

Compte Rendu

- Par la suite, ma 3^e semaine de mon stage avait pour but de continuer mon travail et de poursuivre ma fiche de suivi, pour ce faire nous avons utilisé de nouvelles notions que nous allons aborder par la suite.

Nous avons besoin de remplir la colonne Nombre occurrence du Mois de notre tableau Anomalies. Compter le nombre d'occurrence est important, cela permet de connaître le nombre d'incidents sur une période définie et de voir combien de fois cela se reproduit.

Pour compléter cette colonne, c'est un peu plus difficile car il y a plusieurs procédures à créer pour le bon fonctionnement de la récolte des données et aussi utilisé certain calcul car la période qui nous intéressent pendant l'année est le mois.

- Nous avons dans notre fiche **Liste_incidents** une colonne « Incidents lié ». Cette colonne nous intéresse car c'est dans cette colonne que nous avons besoin de récolter nos informations pour compléter le Nombre d'occurrence du mois.

La colonne « **incident lié** » comportent des données que si nous avons une anomalie. Donc si nous avons « oui » en anomalies, nous avons logiquement des incidents liés qui s'y trouvent.

Le nombre d'incidents lié représente donc le nombre d'occurrence. En lisant l'incidents nous avons déjà sa date.

Remarque : Certains incidents ne se trouvent pas dans l'intervalle défini, donc ils ne doivent pas être pris en compte car c'est que ceux qui se trouvent dans notre intervalle (date) que nous devons traiter.


Voilà à quoi ressemble un incident : **22017943**

Nous avons notre incident, il y a une façon précise pour le lire et surtout le comprendre.

Les 2 premiers chiffres représentent l'année : **22** (2022)

Les 2 suivants représentent le mois : **01** (Janvier)

Puis les 4 derniers représentent le numéro de l'incidents : 7943 (n°7943)



Explication : Durant le mois de mars, nous avons qu'un seul incident comportant 3 incidents liés, alors nous avons donc 3 occurrences pour le mois de mars.

Pour commencer, nous avons créé une fonction s'appelant **extraireNbIncidents**.

Cette fonction aura pour but comme son nom l'indique de nous renvoyer le nombre d'incidents qui se répètent dans le mois. On lui entre en paramètre tout ce qui est nécessaire pour compléter la colonne Nombre d'occurrence.

Nous commençons la fonction par déclarer nos différentes variables et en précisant leur type.

```
Function extraireNbIncidents(chaine As String)

    Dim tracesCompletes As Boolean
    Dim chainePeriode As String
    Dim tabIncidentsLies() As String, tabPeriode() As String
    Dim anneeDebut As Long, anneeFin As Long, moisDebut As Long, moisfin As Long
    Dim anneePrefixe As Long, nbOccurrences As Long
```

Comme vous aurez pu le remarquer nous avons utilisé **Function** et pas un **Sub**

Remarque :


La différence **Sub** et **Function** :

- Une fonction (**Function**) retourne une valeur.
- Une procédure (**Sub**) exécute une action et n'a pas de valeur de retour.

Selon quelques recherche :

Une fonction est généralement utilisée pour augmenter le nombre de fonctions déjà existantes dans Excel. Elle peut contenir des paramètres, et retournera donc une valeur dans la cellule dans laquelle elle est utilisée. Lorsque l'on écrit en VBA, et dans le souci d'organiser le code, une fonction pourra ne pas retourner de valeur dans une cellule mais dans le code (à la procédure appelante par exemple).

Le paramètre de la fonction est la chaîne de caractères.



Nous commençons par initialiser nos variables :

- Comme le **nombre d'occurrence**
- Puis l'**année préfixe** :

Nous l'avons initialisé à **2000** car quand nous allons procéder au calcul de l'année nous avons besoin que du chiffre de l'année par exemple avec « 2022 » on aurait fait 2022-2000 afin d'avoir que le « 22 ».

```
' init
tracesCompletes = False
anneePrefixe = 2000
nbOccurrences = 0
```

D'abord, nous devons commencer par sélectionner la date souhaitée par l'utilisateur.

- Nous cherchons dans la feuille **Liste_incidents** la période d'observation.
- Ensuite, je l'ai stockée dans une variable « **chainePeriode** » pour en faire un **Split**.

Le **Split** a pour intérêt de scinder une chaîne de caractères, dans notre situation nous avons 2 dates à extraire.

Par exemple : « Rapport Du 01/01/2022 au 22/04/2022 »


Pour cela, nous avons effectué :

```
chainePeriode = Sheets("Liste_incidents").Range("A1")
tabPeriode = Split(chainePeriode, "/")
```

La fonction **split** nous permet d'obtenir comme résultat :

Rapport du 01	01	2022 au 22	04	2022
---------------	----	------------	----	------

Vu que nous avons sélectionné directement la chaîne de caractères grâce au code, nous avons besoin que de la période d'observation et non de toute la chaîne de caractères.



Une fois la chaîne de caractères scindée je l'ai stockée dans une variable " **tabPeriode** ".

Les " / " nous permettent de séparer les différentes parties des 2 dates, c'est un paramètre à préciser afin d'avoir une séparation.

Ensuite, nous récupérons les informations qu'ils nous sont utiles pour la suite comme le **mois début**, **mois fin**, **année début** et **année fin**.

Nous affectons **les éléments de la chaîne de caractères** aux différentes **variables**.

Nous avons procédé à la suite avec ceci :

```
' le tableau commence à l'indice 0, donc on a besoin des indices 1, 2, 3 et 4
moisDebut = CInt(tabPeriode(1))
moisfin = CInt(tabPeriode(3))
anneeDebut = CInt(Left(tabPeriode(2), 4)) - anneePrefixe
anneeFin = CInt(tabPeriode(4)) - anneePrefixe
```

Le **CInt** permet de convertir du texte en nombre.

Le **Left** permet de récupérer les 4 premiers caractères (on ne veut pas suite), puis on convertit et on soustrait le préfixe.


Dans notre situation, notre chaîne de caractères représente nos incidents et vu qu'ils sont à la suite, nous les scindons afin d'avoir une vraie liste d'incidents (tableau).


Pour cela j'ai utilisé :

```
Debug.Print chaine & Chr(13)

tabIncidentsLies = Split(chaine, "; ")
```

22048756 ; 22068754 ; 22047854





Le résultat du **Split** nous permet d'avoir :

22046958 22105478 22056987

Le Chr permet de sauter des lignes et donc d'éviter de se perdre dans la lecture d'avoir une vrai liste incidents comme :

22046958

22105478

22056987

La partie la plus intéressante et surtout la plus compliqué de notre fonction est venue.

Nous avons créé une boucle permettant déjà de parcourir le tableau indice par indice jusqu'au dernier grâce au Ubound.

```
For i = 0 To UBound(tabIncidentsLies)
```





Nous avons ensuite ajouté un **If** permettant d'afficher en logs les détails des incidents qui se trouvent dans notre période d'observation.

```
For i = 0 To UBound(tabIncidentsLies)
    If tracesCompletes Then
        Debug.Print "détail des tests pour incident : " & tabIncidentsLies(i)
        Debug.Print Left(tabIncidentsLies(i), 2) & " >= " & anneeDebut & " ? " & (Left(tabIncidentsLies(i), 2) >= anneeDebut)
        Debug.Print Left(tabIncidentsLies(i), 2) & " <= " & anneeFin & " ? " & (Left(tabIncidentsLies(i), 2) <= anneeFin)
        Debug.Print Mid(tabIncidentsLies(i), 3, 2) & " >= " & Format(moisDebut, "00") & " ? " & (Mid(tabIncidentsLies(i), 3, 2) >= Format(moisDebut, "00"))
        Debug.Print Mid(tabIncidentsLies(i), 3, 2) & " <= " & Format(moisfin, "00") & " ? " & (Mid(tabIncidentsLies(i), 3, 2) <= Format(moisfin, "00"))
    End If
```

- Ensuite, nous devons tester si le mois et l'année de l'incident se trouvent dans le bon intervalle (date).
- Puis, on affiche un message à l'utilisateur en lui précisant que les incidents sélectionné se trouvent pas dans notre période d'observation.

(Pour ne pas se retrouver avec les incidents qui ne sont compris dans la période d'observation)

```

If Left(tabIncidentsLies(i), 2) >= anneeDebut _
    And Left(tabIncidentsLies(i), 2) <= anneeFin _
    And Mid(tabIncidentsLies(i), 3, 2) >= Format(moisDebut, "00") _
    And Mid(tabIncidentsLies(i), 3, 2) <= Format(moisfin, "00") Then
    ' si oui, on garde
    nbOccurrences = nbOccurrences + 1
    Debug.Print tabIncidentsLies(i) & " ==> OK"
Else
    ' si non, on ne fait rien
    Debug.Print tabIncidentsLies(i) & " ==> pas bon, " & Mid(tabIncidentsLies(i), 3, 2) & "/" & Left(tabIncidentsLies(i), 2) & " n'est pas dans l'intervalle défini."
End If

```

Left permet de récupérer le début d'une chaîne.

Exemple **Left** ("toto", 2) récupère "to"


Mid permet de récupérer une longueur de chaîne à partir d'un indice.

Exemple **Mid** ("toto", 2, 2) récupère "ot"

Format quant à lui donne une forme définie à une valeur.

Il renvoie une chaîne de caractères en fonction d'une date ou d'un nombre dans le format spécifié.

Exemple **Format** (7, "000") renvoie "007"



En gros on lui dit complète ce nombre avec des zéros pour obtenir une chaîne à 3 chiffres

Si l'incident se trouve dans notre période d'observation alors on garde et on incrémente le compteur de **nbOccurrences** pour les compter, ensuite la boucle continue et nous passons au suivant.

Pour finir, on retourne le nombre d'incidents trouvés sur la totalité à la fonction appelante (ChargerAnomalie).

```
        If tracesCompletes Then Debug.Print " "
Next i                                     ' fin de la boucle, on incrémente le compteur i (indice) et on recommence
Debug.Print Chr(13) & nbOccurrences & " incidents trouvés sur " & UBound(tabIncidentsLies) + 1
End Sub
```




Conclusion

- Une fois la fonction terminée, nous devons mettre en place les appels des fonctions. Nous avons ajouté des commentaires au code qui nous permettent d'abord de mieux lire le code, mieux le comprendre et aider le prochain développeur (lui apporter une facilité de lecture).
- Pour conclure, nous avons créé plusieurs Fonctions/Procédures pour mener à bien le fonctionnement de la fiche de suivi, Nous avons découvert de nouvelles notions qui ont toutes un rôle bien important dans notre projet. Une fois les notions découvertes nous avons effectué le travail que nous venons de réaliser pour mener à bien notre projet.

