

FICHE DE COURS GRATUITE

VBA EXCEL | BASES

# Sub et Function

Tout code VBA vit dans une procédure. Une Sub agit, une Function calcule et renvoie un résultat. Savoir choisir entre les deux, et les faire dialoguer, c'est la base de toute macro bien construite.

|                      |                   |
|----------------------|-------------------|
| Version              | V1.2              |
| Dernière mise à jour | 29/10/2026        |
| Niveau               | 1 / 3, Découverte |
| Durée de lecture     | 7 min             |
| Auteur               | Clément Delobel   |



Version en ligne  
mesétudesdecas.fr/vba/  
sub-et-function.html

## 01 Définition

### EN UNE PHRASE

Une **Sub** est une procédure qui exécute des actions. Une **Function** est une procédure qui **renvoie une valeur**, utilisable dans le code ou directement dans une cellule Excel.

### Sub

Met en forme, copie, supprime, envoie un message.  
Elle apparaît dans la liste des macros (Alt + F8).

### Function

Calcule et renvoie un résultat. Elle n'apparaît pas dans la liste des macros, mais dans les fonctions Excel.

## 02 Syntaxe

### SYNTAXE GÉNÉRALE

```
Sub NomDeLaMacro()  
    ' instructions  
End Sub  
  
Function NomDeLaFunction(argument As Type) As TypeRetour  
    ' calcul  
    NomDeLaFunction = resultat  
End Function
```

| Élément            | Rôle                                                        |
|--------------------|-------------------------------------------------------------|
| Sub / Function     | Mot-clé qui ouvre la procédure.                             |
| (argument As Type) | Données transmises à la procédure. Facultatif pour une Sub. |
| As TypeRetour      | Type de la valeur renvoyée par une Function.                |

| Élément                | Rôle                                                                |
|------------------------|---------------------------------------------------------------------|
| NomDeLaFunction = ...  | Affecter une valeur au nom de la fonction, c'est ce qui la renvoie. |
| End Sub / End Function | Fin de la procédure.                                                |

### 03 Créer une fonction personnalisée

Une Function écrite dans un **module standard** devient utilisable dans les cellules, comme SOMME ou RECHERCHEX.

UNE FONCTION TTC AVEC TAUX FACULTATIF

```
Function TTC(montantHT As Double, Optional taux As Double = 0.2) As Double
    TTC = montantHT * (1 + taux)
End Function
```

Dans la feuille, on écrit `=TTC(A2)` pour un taux de 20 %, ou `=TTC(A2; 0,055)` pour un taux de 5,5 %. Le mot-clé `Optional` rend l'argument facultatif et lui donne une valeur par défaut.

 **ASTUCE**

#### Retrouver ses fonctions

Vos fonctions personnalisées apparaissent dans l'assistant fonction (bouton fx), catégorie **Personnalisées**.

### 04 Appeler une procédure depuis une autre

Découper une macro en petites procédures la rend plus lisible. Une Sub peut en appeler une autre, et utiliser le résultat d'une Function.

UNE MACRO PRINCIPALE QUI DÉLÈGUE

```
Sub Traitement()
    PreparerFeuille
    Call MettreEnForme("A1:F1")
    MsgBox "Total TTC : " & TTC(Range("B10").Value)
End Sub

Sub PreparerFeuille()
    Range("A2:F500").ClearContents
End Sub

Sub MettreEnForme(adresse As String)
    Range(adresse).Font.Bold = True
End Sub
```

### 05 ByVal et ByRef

Par défaut, VBA transmet les arguments **ByRef** (par référence) : si la procédure appelée modifie l'argument, la variable d'origine est modifiée aussi. Avec **ByVal**, la procédure travaille sur une copie.

## LA DIFFÉRENCE EN PRATIQUE

```
Sub Doubler(ByVal n As Long)
    n = n * 2          ' la variable d'origine ne change pas
End Sub
```

## 06 Pièges à éviter

 PIÈGE**Oublier d'affecter le résultat**

Une Function qui ne contient pas la ligne `NomDeLaFunction = ...` renvoie 0 ou une valeur vide, sans message d'erreur.

 PIÈGE**Une fonction de cellule ne modifie rien**

Appelée depuis une cellule, une Function ne peut que renvoyer une valeur : elle ne peut pas mettre en forme, écrire dans une autre cellule ou supprimer une ligne.

 PIÈGE**Le mauvais module**

Une fonction placée dans le module d'une feuille (Feuil1) ou dans ThisWorkbook n'est pas visible dans les cellules. Insérez un module standard : **Insertion > Module**.

 PIÈGE**Private cache la macro**

Une `Private Sub` n'apparaît pas dans la liste des macros. Pratique pour les sous-procédures, déroutant quand on l'ignore.