**  
**C** ← A+B  
**Ecrire C**  
**Fin**

## C. Les variables et constantes en VBA

### 1. Le début et la fin d'un programme

Tout comme un algorithme, un programme VBA a un nom. Il commence, en principe (excepté pour une fonction), par le mot-clé « **Sub** », suivi du nom du programme et de « **()** » et se termine par « **End Sub** »

### Exemple

```
Sub diviseurs()  
    a = Cells(1, 1)  
    n = Int(Sqr(a))  
  
    i = 1  
  
    For k = 1 To n  
  
        q = Int(a / k)  
        r = a - q * k  
  
        If r = 0 Then  
            Cells(i, 2) = k  
            Cells(i, 3) = q  
            i = i + 1  
        End If  
  
    Next  
  
End Sub
```

```

(Général)
Sub TTC() 'permet de calculer un prix TTC à partir du HT
End Sub

```

La parenthèse anglo-saxonne ( quote ) permet d'introduire un commentaire (en vert) qui est sans effet sur le programme mais permettra de le documenter.

## 2. Les variables

Les variables définies dans l'en-tête de l'algorithme sont déclarées en début de programme ( tab. 16.2 ). Cette déclaration (« Dim nom\_de\_variable As Type ») est facultative mais très utile pour contrôler le programme ou résoudre un éventuel problème

### Déclaration de variable

<b>Dim</b>	L'instruction qui débute par Dim permet de déclarer la variable.
<b>Nom_de_variable</b>	Ce nom sans espaces est celui qui a été retenu pour la variable, il doit être explicite.
<b>As</b>	Ce terme va permettre de définir le type de la variable.
<b>Type</b>	Numérique, texte, date.

### Exemple :

```
Dim TTC As Single
```

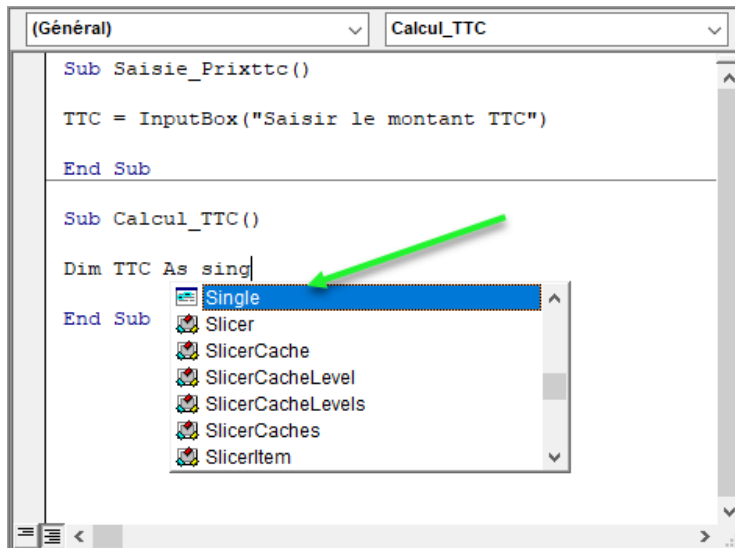
Les types de variables diffèrent selon leur contenu

<b>String</b>	Texte
<b>Byte</b>	Nombre entier de 0 à 255
<b>Integer</b>	Autres entiers plus élevés, y compris les entiers négatifs
<b>Long</b>	Entiers longs positifs ou négatifs
<b>Single</b>	Nombre décimaux sans fixation de la virgule (valeurs peu importantes)
<b>Double</b>	Nombre décimaux sans fixation de la virgule (valeurs élevées)
<b>Date</b>	Date et heure
<b>Boolean</b>	Logique, True (vrai) ou False (Faux)

<b>Variant</b>	Tout type, qui sera, par défaut, celui d'une variable non déclarée, la première valeur affectée définira le type.
<b>Object</b>	Objet Microsoft

### Exemple de déclaration d'une variable.

Nous sommes assistés...



### Les paramètres (constantes)

La constante est définie comme une variable mais avec le mot clé « Const » et, de préférence, l'indication du type. Il suffit ensuite de préciser sa valeur comme suit :

**Const Nom\_Constante As Type = valeur**

**'Taux TVA indiqué en format décimal**

**Const TXTVA As Single = 0.2**

# Les familles d'instruction

## D. Les instructions de base

### 1. Les instructions d'entrée

Pour saisir des données d'entrée, nous pouvons :

1. Utiliser la boîte de saisie InputBox()
2. Utiliser l'une des instructions Cells ou Range

L'instruction mobilisant la fonction « Inputbox » permet d'affecter une valeur à une variable.

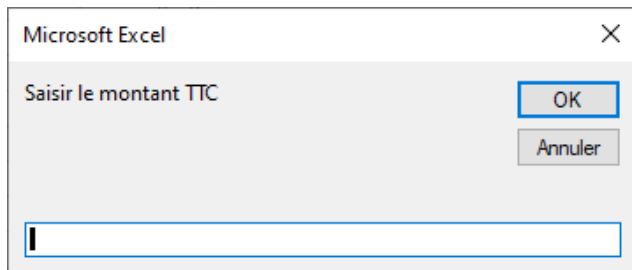
Sa syntaxe est la suivante :

```
Nom_Variable = Inputbox("message personnalisé pour guider l'utilisateur")
```

Lors de l'exécution du programme, une boîte de dialogue apparaîtra ; elle indique à l'utilisateur l'action à effectuer.

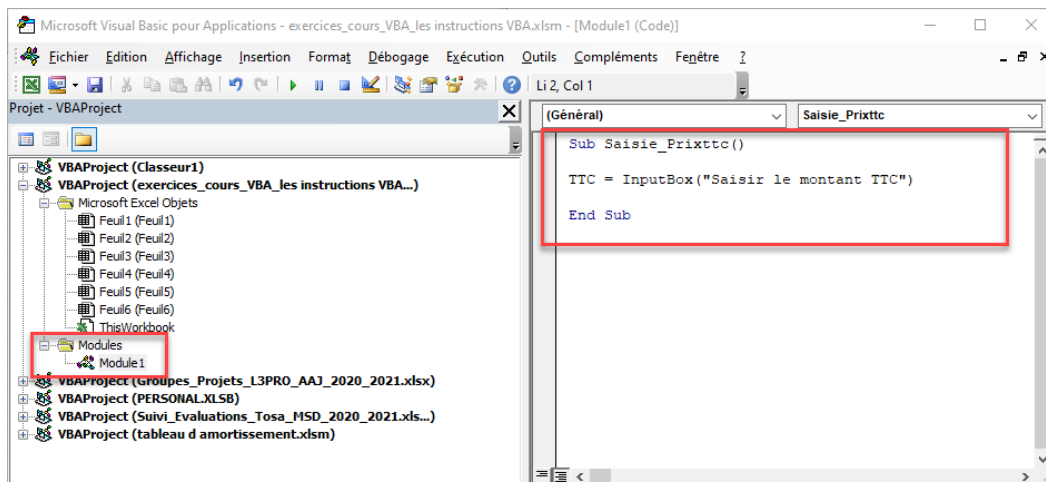
**Exemple :**

```
TTC = InputBox("Saisir le montant TTC")
```



Pour faire afficher cette boîte de dialogue et comme indiqué par l'image ci-dessous :

- Ouvrir l'éditeur VBA
- Insérer un module
- Et saisir une procédure sub et lui donner un nom comme « Saisie\_Prix\_TTC » par exemple.
- Mettre l'instruction :
- `TTC = InputBox("Saisir le montant TTC")`



Une fois la donnée « Montant\_TTC » est saisie dans la boîte de dialogue, nous allons récupérer la valeur résultat « TTC » et l'affecter à notre une cellule d'une feuille du classeur Excel actif par exemple.

Pour le faire, Les méthodes « **Cells** » et « **Range** » permettent également l'affectation de valeurs d'entrée en se référant au contenu de cellules ( tab. 16.4 ).

Méthode	Argument
<b>Range</b>	("référence cellule")
<b>Cells</b>	(index_ligne,index_colonne)

### Exemple

Le montant TTC est renseigné dans la cellule B1 de la Feuille active. L'affectation du contenu de cette cellule à la variable TTC est réalisée par :

```
TTC = Range ( " B1 " ) ou TTC = Cells (2,2)
```

Quand la cellule à laquelle on fait référence se trouve dans la feuille à partir de laquelle on exécute la macro-commande (feuille active ou Active Sheet ), la référence peut être omise.

Quand la cellule se trouve dans une autre feuille, cette dernière doit être activée par, au choix :

```
Application.Goto Sheets("Nomfeuille").Range("référence cellule")
```

```
Application.Goto Sheets("Nomfeuille").Cells(index_ligne, index_colonne)
```

On peut également activer la feuille de calcul au préalable, puis préciser l'instruction, en rédigeant :

```
Sheets("Nomfeuille").Activate
```

La sélection d'une plage est effectuée par :

```
Range(référence plage).select
```

Pour sélectionner une plage nommée "Nom\_plage", on écrit :

```
Range("Nom_plage")
```

## 2. Les instructions de calcul

Les programmes nécessitent souvent de réaliser des calculs. Le résultat d'un calcul est affecté à une variable et peut mobiliser divers opérateurs mathématiques appliqués à des variables. Ils s'écrivent simplement.

L'utilisation des parenthèses permet de respecter les règles de calcul sous-jacentes.

### Exemple :

HT et TTC sont des variables préalablement définies, TXTVA est un paramètre également défini et affecté d'une valeur :

```
HT = TTC / (1 + TXTVA)
```

Pour faire référence au contenu d'une cellule dans une formule de calcul, on peut également utiliser les méthodes « Cells » ou « Range ».



### Exemple :

Le contenu du montant hors taxe étant renseigné dans la cellule B1, le paramètre TXTVA est défini et affecté d'une valeur, la variable TTC est également définie.

```
TTC = Range("B1") * (1 + TXTVA)
```

### 3. Les instructions de sortie

Pour afficher un résultat dans une cellule, on peut utiliser :

1. Utiliser la boîte d'affichage **MsgBox()**
2. Utiliser l'une des méthodes ou instructions **Cells** ou **Range**

### Exemple :

La cellule B2 reçoit la valeur de la variable TTC calculée par le programme :

```
Range("B2") = TTC
```

Très utile en VBA, l'instruction MsgBox permet la sortie (affichage) des résultats d'un programme en rédigeant simplement :

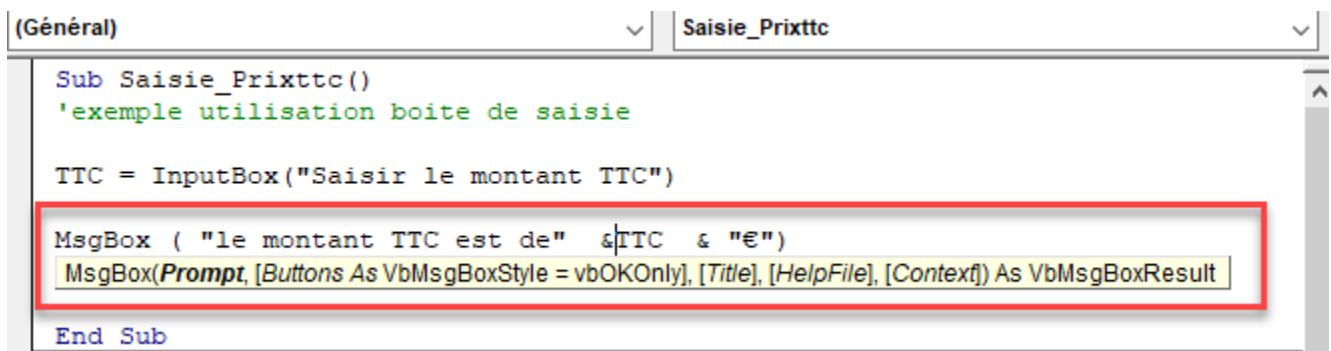
```
MsgBox("Texte explicatif" & Nom_Variable & "Texte explicatif")
```

Le signe « & » (« esperluette » ou « et commercial ») est un opérateur de concaténation permettant d'assembler du texte (chaîne de caractères), des valeurs et des expressions.

Exemple

L'instruction suivante affiche le résultat dans une boîte de dialogue :

```
MsgBox ("le montant TTC est de" & TTC & "€")
```



```
(Général) | Saisie_Prixttc  
Sub Saisie_Prixttc()  
'exemple utilisation boite de saisie  
  
TTC = InputBox("Saisir le montant TTC")  
  
MsgBox ("le montant TTC est de" & TTC & "€")  
MsgBox(Prompt, [Buttons As VbMsgBoxStyle = vbOKOnly], [Title], [HelpFile], [Context]) As VbMsgBoxResult  
  
End Sub
```

La fonction MsgBox permet également d'afficher une information utile pour l'utilisateur. Elle peut surtout poser une question à l'utilisateur via une boîte de dialogue. La question posée attendra une réponse à valider par un bouton VBA.

Sa syntaxe est la suivante :

```
Nom_Variable = MsgBox("Message", Type, "Titre de la fenêtre")
```

- Variable : variable de type Entier qui recevra la valeur du bouton
- Message : texte affiché comme message dans la boîte de dialogue

- Type : bouton VBA qui peut être référencé par son nom ou une expression numérique qui contrôle les boutons et les icônes à afficher
- Titre : message affiché dans la barre de titre de la boîte de dialogue

### Exemple

REP = msgbox("voulez-vous calculer un autre montant", vbYesNo, "Stop ou Encore")

La variable REP prend la valeur 6 si la réponse est « oui » ; 7, dans le cas contraire, ce qui pourra être utile pour rédiger certains programmes avec des boucles. L'aide Excel donne ces différentes valeurs de réponse pour les boutons.

## A faire en TD

### 4. Exercice Récapitulatif de programme simple

Le programme suivant résume les différentes instructions de base à mobiliser en VBA. Il calcule un montant TTC à partir d'un prix HT choisi par l'utilisateur.

```

Algorithme MontantTTC
Variables : TTC, HT : réels
Constante TXTVA = 0.2
Début
    Lire HT
    TTC ← HT*(1+TVA)
    Ecrire HT, TTC
Fin
  
```

Résultat :

