

# Queste de savoir

Les plages nommées avec Excel en VBA

---

dimanche 25 août 2024



# Table des matières

	Introduction . . . . .	1
1.	Noms définis . . . . .	1
1.1.	Définition d'un nom . . . . .	2
1.2.	Gestionnaire de noms . . . . .	3
2.	Syntaxe VBA . . . . .	8
2.1.	Lecture d'une plage nommée . . . . .	8
2.2.	La collection Names . . . . .	9
2.3.	L'objet Name . . . . .	10
3.	Exemple final . . . . .	10
	Conclusion . . . . .	12

## Introduction

Un élément assez sommaire d'Excel, mais somme toute pas assez utilisé, est les plages nommées.

Donner un nom à une plage a de multiples avantages. D'une part, il est plus simple de retenir ou de retrouver un nom qu'une adresse. D'autre part, cela apporte aussi plus de flexibilité, car il n'y a pas d'impact quant au nom de la plage quand nous modifions la structure de la feuille (ajout de ligne, suppression de colonne, ...) contrairement à l'adresse.

De plus, une plage nommée peut être liée au classeur ou à une feuille. De ce fait, il est ainsi possible d'utiliser le même nom dans des feuilles différentes. C'est pratique lorsque plusieurs feuilles sont formatées pareilles ou contiennent des éléments similaires, par exemple avec une valeur "quantité produite" pour des feuilles "Lundi" et "Mardi".

Les plages nommées permettent donc de simplifier la conception et la maintenance de nos fichiers Excel et de nos macros.

Nous allons rappeler leur fonctionnement dans ce billet.

## 1. Noms définis

Depuis Excel, les plages nommées se gèrent avec le groupe "Noms définis" du ruban "Formules".



## 1. Noms définis

Pour le fil rouge de ce billet, nous démarrons en ajoutant trois feuilles à notre classeur : "Paramètres", "Lundi" et "Mardi". Leur contenu sera rempli au fur et à mesure.

### 1.1. Définition d'un nom

Il y a plusieurs façons de définir un nom. Ici, nous optons pour le choix "Définir un nom" du menu contextuel ou depuis le groupe "Noms définis" :

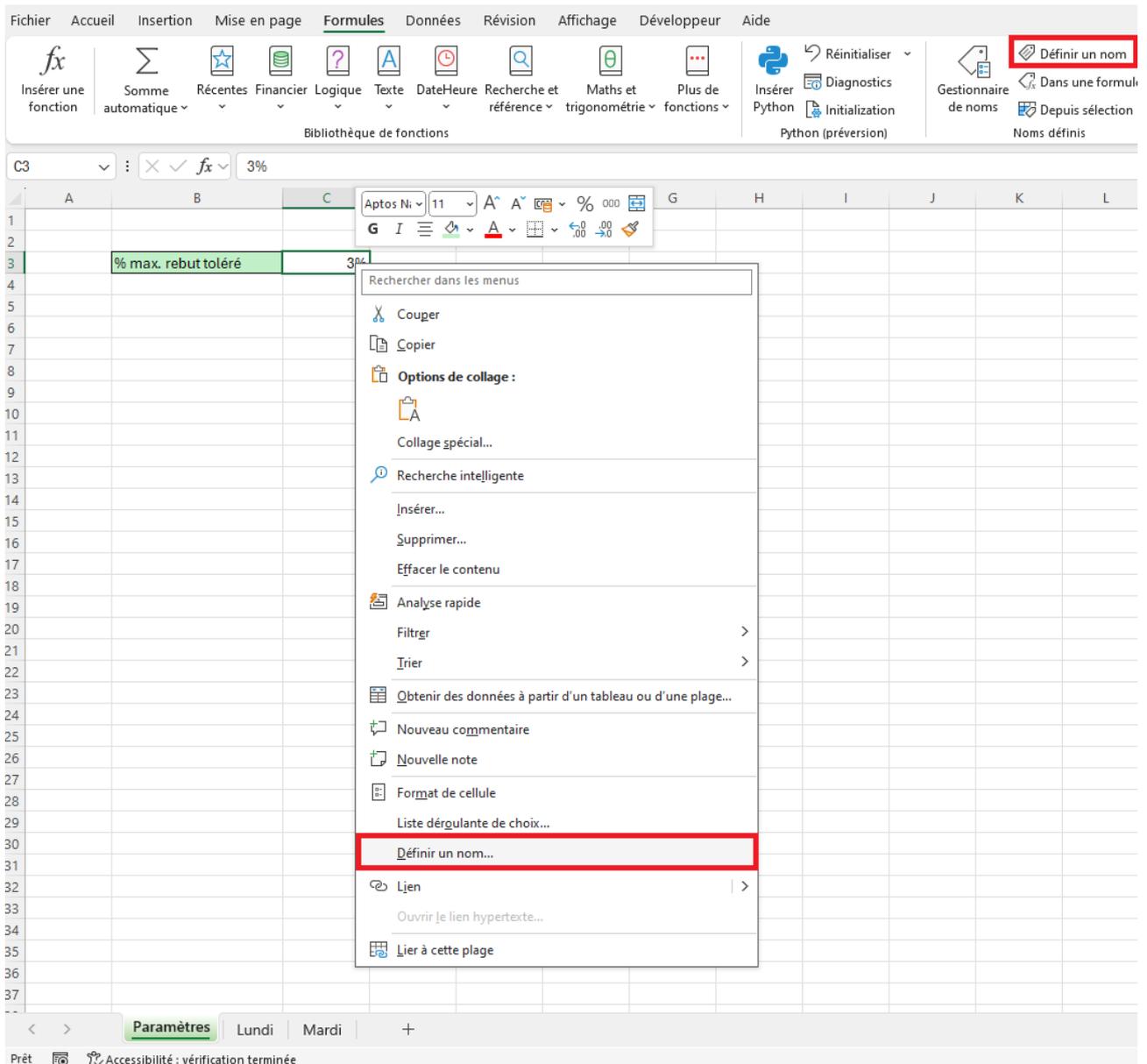


FIGURE 1.1. – Définir un nom (1).

Une fenêtre apparaît. Nous modifions le nom et le commentaire comme montré ci-dessous. La zone et la référence nous conviennent pour ce premier exemple.

## 1. Noms définis

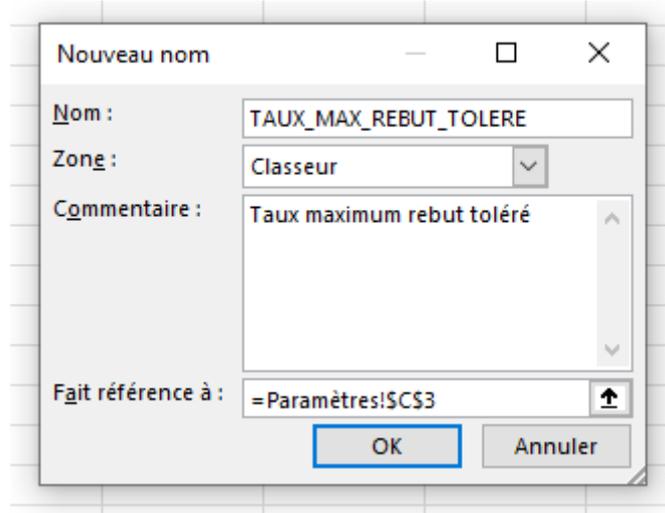
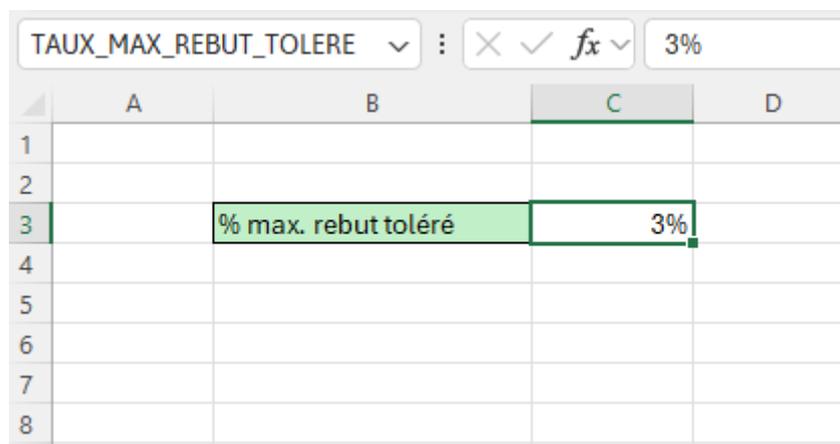


FIGURE 1.2. – Définir un nom (2).



Pour le nom, j'ai pour habitude d'utiliser des majuscules et des tirets bas `_`, parce que je vois ce nom comme une constante avec mon côté programmeur, mais libre à vous de faire autrement du moment qu'il soit parlant pour les autres usagers.

Après avoir validé, nous voyons que notre cellule **C3** a désormais le nom créé.



	A	B	C	D
1				
2				
3		% max. rebut toléré	3%	
4				
5				
6				
7				
8				

FIGURE 1.3. – Définir un nom (3).

### 1.2. Gestionnaire de noms

Le gestionnaire de noms liste les noms définis pour les plages standards, mais aussi pour d'autres éléments tels que les listes de données ou encore les zones d'impression.

Nous ouvrons le gestionnaire des noms via le bouton approprié du ruban. La fenêtre se présente ainsi :

## 1. Noms définis

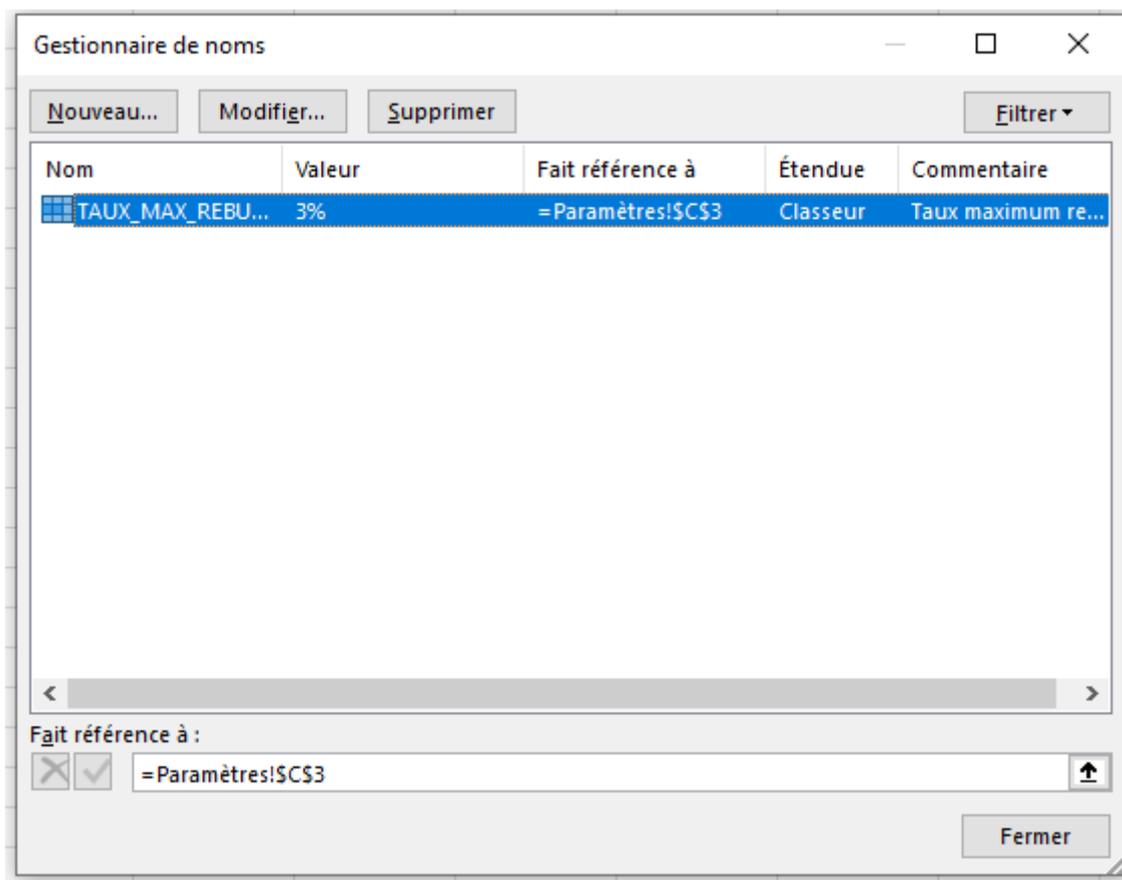


FIGURE 1.4. – Gestionnaire de noms.

Nous pouvons filtrer (avec la zone liste à droite) et trier (avec les entêtes de colonne) ces noms. De plus, nous pouvons également ajouter, modifier ou supprimer des noms définis.

Jusqu'à présent, nous n'avons créé qu'un seul nom qui a l'étendue classeur. Nous allons en ajouter trois avec le bouton "Nouveau..." pour la feuille "Lundi" : "QTE\_PRODUITE", "QTE\_REBUT" et "TAUX\_REBUT".

## 1. Noms définis

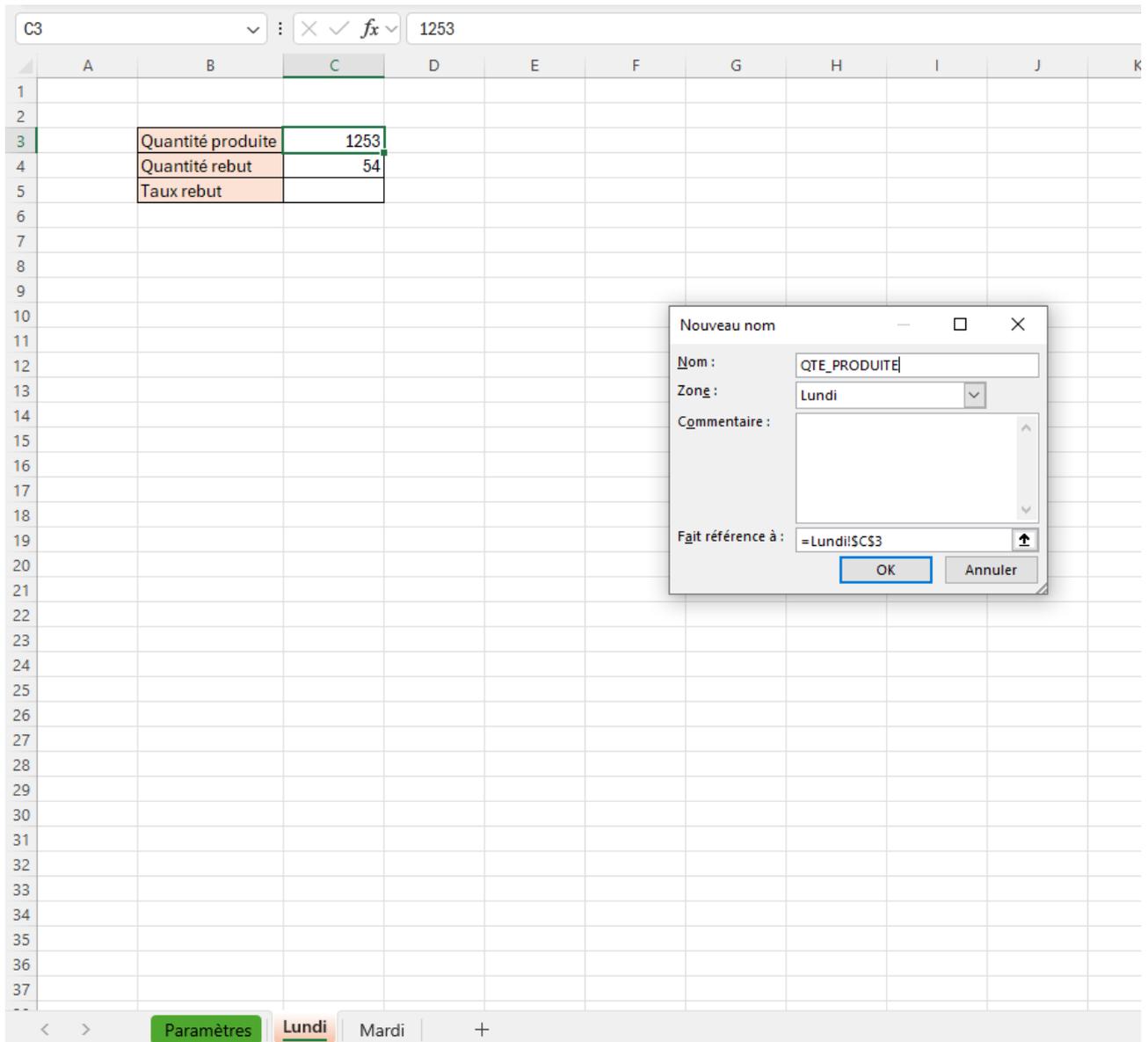


FIGURE 1.5. – Définir un nom pour une feuille.

Nos plages nommées créés, nous en profitons pour définir une formule. Remarquons que celle-ci est ainsi plus aisée à écrire et à lire :

## 1. Noms définis

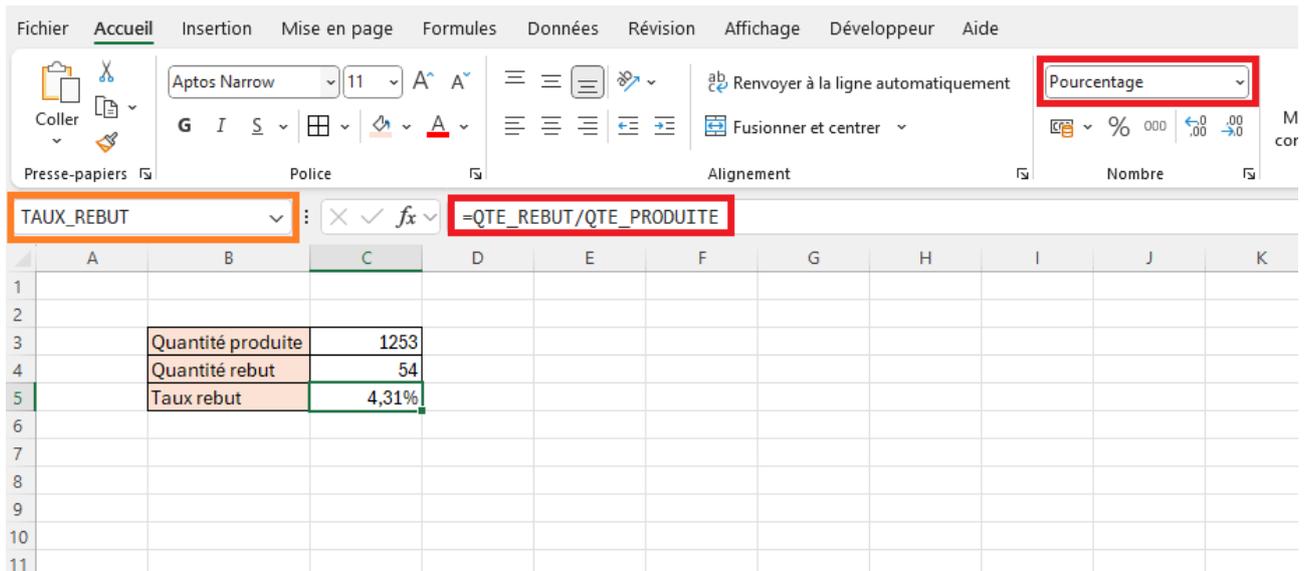


FIGURE 1.6. – Utilisation de plages nommées dans une formule.

i

Excel déduit automatiquement que nous faisons référence à la plage nommée de la feuille courante "Lundi" donc nous pouvons omettre le préfixe `Lundi!` ici.

Notons encore une fois comment la "Zone Nom" (encadrée en orange ci-dessus) nous permet de naviguer rapidement dans les plages nommées de la feuille courante et du classeur.

Passons maintenant aux autres jours de la semaine.

?

Cela veut dire que nous allons devoir tout refaire ? 🍊

Nous pourrions, ça nous ferait de l'entraînement, mais il y a plus rapide ! Nous pouvons simplement copier-coller notre tableau d'une feuille à l'autre, en l'occurrence de "Lundi" à "Mardi". Nous vérifions, et il faut tout de même redéfinir le nom pour le taux de rebut en `Mardi!C5`. Cela fait, nous modifions nos valeurs :

## 1. Noms définis

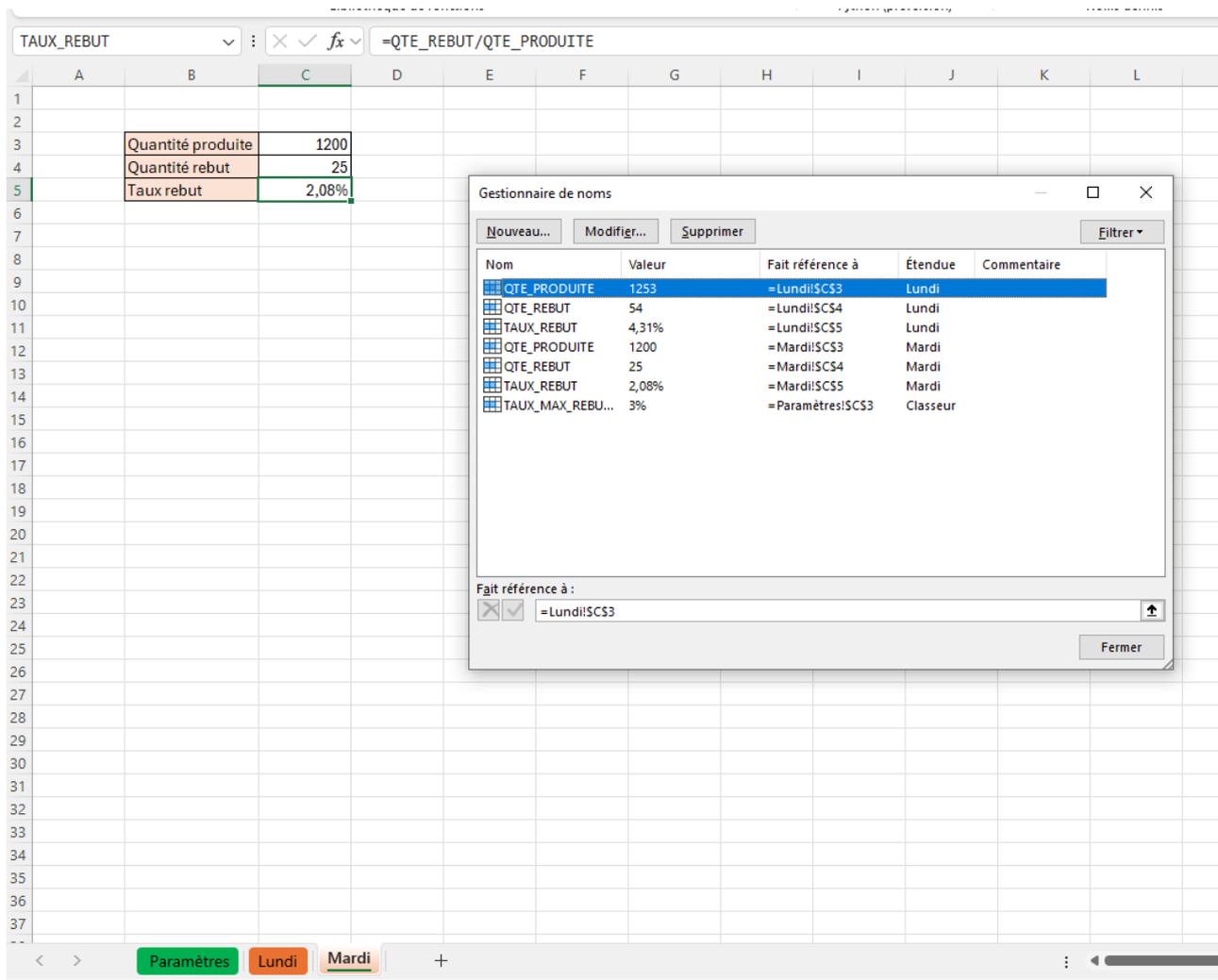


FIGURE 1.7. – Copier-coller plages nommées zone feuille.

Pour les derniers jours, nous allons faire encore mieux en utilisant la fonctionnalité "Déplacer ou copier..." puis en renommant les feuilles collées. Cette fois-ci, tous les noms sont bien reportés.

## 2. Syntaxe VBA

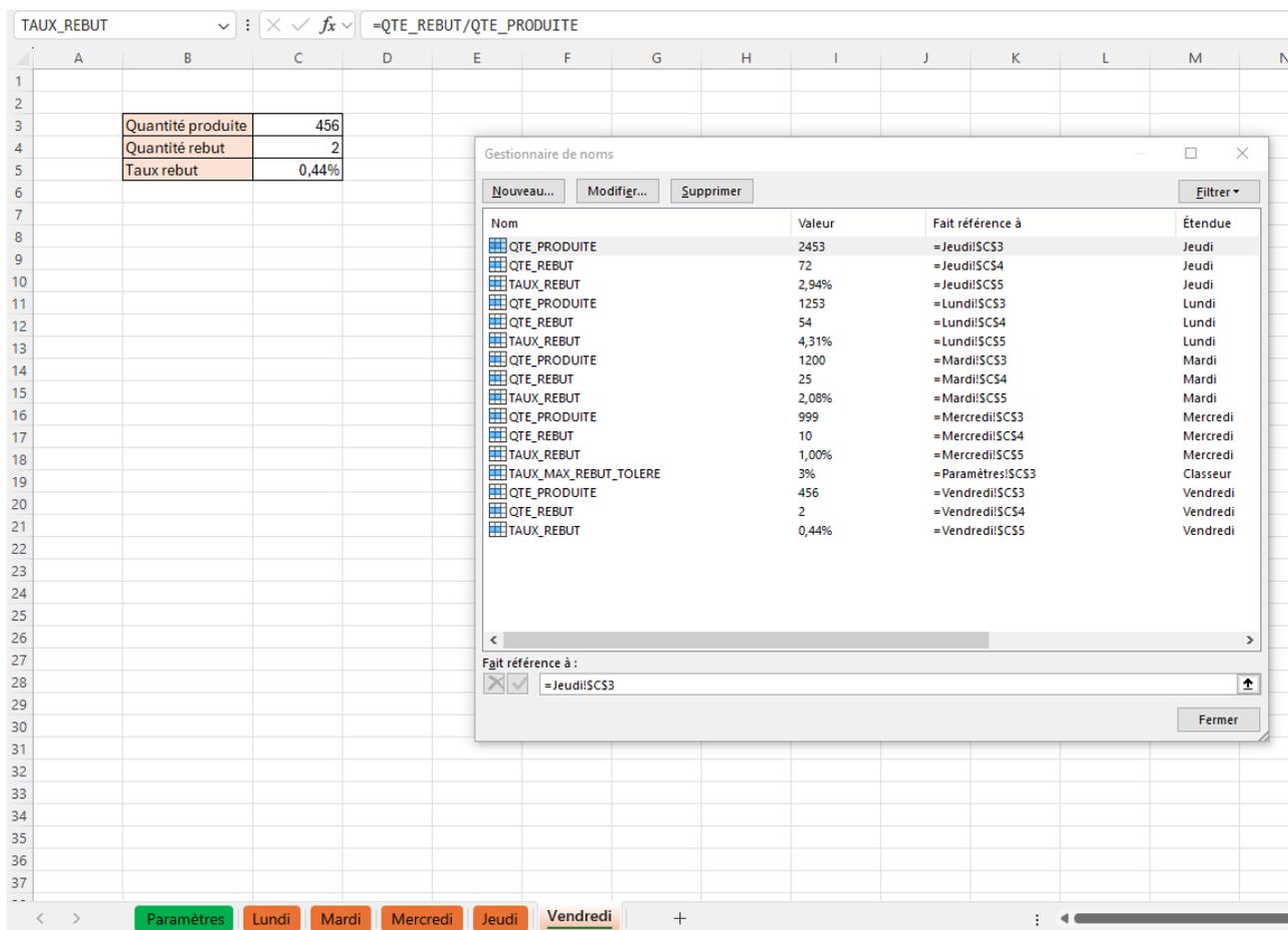


FIGURE 1.8. – Résultat final après copie et renommage des feuilles collées.

Le gestionnaire de noms contient bien tous nos noms définis selon l'étendue souhaitée.

*i*

Nous n'aurons pas recours à de grandes plages pour ce cours, sachez simplement qu'une plage nommée s'étendant sur plusieurs lignes ou colonnes sera automatiquement redimensionnée lors de l'ajout ou la suppression de ligne ou colonne à l'intérieur de celle-ci.

Au cours de cette section, nous avons vu comment gérer les noms définis du classeur avec Excel.

## 2. Syntaxe VBA

Il est temps de voir un peu de syntaxe VBA concernant les plages nommées.

### 2.1. Lecture d'une plage nommée

L'accès à une plage nommée peut se faire avec son nom au lieu de son adresse :

## 2. Syntaxe VBA

```
1 Sheets("Paramètres").Range("TAUX_MAX_REBUT_TOLERE") ' 0.03
2 ' équivalent de Sheets("Paramètres").Range("C3")
```

### 2.2. La collection Names

Les noms définis sont stockés dans la collection `Names` de la feuille, du classeur ou encore de l'application.

#### 2.2.1. Ajouter un nom

Pour rajouter un nom, il faut faire appel à la méthode `Add` de celle-ci en indiquant au moins un nom et une adresse.

```
1 Private Sub Exemple_Ajout_Names()
2     Sheets("Paramètres").Range("B4") = "Quantité min. à produire"
3     Sheets("Paramètres").Range("C4") = 1000
4
5     ' Ajout étendue classeur
6     ActiveWorkbook.Names.Add Name:="QTE_MIN_PRODUCTION",
7         RefersTo:)="Paramètres!$C$4"
8
9     ' Ajout étendue feuille paramètres
10    ActiveWorkbook.Sheets("Paramètres").Names.Add
11        Name:="QTE_MIN_PRODUCTION", RefersTo:)="Paramètres!$C$4"
12 End Sub
```

Elle accepte plusieurs types de coordonnées possibles comme indiqué dans la documentation. Si nous souhaitons ajouter un commentaire, il faut le faire ensuite en modifiant la propriété adéquate.

#### 2.2.2. Parcourir les noms

Comme `Names` est une collection, nous la parcourons comme attendu :

```
1 Private Sub Exemple_Iteration_Names()
2     Dim vName As Variant
3     For Each vName In ActiveWorkbook.Names
4         Debug.Print vName.Name, vName.Comment
5     Next vName
6 End Sub
```

### 3. Exemple final

## 2.3. L'objet Name

L'objet `Name` correspond à un nom défini.

### Modification

Il possède plusieurs propriétés, que ce soit pour lire/écrire le commentaire (`Comment`) ou encore la référence (`RefersTo` / `RefersToR1C1` / ...).

```
1 Private Sub Exemple_Ajout_Modification_Name()  
2     ' Nom défini étendue classeur  
3     ActiveWorkbook.Sheets("Paramètres").Names("QTE_MIN_PRODUCTION"  
4         ).Comment =  
5         "Feuille"  
6 End Sub
```



Je ne sais pas à quoi c'est dû, mais j'ai rencontré des comportements étranges à cause de ces propriétés commentaires durant mes tests. Il se peut que vous rencontriez vous-aussi des problèmes, ce qui vous évitera de chercher...

### Suppression

Un nom peut être supprimé avec son unique méthode `Delete`.

```
1 Private Sub Exemple_Suppression_Name()  
2     ' Nom défini étendue feuille  
3     ActiveWorkbook.Sheets("Paramètres").Names("QTE_MIN_PRODUCTION"  
4         ).Delete  
5 End Sub
```

Durant cette section, nous avons vu comment se traduisaient les plages nommées en VBA.

## 3. Exemple final

Voici un exemple final afin de montrer la flexibilité apportée par les plages nommées dans les macros.

Dans celui-ci, nous nous basons sur le classeur construit jusqu'à présent pour déterminer les jours où le taux de rebut a été supérieur au toléré ou bien la production n'a pas atteint son objectif (hors vendredi que nous considérons comme non travaillé en entier pour cette seconde condition).

### 3. Exemple final

```
1 Option Explicit
2
3 ' Feuille des paramètres
4 Private Const FEUILLE_PARAMS_NOM As String = "Paramètres"
5 Private Const PLAGE_TAUX_MAX_REBUT_TOLERE As String =
6     "TAUX_MAX_REBUT_TOLERE"
7
8 Private Const PLAGE_QTE_MIN_PRODUCTION As String =
9     "QTE_MIN_PRODUCTION"
10
11 ' Autres feuilles
12 Private Const PLAGE_QTE_PRODUITE As String = "QTE_PRODUITE"
13 Private Const PLAGE_TAUX_REBUT As String = "TAUX_REBUT"
14
15 Private Const JOUR_VENDREDI As String = "Vendredi"
16 Private Function JOURS()
17     JOURS = Array("Lundi", "Mardi", "Mercredi", "Jeudi",
18                 JOUR_VENDREDI)
19 End Function
20
21 Private Sub Main()
22     Application.ScreenUpdating = False
23
24     Dim dTauxMaxRebutTolere As Double
25     Dim dQteMinProductionTotere As Double
26     Dim vJour As Variant
27
28     dTauxMaxRebutTolere = Sheets(FEUILLE_PARAMS_NOM).Range(PLAGE_T
29     AUX_MAX_REBUT_TOLERE)
30     dQteMinProductionTotere =
31     Sheets(FEUILLE_PARAMS_NOM).Range(PLAGE_QTE_MIN_PRODUCTION)
32     For Each vJour In JOURS()
33         With Sheets(vJour)
34             If .Range(PLAGE_TAUX_REBUT) > dTauxMaxRebutTolere Or _
35             .Name <> JOUR_VENDREDI And
36             .Range(PLAGE_QTE_PRODUITE) <
37             dQteMinProductionTotere Then
38                 Debug.Print (vJour)
39                 Beep
40                 ' Lundi
41                 ' Mercredi
42             End If
43         End With
44     Next vJour
45
46     Application.ScreenUpdating = True
47 End Sub
```

*Conclusion*

## Conclusion

Durant ce billet, nous avons rappelé l'intérêt des plages nommées et leur usage.

Opter pour les plages nommées, c'est s'offrir un peu plus de tranquillité.

À bientôt !

Quelques ressources :

- La [documentation concernant la collection Names](#) ↗
- La [documentation concernant l'objet Name](#) ↗