
EXCEL VBA 講座

sheet 振り分け

【目次】

sheets の中身をデータによって分類する.....	3
全行ループを作る	3
各市のシートの行数を覚えておく連想配列を作る	3
各シートにオートフィルタを掛ける	4
変更が目に見えると遅くなる	5
これらの処理をボタンから呼び出す	5
現在のシートを印刷する	6

sheets の中身をデータによって分類する

前回の練習で csv を sheet1 に読み込むところまで出来ました。今度は市ごとに Sheet に分解してコピーしてみましょう。またデータの先頭に項目名を追加してあります。

全行ループを作る

Sheet1 に読み込んだ csv の全行数を数えてみましょう。

```
'Sheet1 の行を数える
```

```
lastR = Worksheets(1).Range("A1048576").End(xlUp).Row
```

```
'全行読み込み
```

```
For i = 2 To lastR
```

Excel の 1 番下の行から上をめぐって最初にデータがある行までさかのぼったところが最終行です。ループでは 2 行目から最終行までループします。

各市のシートの行数を覚えておく連想配列を作る

上の方法を行えば各シートの最終行を探ることができるのですが、それを毎回すると遅くなります。今回は Scripting.Dictionary という連想配列を使ってみましょう。ここに各市の行数を保存します。

```
Set sd = CreateObject("Scripting.Dictionary") '連想配列  
sd.add "ABC",1 'ABC という配列を作り、1 というデータを保存する  
i = sd("ABC") 'ABC という配列の中身を変数 i に取得する  
if sd("ABC").Exists 'ABC という配列が存在するか？
```

では、この機能を使ってすでに発見した市なのか調べ、あるなら行番号を取得するようにしましょう。

```
cityName = Worksheets(1).Cells(i, 9).Value  
If Not sd.Exists(cityName) Then
```

9 列目にこのデータの市名が入っています。これを cityName に取得して、それが sd の中にあるかどうか確認しましょう。そして存在しなければシートを追加して項目を 1 行目にいれて、内容をコピーしてから、行数を 2 で登録しましょう。

'市が無ければ sheet を作成

```
Worksheets.Add after:=Worksheets(Worksheets.Count)
Worksheets(Worksheets.Count).Name = cityName
Worksheets(1).Range("A1:P1").Copy
Worksheets(cityName).Range("A1:P1").PasteSpecial
sd.Add cityName, 2 'シート名と行番号を初期化
End If
```

これで初めて出てきた市の分の初期化はできました。そして読み込んだ行のそのシートへのコピーをして、行のカウントアップします。ついでに各シートでの連番に変えておきます。

```
Worksheets(1).Range("A" & i & ":P" & i).Copy
Worksheets(cityName).Range("A" & sd(cityName) & ":P" &
sd(cityName)).PasteSpecial
Worksheets(cityName).Cells(sd(cityName), 1).Value = sd(cityName) - 1
sd(cityName) = sd(cityName) + 1 '行数カウントアップ
Next
```

Range で範囲をコピーして、同じく Range でコピー先を指定して貼り付けます。この処理を「dispatchSheets」関数として保存します。

各シートにオートフィルタを掛ける

すべてのシートにオートフィルタをつけるのも作ってみましょう。

シートのは数は Worksheets.Count で分かります。

```
Sub setFilter()
  For i = 4 To Worksheets.Count
    Worksheets(i).Activate
    ActiveSheet.Columns("A:P").Select
    Selection.AutoFilter
    ActiveSheet.Range("A1").Select
  Next
End Sub
```

すべてのシートをアクティブにしながら、カラムを選択して、AutoFilter を掛けます。そのままだと全体が選択された状態になるので仮に A1 を選択して selection を外しておきます。

変更が目に見えると遅くなる

ところで、こんな作業をさせると、めまぐるしく表示が変わります。実はこれをする Excel が頑張っで見せようとするので、その分処理が遅くなってしまいます。何千行も読むときには気になります。そこで表示だけ止めておいて、処理だけすすめる方法があります。

```
Application.ScreenUpdating = False
    「遅い処理」
Application.ScreenUpdating = True
```

気になるときは入れてみましょう。

これらの処理をボタンから呼び出す

いままで作った関数のまとめです

```
Sub dispatchSheets()
    Dim lastR As Integer
    Dim i As Integer
    Dim cityName As String

    Set sd = CreateObject("Scripting.Dictionary") '連想配列

    Application.ScreenUpdating = False

    'Sheet1 の行を数える
    lastR = Worksheets(1).Range("A1048576").End(xlUp).Row

    '全行読み込み
    For i = 2 To lastR
        cityName = Worksheets(1).Cells(i, 9).Value
        If Not sd.Exists(cityName) Then
            '市が無ければ sheet を作成
            Worksheets.Add after:=Worksheets(Worksheets.Count)
            Worksheets(Worksheets.Count).Name = cityName
        End If
    Next i
End Sub
```

```

Worksheets(1).Range("A1:P1").Copy
Worksheets(cityName).Range("A1:P1").PasteSpecial
sd.Add cityName, 2 'シート名と行番号を初期化
End If
Worksheets(1).Range("A" & i & ":P" & i).Copy
Worksheets(cityName).Range("A" & sd(cityName) & ":P" &
sd(cityName)).PasteSpecial
Worksheets(cityName).Cells(sd(cityName), 1).Value = sd(cityName) - 1
sd(cityName) = sd(cityName) + 1 '行数カウントアップ
Next

Application.ScreenUpdating = True
Set sd = Nothing

End Sub

```

これらの処理をボタンから呼び出します

```

Private Sub CommandButton1_Click()
    Call csvRead
    Call dispatchSheets
    Call setFilter
End Sub

```

これで csv から読み込んで整理するところまでできあがりです。

現在のシートを印刷する

フィルターなどで絞り込んでからそのシートを印刷するながれを作ってみましょう。

やりたいことをマクロ登録して修正します。

全カラムを出すのは無理なので、今回は連番からメールアドレスまで出します。

また印刷時の各シートの先頭行に関しては、ページレイアウトの印刷タイトルで設定しておきます。

手順は

1. メールアドレス以降のカラムを非表示にする
2. A 1 から連続の一番下まで選択する

3. そこから右に連続の右端まで選択する
4. AからFのカラムをオートフィットする
5. 選択範囲を印刷する

```
Sub setPrintArea()  
    Columns("G:P").Select  
    Selection.EntireColumn.Hidden = True  
    Range("A1").Select  
    Range(Selection, Selection.End(xlDown)).Select  
    Range(Selection, Selection.End(xlToRight)).Select  
    Columns("A:F").EntireColumn.AutoFit  
    'ActiveWindow.SelectedSheets.PrintOut Copies:=1, Collate:=True, _  
    '    IgnorePrintAreas:=False  
End Sub
```

このマクロをマクロ一覧から呼び出して印刷します。ただ、実際にそのまま印刷してもその時の文字長などによってうまくいかない場合がありますので、印刷だけは手動でやったほうがいいでしょう。今はコメントにしてあります。

印刷するカラムなどを変更したい場合は、マクロを修正してください。
また、全シート印刷したい場合は、オートフィルタでやったようにループして印刷してください。