FICHE DE COURS GRATUITE

VBA EXCEL | BOUCLES

La boucle For Next

C'est la boucle que l'on écrit le plus en VBA. Dès qu'il faut répéter la même action sur dix, cent ou dix mille lignes, For Next fait le travail à votre place. Et pour bien la comprendre, le plus efficace est encore de la regarder tourner.

Version	V1.2
Dernière mise à jour	29/10/2026
Niveau	2 / 3, Intermédiaire
Durée de lecture	9 min
Auteur	Clément Delobel

Version en ligne
mesetudesdecas.fr/vba/boucle-for-next.html

01 Définition

EN UNE PHRASE

For Next répète un bloc d'instructions **un nombre de fois connu à l'avance**, en faisant varier un compteur d'une valeur de départ à une valeur de fin.

Le compteur, souvent nommé `i`, sert à deux choses : savoir combien de tours il reste, et désigner une ligne, une colonne ou un élément différent à chaque tour. C'est ce qui permet d'écrire `Cells(i, 1)` pour traiter A1, puis A2, puis A3, sans recopier la même ligne de code.

02 Syntaxe

SYNTAXE GÉNÉRALE

```
For compteur = debut To fin [Step pas]
    ' instructions répétées à chaque tour
    [Exit For]
Next [compteur]
```

Élément	Rôle
`compteur`	Variable numérique, déclarée de préférence `As Long`. Elle change de valeur à chaque tour.
`debut`	Valeur du compteur au premier tour.
`fin`	Dernière valeur traitée. La boucle s'arrête dès que le compteur la dépasse.
`Step pas`	Facultatif. De combien le compteur avance à chaque tour : 1 par défaut, 2 pour une ligne sur deux, -1 pour remonter.
`Exit For`	Facultatif. Sort immédiatement de la boucle, sans attendre la fin.

Élément	Rôle
`Next compteur`	Fin du bloc répété. Rappeler le nom du compteur est facultatif mais rend le code plus lisible.

03 Simulateur pas à pas

Cliquez sur **Pas à pas** (ou appuyez sur `F8`) pour exécuter le code une ligne à la fois, exactement comme dans l'éditeur VBA. La ligne surlignée en jaune est celle qui **va** être exécutée. Observez la variable `i` dans les Variables locales et la feuille qui se remplit.

Simulateur interactif

Cette fiche existe en version interactive : le code s'exécute ligne par ligne, les variables évoluent et la feuille se remplit en direct. Scannez le code ou rendez-vous sur la page en ligne.

mesetudesdecas.fr/vba/boucle-for-next.html

Ce que fait la boucle, tour par tour

Pour `For i = 1 To 5` avec `Cells(i, 1).Value = i` :

Tour	Valeur de i	Action	Après Next
1	1	écrit 1 en A1	i passe à 2
2	2	écrit 2 en A2	i passe à 3
3	3	écrit 3 en A3	i passe à 4
4	4	écrit 4 en A4	i passe à 5
5	5	écrit 5 en A5	i passe à 6 : 6 dépasse 5, la boucle s'arrête

💡 ASTUCE

Le même affichage dans Excel

Placez le curseur dans la macro, appuyez sur F8, puis affichez les Variables locales par le menu **Affichage**. Réduisez la fenêtre VBA pour voir la feuille réagir en même temps. Détails dans la fiche **Déboguer pas à pas**.

04 Exemples à copier

Traiter toutes les lignes d'un tableau, jusqu'à la dernière

Le cas le plus courant : on ne sait pas combien de lignes contient le tableau, alors on calcule la **dernière ligne** avant de lancer la boucle.

CALCULER UN PRIX TTC SUR CHAQUE LIGNE

```
Sub CalculerTTC()
    Dim i As Long, derniereLigne As Long

    ' Dernière ligne remplie de la colonne A
    derniereLigne = Cells(Rows.Count, 1).End(xlUp).Row

    ' Ligne 1 = en-têtes, on commence donc à 2
    For i = 2 To derniereLigne
        Cells(i, 3).Value = Cells(i, 2).Value * 1.2
    Next i
End Sub
```

Une ligne sur deux avec Step

COLORER UNE LIGNE SUR DEUX

```
Sub UneLigneSurDeux()
    Dim i As Long, derniereLigne As Long
    derniereLigne = Cells(Rows.Count, 1).End(xlUp).Row

    For i = 2 To derniereLigne Step 2
        Rows(i).Interior.Color = RGB(235, 239, 255)
    Next i
End Sub
```

Supprimer des lignes en partant du bas

SUPPRIMER LES LIGNES DONT LA COLONNE A EST VIDE

```
Sub SupprimerVides()
    Dim i As Long
    For i = Cells(Rows.Count, 2).End(xlUp).Row To 2 Step -1
        If Cells(i, 1).Value = "" Then Rows(i).Delete
    Next i
End Sub
```

Sortir dès qu'on a trouvé, avec Exit For

RECHERCHER UN CLIENT

```
Sub ChercherClient()
    Dim i As Long, derniereLigne As Long, cherche As String
    cherche = InputBox("Nom du client recherché ?")
    derniereLigne = Cells(Rows.Count, 1).End(xlUp).Row

    For i = 2 To derniereLigne
        If Cells(i, 1).Value = cherche Then
            MsgBox cherche & " trouvé en ligne " & i
            Exit For    ' inutile de continuer
        End If
    Next i

    ' Si la boucle est allée au bout, i vaut derniereLigne + 1
    If i > derniereLigne Then MsgBox cherche & " introuvable."
End Sub
```

Deux boucles imbriquées

Une boucle pour les lignes, une autre pour les colonnes : chaque tour de la boucle extérieure déclenche la boucle intérieure en entier.

TABLE DE MULTIPLICATION 10 X 10

```
Sub TableMultiplication()
    Dim ligne As Long, colonne As Long

    For ligne = 1 To 10
        For colonne = 1 To 10
            Cells(ligne, colonne).Value = ligne * colonne
        Next colonne
    Next ligne
End Sub
```

05 Pièges à éviter

⚠ PIÈGE

Supprimer des lignes en descendant

Si vous supprimez la ligne 3, l'ancienne ligne 4 remonte en 3… et la boucle passe directement à 4 : une ligne est sautée. Pour supprimer, on part toujours du bas avec `Step -1`.

⚠ PIÈGE

Une boucle qui ne tourne jamais

`For i = 10 To 1` sans `Step -1` ne fait aucun tour, et sans aucun message d'erreur. Quand le départ est plus grand que la fin, le pas négatif est obligatoire.

⚠ PIÈGE

La fin est calculée une seule fois

La valeur de fin est lue au démarrage de la boucle. Si votre macro ajoute des lignes pendant la boucle, celles-ci ne seront pas traitées.

⚠ PIÈGE

Integer au lieu de Long

Un compteur déclaré `As Integer` plante au-delà de 32 767 (erreur 6, dépassement de capacité). Déclarez toujours vos compteurs `As Long`.

⚠ PIÈGE

Modifier le compteur dans la boucle

Écrire `i = i + 1` à l'intérieur de la boucle s'ajoute au Next et décale tout. Si le pas doit changer, utilisez `Step`, ou passez à une boucle **Do While**.

06 For Next ou For Each ?

For Next

Quand on a besoin du **numéro** : la ligne en cours, une colonne voisine, un pas particulier ou un sens de parcours inversé.

For Each

Quand on veut simplement passer sur **chaque élément** d'un ensemble : chaque cellule d'une plage, chaque feuille d'un classeur. Voir **For Each**.