

Excel マクロ入門

No.2

事務職員協会パソコン研修会 2018

KCS 鹿児島情報専門学校

目次

実践.....	2
シート管理	2
ファイル一覧取得	9
繰返印刷処理.....	12
祝日の取得	16
WebAPI の利用	25
付録.....	29
セル内データのソート	29
Excel 標準処理を使用する	37
ドリル作成	41
座席表作成	61

実践

VBA の基本について説明してきました。

ここから Excel 標準関数や操作にない処理をマクロで作成してみます。

シート管理

作業しているワークブックのシート数が多いときなどシートの移動が大変な場合があります。

見出しの様なシートを 1 枚作成して移動などマクロ化します。

今回作成するサンプルは、ワークブックを開くとき、シートを切り替えるとき、ワークシートをダブルクリックしたときのイベントを利用します。

ワークブックを開くとき

Workbook_Open

' ワークブックを開く時のイベント

Private Sub Workbook_Open()

 ActiveWindow.ScrollRow = 1

 MsgBox "各シートの" & TRGETMARK & _

 "セルをダブルクリックすると目次シートがアクティブになります。"

 Call get_SheetName

End Sub

シートを切り替えるとき

Worksheet_Activate

' シートを切り替えるイベント

Private Sub Workbook_SheetActivate(ByVal Sh As Object)

 MsgBox "Workbook_SheetActivate イベントが発生しました。" & vbCr & _

 "シート名は" & Sh.Name & "です。"

End Sub

ワークシートをダブルクリックしたとき

既定のダブルクリックの操作の前に、ワークシートをダブルクリックしたときに発生します。

Workbook_SheetBeforeDoubleClick

‘ダブルクリックイベント

Private Sub Workbook_SheetBeforeDoubleClick(ByVal Sh As Object, _

 ByVal Target As Range, ByVal Cancel As Boolean)

 If MsgBox("Workbook_SheetBeforeDoubleClick イベントが発生しました。" & vbCr & _

 "キャンセルしますか?", vbInformation + vbYesNo) = vbYes Then

```
Cancel = True
```

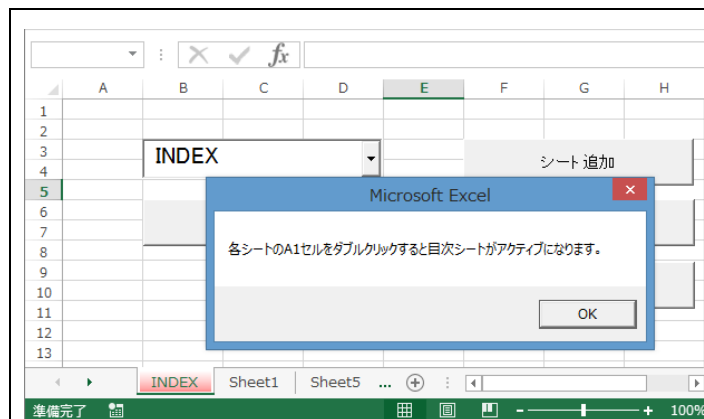
```
End If
```

```
End Sub
```

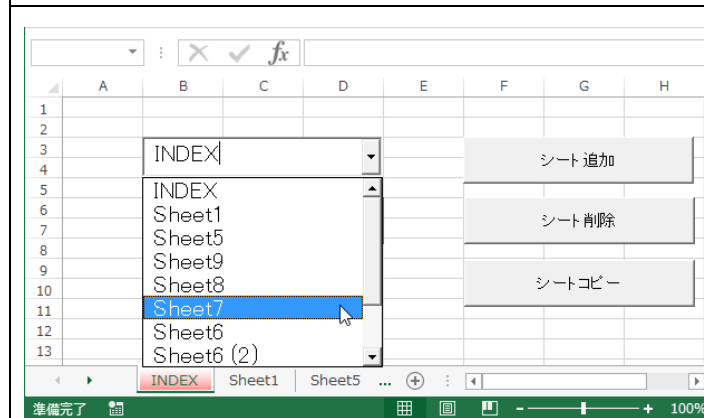
なお上記のプロシーダは「ThisWorkbook」モジュール内に記述します。

今までは標準モジュールに記述していました。

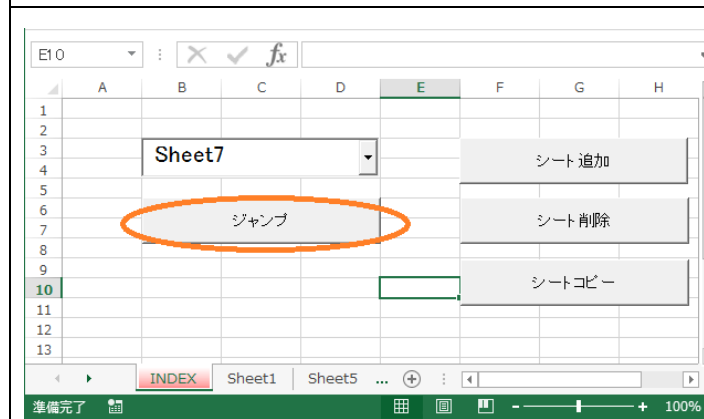
実行例) ブックイベント.xlsm



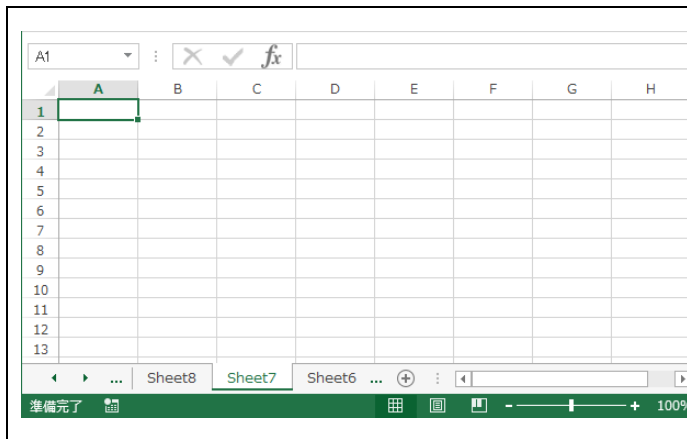
ワークブックを開くとメッセージを表示します。
(ここでシートの一覧を取得・更新します)



コンボボックスをクリックしてシートのリストを表示し、目的のシート名を選択します。



ジャンプボタンをクリックします。



シート名「Sheet7」シートに移動しました。
A1 セルをダブルクリックすると目次シートに移動します。

マクロ

標準モジュール

```
Option Explicit
'目次シート名の宣言
Global Const INDEXSHEET As String = "INDEX"
'ワークブックマクロのマーカー位置
Global Const TRGETMARK As String = "A1"
'
'#-----
'ワークシートを追加する
'@parm    : none
'@return : none
'Version : 0.1
'#-----
'
Sub Sheet_Add()
    Dim strName As String
    '入力ボックスを表示し、文字列の入力を受け付ける
    strName = InputBox("シート名を入力してください.", "ワークシート追加")
    'キャンセルボタンの押下は StrPtr 関数を用いるゼロであればキャンセル
    If StrPtr(strName) <> 0 Then
        Worksheets.Add After:=Worksheets(Worksheets.Count)
        '入力ボックスで文字列の入力があればシート名を設定し、未入力であればデフォルト値を使用
        If Len(strName) <> 0 Then
            Worksheets(Worksheets.Count).Name = strName
        End If
    End If
    '目次シートをアクティブにする
```

```

    Call Sheet_Index
    Call get_SheetName
End Sub
'
'#-----
'ワークシートを削除する
'@parm    : none
'@return : none
'Version : 0.1
'#-----
'
Sub Sheet_Del()
    Dim wrksht As Worksheet
    Dim okno As VbMsgBoxResult
    'コンボボックス変数の定義
    Dim comboBox As MSForms.comboBox
    'オブジェクト"コンボボックス"を変数に設定する
    Set comboBox = Sheets(INDEXSHEET).ComboBox1
    okno = MsgBox(comboBox.Value + "を削除しますか?", vbOKCancel, "ワークシートの削除")
    If okno = vbOK Then
        If comboBox.Value = INDEXSHEET Then
            MsgBox ("このシートは削除できません!")
        Else
            Application.DisplayAlerts = False
            Worksheets(comboBox.Value).Delete
            Call get_SheetName
            Application.DisplayAlerts = True
        End If
    End If
End Sub
'
'#-----
'ワークシートを複写する
'@parm    : none
'@return : none
'Version : 0.1
'#-----
'

```

```

Sub Sheet_copy()
    Dim wrksht As Worksheet
    Dim okno As VbMsgBoxResult
    'コンボボックス変数の定義
    Dim comboBox As MSForms.comboBox
    'オブジェクト"コンボボックス"を変数に設定する
    Set comboBox = Sheets(INDEXSHEET).ComboBox1
    okno = MsgBox(comboBox.Value + "をコピーしますか?", vbOKCancel, "ワークシートのコピー")
    If okno = vbOK Then
        If comboBox.Value = INDEXSHEET Then
            MsgBox ("このシートはコピーできません!")
        Else
            Application.DisplayAlerts = False
            Worksheets(comboBox.Value).Copy After:=Worksheets(comboBox.Value)
            Call get_SheetName
            Application.DisplayAlerts = True
        End If
    End If
End Sub
'
'#-----
'ワークブックのシート名を列挙する
'@parm    : none
'@return : none
'Version : 0.1
'#-----
'

Sub get_SheetName()
    Dim cnt As Integer
    Dim i As Integer
    Dim wrksht As Worksheet
    Dim shtName() As String
    'コンボボックス変数の定義
    Dim cmbobox As MSForms.comboBox
    'オブジェクト"コンボボックス"を変数に設定する
    Set cmbobox = Sheets(INDEXSHEET).ComboBox1
    'コンボボックスのデータをクリアする
    cmbobox.Clear

```

```

'現在アクティブなワークブックのシート数を取得する
cnt = Worksheets.Count
'シート数で配列を再宣言
ReDim shtName(cnt)
i = 0
'ワークシートを1枚ずつ取り出す
For Each wrksht In Worksheets
    'シート名を取得
    shtName(i) = wrksht.Name
    '添え字アップ
    i = i + 1
    'コンボボックスに追加する
    cmbobox.AddItem wrksht.Name
Next
'コンボボックスの最初のデータを選択
cmbobox.ListIndex = 0
End Sub
'
'#-----
'ワークシートをアクティブにする
'@parm    : none
'@return : none
'Version : 0.1
'#-----
'
Sub Sheet_Jump()
    Dim cmbobox As MSForms.comboBox
    'オブジェクト"コンボボックス"を変数に設定する
    Set cmbobox = Sheets(INDEXSHEET).ComboBox1
    'コンボボックスで指定したシートをアクティブにする
    Worksheets(cmbobox.Value).Select
    'アクティブシートの A1 セルをアクティブにする
    Range(TRGETMARK).Activate
End Sub
'
'#-----
'ワークシートを目次をアクティブにする
'@parm    : none

```



```
'@return : none
'Version : 0.1
'#-----
'
Sub Sheet_Index()
    '目次シートをアクティブにする
    Worksheets(INDEXSHEET).Select
    'アクティブシートの A1 セルをアクティブにする
    Range(TRGETMARK).Activate
End Sub
```

ThisWorkbook モジュール

```
' ワークブックを開く時のイベント
Private Sub Workbook_Open()
    ActiveWindow.ScrollRow = 1
    MsgBox "各シートの" & TRGETMARK & "セルをダブルクリックすると目次シートがアクティブになります。"
    Call get_SheetName
End Sub

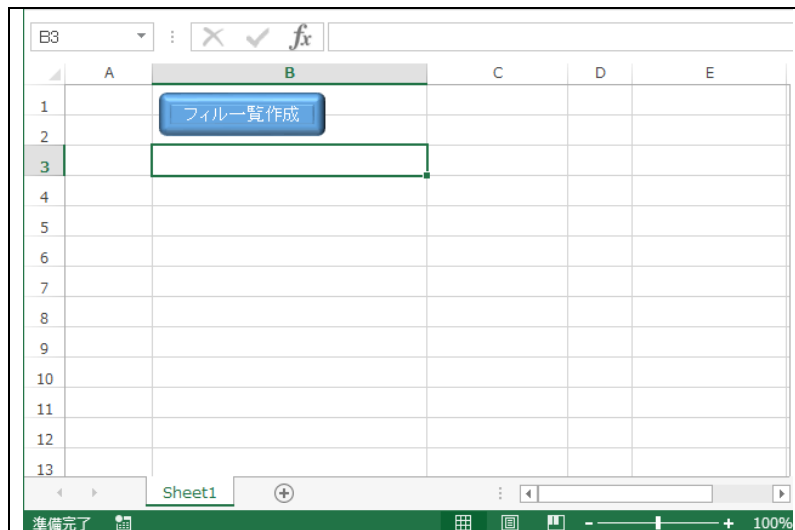
'シートを切り替えるイベント
Private Sub Workbook_SheetActivate(ByVal Sh As Object)
    'MsgBox "Workbook_SheetActivate イベントが発生しました。" & vbCr & _
    '    "シート名は" & Sh.Name & "です。"
    Call get_SheetName
End Sub

'シートの A 1 だったらジャンプする
Private Sub Workbook_SheetBeforeDoubleClick(_
    ByVal Sh As Object, _
    ByVal Target As Range, _
    Cancel As Boolean)
    If Target.Column <> 1 And Target.Row <> 1 Then
        Exit Sub
    End If
    Call Sheet_Index
    Cancel = False
End Sub
```

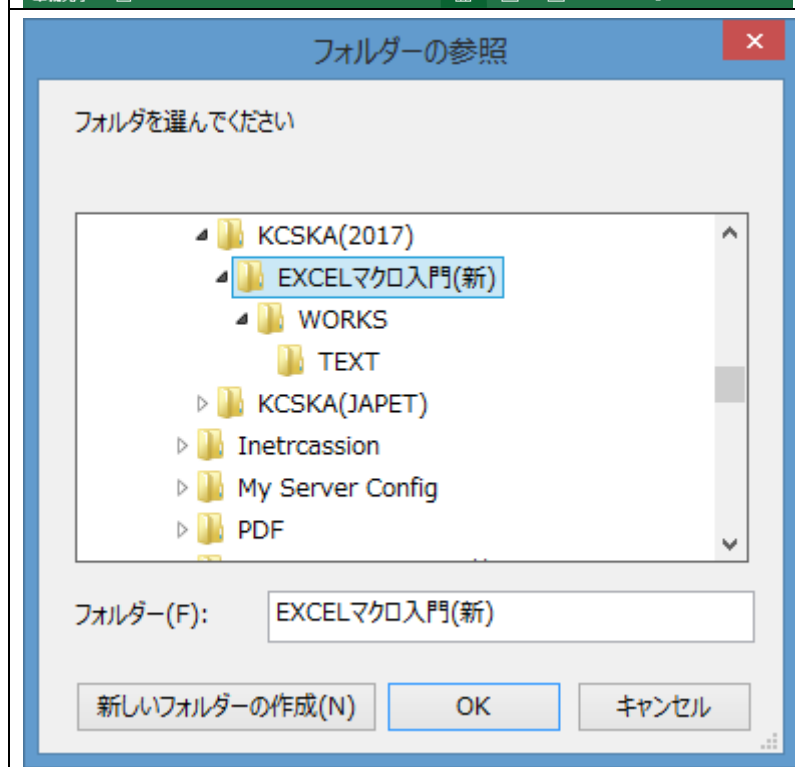
ファイル一覧取得

シート管理はワークブックのみでしたが、今回は指定したフォルダ内のファイル一覧を作成します。
また作成されたフィル名のセルにはハイパーリンク機能で開くようにします。

実行例) ファイル一覧.xlsm



左図では図形にマクロが登録してあります。
図形をクリックします。



フォルダの参照ダイアログが開くので
フォルダを選択し OK ボタンを押下します。
またキャンセルボタンで処理を終了します。

	A	B	C	D	E
1		フィルター作成			
2					
3	ドリル作成.xlsm		D:\96MyCloud\90\	123	Microsoft Excel マ
4	座席表作成.xlsm		D:\96MyCloud\90\	32	Microsoft Excel マ
5	2013_kan_064.xlsx		D:\96MyCloud\90\	13	Microsoft Excel ワ
6	2015成績表.xlsm		D:\96MyCloud\90\	14	Microsoft Excel マ
7	2016成績表.xlsm		D:\96MyCloud\90\	13	Microsoft Excel マ
8	2017成績表.xlsm		D:\96MyCloud\90\	13	Microsoft Excel マ
9	Book1.xls		D:\96MyCloud\90\	37	Microsoft Excel 97-
10	Module1.bas		D:\96MyCloud\90\	0	BAS ファイル
11	ファイル一覧.xlsm		D:\96MyCloud\90\	0	Microsoft Excel マ
12	ウインドウの操作.xlsm		D:\96MyCloud\90\	16	Microsoft Excel マ
13	シート追加取得.xlsm		D:\96MyCloud\90\	33	Microsoft Excel マ

選択されたフォルダ内のファイル一覧をアクティブなシートに作成します。
ファイル名のセルにはハイパーリンクが設定してあります。クリックするとファイルを開きます。

マクロ

標準モジュール

Option Explicit

```
'#-----
'指定されたパスの一覧を取得(再起呼び出し)
'@param   : strPath
'@param   : i
'@return  : none
'Version : 0.1
'#-----
```

Sub GetFiles(strPath As String, i As Integer)

```
Dim strShtName As Worksheet
Dim objFs As Object
Dim objFld As Object
Dim objFl As Object
Dim objSub As Object
'何かエラーが発生したら ジャンプする
On Error GoTo End_Step:
Set objFs = CreateObject("Scripting.FileSystemObject")
Set objFld = objFs.GetFolder(strPath)
Set strShtName = ActiveSheet
Application.StatusBar = strPath
For Each objFl In objFld.Files
    'フルパスからファイル名のみ取得
    strShtName.Cells(i, 2) = objFs.GetFileName(objFl.Path)
    '親フォルダのパスを取得
```

```

strShtName.Cells(i, 3) = objFl.ParentFolder.Path
'ファイルサイズの計算
strShtName.Cells(i, 4) = Int(objFl.Size / 1024)
'ファイルの種類を取得
strShtName.Cells(i, 5) = objFl.Type
'最後の更新日付を取得
strShtName.Cells(i, 6) = objFl.DateLastModified
'ファイル名にフルパスのハイパーリンクを設定
'strShtName.Hyperlinks.Add Anchor:=Cells(i, 2), _
    Address:=objFl.Path
'ファイル名にフルパスのハイパーリンクを設定(ヒント、アンカーテキストも設定)
strShtName.Hyperlinks.Add Anchor:=Cells(i, 2), _
    Address:=objFl.Path, _
    ScreenTip:=objFl.Type, _
    TextToDisplay:=objFs.GetFileName(objFl.Path)
i = i + 1
Next
For Each objSub In objFld.SubFolders
    Call GetFiles(objSub.Path, i)
DoEvents
Next
'GOTO 文のラベル
End_Step:
    Application.StatusBar = False
End Sub
'#-----
'フォルダダイアログを表示し、指定したフォルダ内のファイル一覧を取得する
'@parm    : none
'@return : none
'Version : 0.1
'#-----
Sub Get_FileList()
    Dim Shell As Object
    Dim myPath As Object
    Dim strPath As String
    Dim i As Integer
    '開始行番号指定
    i = 3

```

```

'シェルのオブジェクトを作成する
Set Shell = CreateObject("Shell.Application")
'フォルダ選択ダイアログを表示
Set myPath = Shell.BrowseForFolder(&O0, "フォルダを選んでください", &H1 + &H10, "")
'キャンセルだったら終了
If myPath Is Nothing Then
    Exit Sub
End If
'選択されたフォルダのフルパスを取得
strPath = myPath.Items.Item.Path
'アクティブシートのハイパーリンクを全て削除
ActiveSheet.Hyperlinks.Delete
'アクティブシートの B3 セル以降をクリアする
Range("B" & i, ActiveCell.SpecialCells(xlLastCell)).ClearContents
'画面の描画を抑制する
Application.ScreenUpdating = False
Call GetFiles(strPath, i)
'画面の描画を抑制解除する
Application.ScreenUpdating = True
End Sub

```

繰返印刷処理

成績など管理している際に成績管理表と、個人に配布する個人成績表があり、個人成績表の印刷をマクロ化します。下記のイメージを想定します。

例) 成績表.xlsm

The left screenshot shows a table with the following data:

出席番号	氏名	国語総合S	国語総合G	国語総合R	現代文S	現代文G	現代
1	大津しおり	90	優	9	99	優	1
2	瀬戸 勇	97	優	4	97	優	3
3	大坂 ひと	100	優	1	61	可	3
4	堂原 晋也	70	良	24	82	優	15
5	高柳 美希	81	優	18	70	良	2
6	二宮 遥	84	優	5	86	優	9
7	三村 直人	88	可	26	75	良	2
8	大川 めぐみ	60	可	38	84	優	15
9	溝口 亜希	82	優	16	64	可	3
10	竹内 れいな	67	可	30	69	可	3
11	滝沢 隆太	68	可	28	69	可	3
12	橋口 貴雄	66	可	31	86	優	9
13	栗山 直人	84	可	34	91	優	5
14	関 翔子	80	優	19	68	可	3
15	下川 きみまろ	65	可	32	82	優	15

The right screenshot shows the '成績表' (Performance Table) form. It includes fields for 'クラス' (Class) set to '1年1組', '出席番号' (Attendance Number) set to '1番', and '氏名' (Name) set to '大津しおり'. Below these fields is a table of subjects and scores:

	科目名	点数	評価	クラス順位
1	国語総合	90	優	9位 / 40人
2	現代文	99	優	1位 / 40人

成績表には既に印刷範囲が設定してあり、出席番号欄に番号を入力すると該当学生の成績表が作成される。

Excel の標準関数の VLOOKUP など既に設定しています。今回は番号入力と印刷を繰返処理を用いてマクロ化します。

マクロ

標準モジュール

Option Explicit

'#-----

'印刷したい学生番号の指定、印刷先の指定

'@parm : NONE

'@return : NONE

'Version : 0.1

'Comment :

'#-----

Sub 印刷()

'使用するデータの変数を宣言するよ。

Dim fileName As String 'PDF 保存先フォルダパス&ファイル名

Dim SNO As String 'INPUTBOX 関数の戻り値は String 型 開始番号

Dim ENO As String 'INPUTBOX 関数の戻り値は String 型 終了番号

Dim PRT As Variant 'INPUTBOX メソッドの戻り値は Variant 型

Dim wPrint As String 'メッセージ用

Dim I As Integer '繰り返し処理用添え字

'順次構造(直線型)でいくよ。

'印刷したい学生キーを入力するよ。どこからどこまで。

'まずは開始番号だね。全員の場合は最初の番号を。

SNO = InputBox("開始 No.を入力")

'開始番号で未入力またはキャンセルボタンを押下したら何もしないよ。

'InputBox 関数はキャンセルボタンを押下すると戻り値が NULL ""になる

If SNO = "" Then

MsgBox "未入力または入力キャンセルされました。", vbExclamation

'これ以降処理しないようにプロシージャを終了するよ。エラー処理

Exit Sub

End If

'そして終了番号だね。全員の場合は最後の番号を。

ENO = InputBox("終了 No.を入力")

'終了番号で未入力またはキャンセルボタンを押下したら何もしないよ。

If ENO = "" Then

MsgBox "未入力または入力キャンセルされました。", vbExclamation

Exit Sub

End If

'InputBox メソッドはキャンセルボタンを押下すると false Booleang 型になる

```

PRT = Application.InputBox("P または F を指定してください。" _
    + vbLf + "P:プリンタ F:PDF" + vbLf, "出力先指定", Type:=2)
'出力先指定でキャンセルボタンを押下したら何もしないよ。
'戻り値が Boolean 型であればキャンセルボタン押下
If TypeName(PRT) = "Boolean" Then
    MsgBox "入力がキャンセルされました。", vbExclamation
    'ラベルにジャンプするよ。
    GoTo exitsub
End If
'全角文字を半角文字に変換
PRT = StrConv(PRT, vbNarrow)
'画面のちらつきを抑制する場合（画面の更新を止める）
Application.ScreenUpdating = False
'繰り返し処理用 SNO <= ENO となるようにチェック&入替
Dim swap As Integer    '交換用作業エリア
If SNO > ENO Then
    swap = SNO
    SNO = ENO
    ENO = swap
End If
'入力された開始番号から終了番号まで繰り返す
'反復構造(繰返し型)だね。
For I = SNO To ENO
    '個人成績表の番号欄に代入する
    '同様なシートが作成済みで印刷マクロ未実装の場合はここを改修すれば実装可能。
    Worksheets("印刷").Range("C5") = I
    '出力先の判定
    'Select Case 文で大文字小文字の判定を簡素化するために今回大文字に変換する Ucase()
    'よって大文字だけで判定できるよ。
    '選択構造(分岐型)だね。
    Select Case UCase(PRT)
        'デフォルトプリンタへの出力
        Case "P"
            '直印刷する場合
            ActiveWindow.SelectedSheets.PrintOut Copies:=1, Collate:=True
            '完了メッセージ用だよ。
            wPrint = "プリンタ"

```

'PDF への出力

Case "F"

'PDF で出力するために、パス&ファイル名を作成する

'保存先は Excel ファイルがある場所だよ。

'ThisWorkbook.Path で取得できるんだ。

'フォルダの区切りは¥を使うよ。

'ファイル名は JAPET_番号_サンプル.pdf にしたよ。

'TEXT()関数を使用する

'例) excle 関数は Application.WorksheetFunction を付加するよ。

fileName = ThisWorkbook.Path & _

"¥JAPET_" & Application.WorksheetFunction.Text(I, "00") & _
"_サンプル.pdf"

'TEXT()関数と同等の Format()関数を使用する例)

'fileName = ThisWorkbook.Path & "¥JAPET_" & _

Format(I, "00") & "_サンプル.pdf"

'PDF ファイルとして保存する場合

ActiveSheet.ExportAsFixedFormat Type:= _

xlTypePDF, fileName:=fileName

'完了メッセージ用だよ。

wPrint = "PDF"

'想定外の文字を押されたよ。

Case Else

MsgBox "印刷先指定文字 【" + PRT + "】 が無効です。" + vbLf + _
"キャンセルします。"

Exit Sub

End Select

Next I

'画面のちらつきを抑制解除（画面の更新を行う）

Application.ScreenUpdating = True

MsgBox SNO + "番号から" + ENO + "番まで" + wPrint + "に完了"

'ラベルを付加 VB ではラベルにジャンプ GoTo できる。処理の流れを変えることができる。

'ただし!多用すると処理の流れが複雑なるので、極力上から下へのジャンプに止める。

exitsub:

End Sub

祝日の取得

予定表や出席簿などでカレンダーを利用する機会が多いかと思います。曜日の取得は標準の機能で取得できますが、祝日はできません。あらかじめ手入力で祝日一覧を作成して利用する方法もありますが、今回祝日一覧をマクロ化してみます。下記のイメージです。

例) 祝日取得.xlsm

G5

×

✓

f_x

	A	B	C	D	E	F	G	H
1								
2								
3								
4			2017					
5			日付	曜日	祝日			
6		1	2017/04/29	土	昭和の日			
7		2	2017/05/03	水	憲法記念日			
8		3	2017/05/04	木	みどりの日			
9		4	2017/05/05	金	こどもの日			
10		5	2017/07/17	月	海の日			
11		6	2017/08/11	金	山の日			
12		7	2017/09/18	月	敬老の日			
13		8	2017/09/23	土	秋分の日			
14		9	2017/10/09	月	体育の日			
15		10	2017/11/03	金	文化の日			
16		11	2017/11/23	木	勤労感謝の日			
17		12	2017/12/23	土	天皇誕生日			
18		13	2018/01/01	月	元旦			
19		14	2018/01/08	月	成人の日			
20		15	2018/02/11	日	建国記念の日			
21		16	2018/02/12	月	振替休日			
22		17	2018/03/21	水	春分の日			
23		18						
24		19						
25		20						
		21						

祝日取得

※C5:E34の範囲に祝日と言う名

◀ ▶

Sheet1

祝日

Sheet3

+

準備完了

📊

📈

📉

📌

🔍

🔧

利用イメージ

The figure consists of two side-by-side screenshots of a spreadsheet application. The left screenshot shows a calendar for the year 2017. The calendar is displayed in a grid format with columns for each day of the week (C to X) and rows for each month (1 to 12). The dates 1 through 31 are visible, with some dates highlighted in red (e.g., 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31). The right screenshot shows a similar calendar for 2017, but with a table below it. The table has columns for the day of the week (A to F) and rows for each date (1 to 31). The table contains the day of the week for each date, such as (月) for Monday, (火) for Tuesday, (水) for Wednesday, (木) for Thursday, (金) for Friday, (土) for Saturday, and (日) for Sunday. The table also includes a header row for the month and year (2017年) and a title (月日程表).

マクロ

```
'
'http://memo4kiyo.ti-da.net/e2508363.html 出典
'*****

' 祝日チェック関数 Ver.0.1
'*****

Public Function check_date(chkDate As Date, hld_nam As String) As Boolean
    Dim Res As Boolean
    Dim pre_Date As Date
    Dim pre_Day As Integer
    Dim tmp_Date As Integer
    Dim tmp_Day As Integer
    Dim tmp_y, tmp_m, tmp_d As Integer
    Dim tmp_ymd As Date
    Dim chk_flg, flg As Boolean
    '初期化
    Res = False
    '祝日のチェック
    chk_flg = chk_shukujitu(chkDate, hld_nam)
    tmp_Day = Weekday(chkDate)
    '当日が祝日の場合
    If chk_flg = True Then
        Res = True
    Else
        '当日が祝日でない場合
        '前日の日曜日で、祝日ならば、振替休日とする。
        '前日を取得
        pre_Date = DateAdd("d", -1, chkDate)
        tmp_y = CInt(Format(pre_Date, "yyyy"))
        tmp_m = CInt(Format(pre_Date, "m"))
        tmp_d = CInt(Format(pre_Date, "d"))
        '前日が祝日かをチェック
        flg = False
        flg = chk_shukujitu(pre_Date, hld_nam)
        '前日の曜日をチェック
        pre_Day = Weekday(pre_Date)
        If CInt(Format(chkDate, "m")) = 5 Then
            If CInt(Format(pre_Date, "d")) = 5 Then
```

```

        Select Case Weekday(DateValue(CStr(tmp_y) & "/5/6"))
            Case 3
                Res = True
                hld_nam = "振替休日"
            Case 4
                Res = True
                hld_nam = "振替休日"
            Case Else
                If pre_Day = 1 And flg = True Then
                    Res = True
                    hld_nam = "振替休日"
                End If
            End Select
        End If
    Else
        If pre_Day = 1 And flg = True Then
            Res = True
            hld_nam = "振替休日"
        End If
    End If
End If
check_date = Res
End Function

'*****
' 第○曜日取得関数 Ver.0.1
'*****

'y   : 年
'm   : 月
'n   : n 番目曜日
'nd  : 曜日(1:日曜日,1:月曜日～7:土曜日)
'[戻り値]
' 0 : 該当無し,1～31 : 指定月の第 n 番目の nd 曜日の日付
Public Function get_no_day(y, m, n, nd As Integer) As Integer
    Dim GetCnt As Integer
    Dim GetTmpDate As Date
    Dim Res As Integer
    Dim lastMonth As Date
    Dim i, x As Integer

```

```

'初期値
GetCnt = 0
Res = 0
'指定日の末日を取得
lastMonth = DateAdd("m", 1, DateValue(CStr(y) & "/" & CStr(m) & "/1"))
lastMonth = DateAdd("d", -1, lastMonth)
x = CInt(Format(lastMonth, "d"))
For i = 1 To x
    If Weekday(DateValue(CStr(y) & "/" & CStr(m) & "/" & CStr(i))) = nd Then
        GetCnt = GetCnt + 1
    End If
    If GetCnt = n Then
        Res = i
        Exit For
    End If
Next
'結果を返す
get_no_day = Res
End Function

```

```

'*****

```

```

' 祝日チェック関数 Ver.0.1

```

```

'*****

```

```

Public Function chk_shukujitu(chkDate As Date, hld_nam As String) As Boolean
    Dim chkYear As Integer
    Dim chkMonth As Integer
    Dim chkDay As Integer
    Dim IngAged As Integer
    Dim IngVernal As Integer
    Dim IngAutumnal As Integer
    Dim tmpDate As Date
    Dim cnt As Integer
    Dim Res As Boolean
    ' 初期値
    Res = False
    chkYear = CInt(Format(chkDate, "yyyy"))
    chkMonth = CInt(Format(chkDate, "m"))
    chkDay = CInt(Format(chkDate, "d"))

```

'月毎に祝日のチェックを行う

Select Case chkMonth

Case 1 ' 1 月

' 元旦

If chkYear >= 1949 Then

If chkDay = 1 Then

Res = True

hld_nam = "元旦"

End If

End If

'成人の日

If chkYear >= 1949 And chkYear < 2000 Then

If chkDay = 15 Then

Res = True

hld_nam = "成人の日"

End If

ElseIf chkYear >= 2000 Then

'第二月曜日と一致するか？

If get_no_day(chkYear, 1, 2, 2) = chkDay Then

Res = True

hld_nam = "成人の日"

End If

End If

Case 2 ' 2 月

'建国記念日

If chkYear >= 1967 Then

If chkDay = 11 Then

Res = True

hld_nam = "建国記念の日"

End If

End If

Case 3 ' 3 月

' 春分の日(2099 年まで有効)

If chkYear >= 1949 Then

IngVernal = Int(20.69115 + 0.2421904 * (chkYear - 2000) - Int((chkYear - 2000) / 4))

If CInt(IngVernal) = chkDay Then

Res = True

hld_nam = "春分の日"

```
End If
End If
Case 4 ' 4 月
' 天皇誕生日(1949～1988)/みどりの日(1989～2006)/昭和の日(2007～)
If chkDay = 29 Then
    If chkYear >= 1949 And chkYear < 1989 Then
        Res = True
        hld_nam = "天皇誕生日"
    ElseIf chkYear >= 1989 And chkYear < 2007 Then
        Res = True
        hld_nam = "みどりの日"
    ElseIf chkYear >= 2007 Then
        Res = True
        hld_nam = "昭和の日"
    End If
End If
Case 5 ' 5 月
' 憲法記念日／こどもの日
If chkYear >= 1949 Then
    If chkDay = 3 Then
        Res = True
        hld_nam = "憲法記念日"
    ElseIf chkDay = 5 Then
        Res = True
        hld_nam = "こどもの日"
    End If
End If
' みどりの日
If chkDay = 4 Then
    If chkYear >= 1986 Then
        Res = True
        hld_nam = "みどりの日"
    End If
End If
Case 7 ' 7 月
' 海の日(7 月第 3 月曜日)
If chkYear >= 1996 And chkYear < 2003 Then
    If chkDay = 20 Then
```

```

        Res = True
        hld_nam = "海の日"
    End If
ElseIf chkYear >= 2003 Then
    If get_no_day(chkYear, 7, 3, 2) = chkDay Then
        Res = True
        hld_nam = "海の日"
    End If
End If
Case 8 ' 8 月
    ' 山の日
    If chkYear >= 2016 Then
        If chkDay = 11 Then
            Res = True
            hld_nam = "山の日"
        End If
    End If
Case 9 ' 9 月
    ' 敬老の日
    If chkYear >= 1966 And chkYear < 2003 Then
        If chkDay = 15 Then
            Res = True
            hld_nam = "敬老の日"
        End If
    ElseIf chkYear >= 2003 Then
        IngAged = get_no_day(chkYear, 9, 3, 2)
        If IngAged = chkDay Then
            Res = True
            hld_nam = "敬老の日"
        End If
    End If
    ' 秋分の日
    If chkYear >= 1948 Then
        IngAutumnal = Int(23.09 + 0.2421904 * (chkYear - 2000) - Int((chkYear - 2000) / 4))
        If CInt(IngAutumnal) = chkDay Then
            Res = True
            hld_nam = "秋分の日"
        End If
    End If

```

```

End If
' 国民の休日(敬老の日と秋分の日に挟まれると休み)
If chkYear >= 2003 Then
    '秋分の日が水曜日なら、敬老の日(月曜日)と挟まれるから火曜日は国民の休日となる
    If Weekday(DateValue(CStr(chkYear) & "/9/" & CStr(IngAutumnal))) = 4 Then
        If DateValue(CStr(chkYear) & "/9/" & CStr(IngAged + 1)) =
DateValue(CStr(chkYear) & "/9/" & CStr(chkDay)) Then
            Res = True
            hld_nam = "国民の休日"
        End If
    End If
End If
Case 10 ' 10 月
    ' 体育の日
    If chkYear >= 1966 And chkYear < 2000 Then
        If chkDay = 10 Then
            Res = True
            hld_nam = "体育の日"
        End If
    ElseIf chkYear >= 2000 Then
        If get_no_day(chkYear, 10, 2, 2) = chkDay Then
            Res = True
            hld_nam = "体育の日"
        End If
    End If
Case 11 ' 11 月
    ' 文化の日／勤労感謝の日
    If chkYear >= 1948 Then
        If chkDay = 3 Then
            Res = True
            hld_nam = "文化の日"
        ElseIf chkDay = 23 Then
            Res = True
            hld_nam = "勤労感謝の日"
        End If
    End If
Case 12 ' 12 月
    '天皇誕生日(1989～)

```



```

        If chkYear >= 1989 Then
            If chkDay = 23 Then
                Res = True
                hld_nam = "天皇誕生日"
            End If
        End If
    End Select
    chk_shukujitu = Res
End Function

```

check_date Function プロシージャがメインになります。これは Web 上で検索して見つけたものです。

<http://memo4kiyo.ti-da.net/e2508363.html>

これを利用してマクロを自作しました。

```

Option Explicit
Const TARGET_SHEET As String = "祝日"
Const TARGET_YEAR As String = "C3"
Const TARGET_CELL As String = "C5"
Sub get_Date()
    Dim trgSheet As Worksheet
    Dim srcSheet As Worksheet
    Dim trgYear As Integer
    Dim trgCell As Range
    '実行前のアクティブなシートを保存
    Set srcSheet = ActiveSheet
    '祝日シートを定義
    Set trgSheet = Worksheets(TARGET_SHEET)
    '出力開始セルの定義
    Set trgCell = trgSheet.Range(TARGET_CELL)
    '作成年の取得
    trgYear = trgSheet.Range(TARGET_YEAR).Value
    Call Make_Date(trgSheet, trgCell, trgYear)
    '実行前のシートを選択
    srcSheet.Select
End Sub

Public Sub Make_Date(makeSheet As Worksheet, makeCell As Range, makeYear As Integer)
    Dim tmpString As String
    Dim sdate As Date

```

```

Dim StartDate As Date
Dim endDate As Date
Application.ScreenUpdating = False
'取得する期間は、民間の1会計期にあわせ4月から翌年3月までとする
StartDate = makeYear & "/04/01"
endDate = (makeYear + 1) & "/03/31"
makeSheet.Select
makeCell.Activate
Range(Selection, Selection.End(xlToRight)).Select
Range(Selection, Selection.End(xlDown)).Select
Selection.ClearContents
makeCell.Activate

For sdate = StartDate To endDate
    If (check_date(sdate, tmpString)) Then
        ActiveCell.Offset(0, 0).Value = sdate
        ActiveCell.Offset(0, 1).Value = WeekdayName(Weekday(sdate), True)
        ActiveCell.Offset(0, 2).Value = tmpString
        ActiveCell.Offset(1, 0).Select
    End If
Next
Application.ScreenUpdating = True
End Sub

```

マクロを作成する際はインターネットで関連する項目を検索してみると良いです。

WebAPI の利用

インターネット上では URL に検索データを付加して呼び出すと結果を返すサイトがあります。

これを利用してマクロ化します。今回は郵便番号から住所を取得します。

まずサイトは <http://zip.cgis.biz/> を利用します。

利用例

```
http://zip.cgis.biz/csv/zip.php?zn=1030000
```

郵便番号 103-0000 を現行郵便番号から検索し該当する住所を格納した文字コード euc-jp の CSV ファイルを出力します。

検索結果

```

"ZipSearchXML","1.01","http://zip.cgis.biz/csv/zip.php?zn=1030000","1030000","none","1","1030000","0",
"1"
"トウキョウト","チュウオウク","イニケイサガ ナイ アイ","none","東京都","中央区","以下に掲載がない場合","none"

```

イメージ 住所関連.xlsx

=getZip2Address(C7)							
3	C	D	E	F	G	H	I
	890-0058						
	890-0059						
	890-0060						
	890-0061	鹿児島県鹿児島市天保山町					
	890-0062	鹿児島県鹿児島市与次郎					
	890-0063	鹿児島県鹿児島市鴨池					
	890-0064	鹿児島県鹿児島市鴨池新町					
	890-0065	鹿児島県鹿児島市郡元	鹿児島県	鹿児島市		郡元	
	890-0066	鹿児島県鹿児島市真砂町	鹿児島県	鹿児島市	真砂町		
	890-0067	鹿児島県鹿児島市真砂本町	鹿児島県	鹿児島市	真砂本町		
	890-0068	鹿児島県鹿児島市東郡元町	鹿児島県	鹿児島市	東郡元町		
	890-0069	鹿児島県鹿児島市南郡元町	鹿児島県	鹿児島市	南郡元町		
	890-0070						
	890-0071	鹿児島県鹿児島市三和町	鹿児島県	鹿児島市	三和町		
	890-0072	鹿児島県鹿児島市新栄町	鹿児島県	鹿児島市	新栄町		
	890-0073	鹿児島県鹿児島市宇宿	鹿児島県	鹿児島市		宇宿	
	890-0074	鹿児島県鹿児島市宇宿町	鹿児島県	鹿児島市	宇宿町		
	890-0075	鹿児島県鹿児島市桜ヶ丘(7、8丁目)	鹿児島県	鹿児島市		桜ヶ丘(7、8丁目)	
	079-1143	北海道赤平市赤平	北海道	赤平市		赤平	
	079-1144	北海道赤平市美園町	北海道	赤平市	美園町		
	079-1145						
	079-1146						

マクロ

'http://vba-geek.jp/blog-entry-219.html 出典

Function getZip2Address(strZipcode)

Dim objXMLHttp As Object, zipArr

""-""ハイフンが入っていた場合は取り除く

strZipcode = Replace(strZipcode, "-", "")

Set objXMLHttp = CreateObject("MSXML2.XMLHTTP")

objXMLHttp.Open "GET", "http://zip.cgis.biz/csv/zip.php?zn=" & strZipcode, False

objXMLHttp.Send

'API の結果を配列に代入する

zipArr = Split(Replace(objXMLHttp.responseText, "''", ""), ",")

'正常な値が返ってきた場合は配列の要素数が 15 になる

If UBound(zipArr) = 15 Then

```

        getZip2Address = zipArr(12) & zipArr(13) & zipArr(14)
    Else
        '郵便番号が間違っている場合や未入力の場合は、空文字を返す
        getZip2Address = ""
    End If
End Function

```

```

Sub DivAddr()
    Dim kugiri As Long
    Dim addr As String
    Dim arrayAddr(5) As String
    Dim wrange As Range
    Dim cell As Variant
    Dim rowCnt As Long

    rowCnt = 0
    Set wrange = Range(ActiveCell, ActiveCell.End(xlDown))
    For Each cell In wrange
        addr = cell.Value
        '半角文字を全角に変換
        addr = StrConv(addr, vbWide)
        '都道府県の検索
        kugiri = InStr(addr, "県")
        If kugiri = 0 Then '県がみつからない
            kugiri = InStr(addr, "都")
            If kugiri = 0 Then '都がみつからない
                kugiri = InStr(addr, "道")
                If kugiri = 0 Then '道がみつからない
                    kugiri = InStr(addr, "府")
                Else '府がみつからない
                    End If
            End If
        End If
        End If
        arrayAddr(1) = Mid(addr, kugiri + 1)
        arrayAddr(0) = Left(addr, kugiri)

        '郡市の検索
        kugiri = InStr(arrayAddr(1), "市")
    
```

```
If kugiri = 0 Then '市が見つからない
    kugiri = InStr(arrayAddr(1), "郡")
    If kugiri = 0 Then '郡が見つからない
        Else
        End If
    End If
End If
arrayAddr(2) = Mid(arrayAddr(1), kugiri + 1)
arrayAddr(1) = Left(arrayAddr(1), kugiri)

'区町村の検索
kugiri = InStr(arrayAddr(2), "区")
If kugiri = 0 Then '区が見つからない
    kugiri = InStr(arrayAddr(2), "町")
    If kugiri = 0 Then '町が見つからない
        kugiri = InStr(arrayAddr(2), "村")
    Else '村が見つからない
    End If
End If
arrayAddr(3) = Mid(arrayAddr(2), kugiri + 1)
arrayAddr(2) = Left(arrayAddr(2), kugiri)

ActiveCell.Offset(rowCnt, 1) = arrayAddr(0)
ActiveCell.Offset(rowCnt, 2) = arrayAddr(1)
ActiveCell.Offset(rowCnt, 3) = arrayAddr(2)
ActiveCell.Offset(rowCnt, 4) = arrayAddr(3)
ActiveCell.Offset(rowCnt, 5) = arrayAddr(4)
rowCnt = rowCnt + 1
Next
End Sub
```

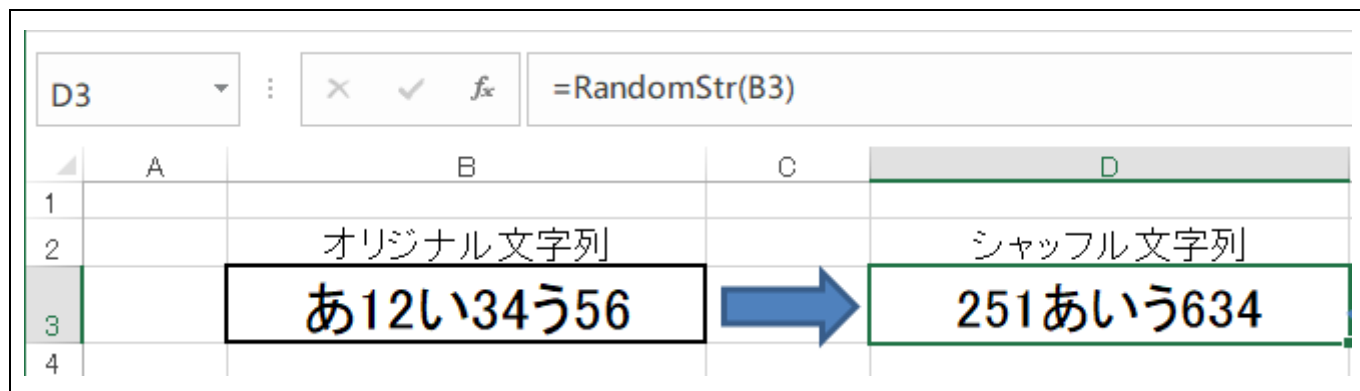
付録

セル内データのソート

行や列のソート(並べ替え)は標準で強力な処理が実装されていますが、セル内のデータを並べ替える機能はありません。ここではセル内データをランダム(シャッフル)に並べ替える処理を作成します。

講座で触れません。ご活用下さい。

イメージ セルデータシャッフル.xlsm



概要

セルに入力された文字列をランダムに並べ替えます。マクロ (VBA)で作成します。

Function プロシージャ名は RandomStr とし標準関数と同等な操作を可能とします

1. セルのデータを配列に格納(文字列の分割)
2. 乱数を用いて配列内を並べ替える
3. 配列のデータを連結する。(文字の連結)
4. 連結した文字列をプロシージャーの戻り値とする

コードは下記のようになります。

```
'#-----
'セルデータをランダムに並べ替える
'@parm    :wRange
'@return : ランダム文字列
'Version : 0.1
'#-----
'

Function RandomStr(wRange As Range) As Variant
    Dim wStr      As String
    Dim wTmp      As Variant
    Dim aryTbl() As Variant
    Dim idx       As Long
    Dim strCnt    As Long
```

```
Dim rndTmp As Long
Dim c As Variant
'エラーが発生した場合ラベルにジャンプする
On Error GoTo Err_Trap

'セルデータの取りだし
wStr = wRange.Value

'文字数の取得
strCnt = Len(wStr)

'配列を文字数で再設定
ReDim aryTbl(strCnt)

'文字列を配列に格納(文字列の分割)
'Mid 関数で取り出す文字を決める
For idx = 0 To strCnt - 1
    aryTbl(idx) = Mid(wStr, idx + 1, 1)
Next idx

'乱数の初期化
Randomize

'乱数を用いて並べ替え
'配列はゼロオリジンの為、添え字は 0 ～文字数 - 1 まで
For idx = 0 To strCnt - 1

    '乱数の取得
    'Rnd は 0 から 1 までの乱数を発生させる
    '0 から文字数までの乱数発生にするため、
    '文字数を乗算し整数に変換する
    'すなわち配列の添え字に適用できる。
    rndTmp = Int(Rnd * strCnt)

    '基準(idx)と対象(rndTmp)の入れ替え
    wTmp = aryTbl(rndTmp)
    aryTbl(rndTmp) = aryTbl(idx)
    aryTbl(idx) = wTmp
```

Next idx

'配列のデータを連結する(文字の連結)

wTmp = ""

'For Each 文は対象オブジェクトから 1 個ずつ取り出す

For Each c In aryTbl

wTmp = wTmp & c

Next

'戻り値に連結した文字列を設定する

RandomStr = wTmp

Exit Function

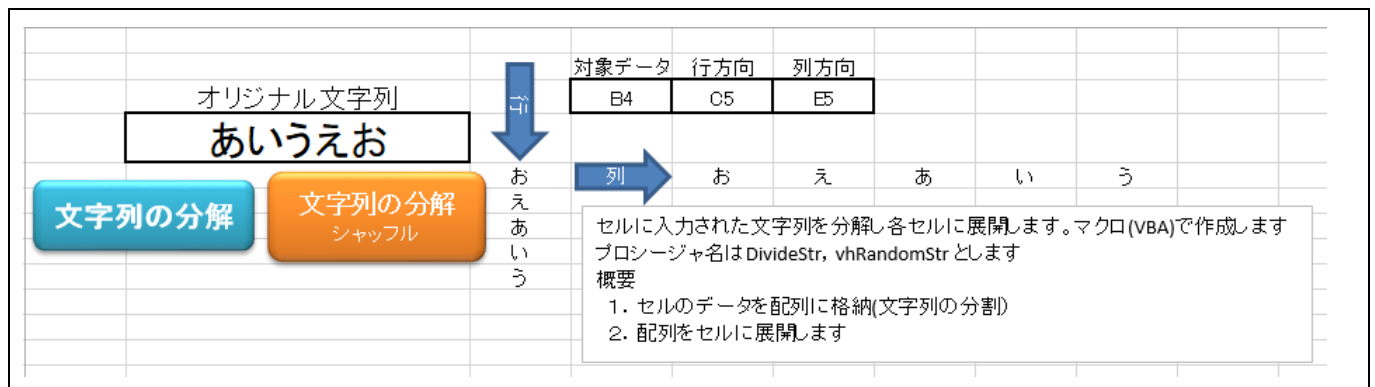
Err_Trap:

MsgBox "エラーコード NO." & Err.Number & vbCrLf & Err.Description

End Function

セル内データの分解

セル内データを分解し 1 文字 1 セルに展開する



概要

セルに入力された文字列を分解し各セルに展開します。マクロ(VBA)で作成します

Sub プロシージャ名は DivideStr とします

概要

1. セルのデータを配列に格納(文字列の分割)
2. 配列をセルに展開します

コードは下記のようになります。

```
'#-----
'セルデータを分解してセルに配置する
'@parm    : none
'@return : none
'Version : 0.1
'#-----
'

Sub DivideStr()
    Dim wStr      As String
    Dim tStr      As String
    Dim hStr      As String
    Dim vStr      As String
    Dim aryTblh() As Variant
    Dim aryTblv() As Variant
    Dim idx       As Long
    Dim strCnt     As Long

    On Error GoTo Err_Trap

    '対象セル位置取得
    tStr = Range("D3").Value
    '行方向開始位置取得
    vStr = Range("E3").Value
    '列方向開始位置取得
    hStr = Range("F3").Value

    wStr = Range(tStr).Value
    '文字数の取得
    strCnt = Len(wStr)

    'ワークシートへ展開するために2次元テーブルにする
    '列方向への展開用テーブル
    ReDim aryTblh(1, strCnt)
    '行方向への展開用テーブル
    ReDim aryTblv(strCnt, 1)
```

```

'文字列を配列に格納する(文字列の分解)
For idx = 0 To strCnt - 1
    aryTblv(idx, 0) = Mid(wStr, idx + 1, 1)
    aryTblh(0, idx) = Mid(wStr, idx + 1, 1)
Next idx

'開始位置を選択する
Range(vStr).Select
'行方向のクリア
With Range(ActiveCell, ActiveCell.End(xlDown))
    .Clear
End With
'選択されたセルから行方向へ文字数分の範囲
'文字の配置：縦横センターに
'2次元配列をセル範囲に代入
With Range(ActiveCell, ActiveCell.Offset(strCnt - 1, 0))
    .Select
    .HorizontalAlignment = xlCenter
    .VerticalAlignment = xlCenter
    .Value = aryTblv
End With

'開始位置を選択する
Range(hStr).Select
'列方向のクリア
With Range(ActiveCell, ActiveCell.End(xlToRight))
    .Clear
End With
'選択されたセルから列方向へ文字数分の範囲
'文字の配置：縦横センターに
'2次元配列をセル範囲に代入
With Range(ActiveCell, ActiveCell.Offset(0, strCnt - 1))
    .Select
    .HorizontalAlignment = xlCenter
    .VerticalAlignment = xlCenter
    .Value = aryTblh
End With

```

```
Range(tStr).Select
```

```
Exit Sub
```

```
Err_Trap:
```

```
MsgBox "エラーコード NO." & Err.Number & vbCrLf & Err.Description
```

```
End Sub
```

概要

セルに入力された文字列を分解し各セルにランダムに展開します。マクロ(VBA)で作成します

Sub プロシージャ名は vhRandomStr とします

概要

1. セルのデータを配列に格納(文字列の分割)
2. 配列をセルに展開します

コードは下記のようになります。

```
'
'#-----
'セルデータをランダムにセルに並べ替える
'@parm    : none
'@return : none
'Version : 0.1
'#-----
'

Sub vhRandomStr()
    Dim wStr      As String
    Dim tStr      As String
    Dim hStr      As String
    Dim vStr      As String
    Dim wTmp      As Variant
    Dim aryTblh() As Variant
    Dim aryTblv() As Variant
    Dim idx       As Long
    Dim strCnt    As Long
    Dim rndTmp    As Long
```

```
On Error GoTo Err_Trap

Application.ScreenUpdating = False

'対象セル位置取得
tStr = Range("D3").Value
'行方向開始位置取得
vStr = Range("E3").Value
'列方向開始位置取得
hStr = Range("F3").Value

wStr = Range(tStr).Value
'文字数の取得
strCnt = Len(wStr)

'ワークシートへ展開するために2次元テーブルにする
'列方向への展開用テーブル
ReDim aryTblh(1, strCnt)
'行方向への展開用テーブル
ReDim aryTblv(strCnt, 1)

'文字列を配列に格納する(文字列の分解)
For idx = 0 To strCnt - 1
    aryTblv(idx, 0) = Mid(wStr, idx + 1, 1)
    aryTblh(0, idx) = Mid(wStr, idx + 1, 1)
Next idx

'乱数の初期化
Randomize
'乱数を用いて並べ替え
'配列はゼロオリジンの為、添え字は0～文字数－1まで
For idx = 0 To strCnt - 1

    '乱数の取得
    'Rnd は0から1までの乱数を発生させる
    '0から文字数までの乱数発生にするため、
    '文字数を乗算し整数に変換する
    'すなわち配列の添え字に適用できる。
```

```
rndTmp = Int(Rnd * strCnt)
```

```
'行方向への展開テーブルの並べ替え
```

```
wTmp = aryTblv(rndTmp, 0)
```

```
aryTblv(rndTmp, 0) = aryTblv(idx, 0)
```

```
aryTblv(idx, 0) = wTmp
```

```
'列方向への展開テーブルの並べ替え
```

```
wTmp = aryTblh(0, rndTmp)
```

```
aryTblh(0, rndTmp) = aryTblh(0, idx)
```

```
aryTblh(0, idx) = wTmp
```

```
Next idx
```

```
'開始位置を選択する
```

```
Range(vStr).Select
```

```
'行方向のクリア
```

```
'途中にカラのセルがあるとクリアが出来ない場合あり
```

```
With Range(ActiveCell, ActiveCell.End(xlDown))
```

```
.Clear
```

```
End With
```

```
'選択されたセルから行方向へ文字数分の範囲
```

```
'文字の配置：縦横センターに
```

```
'2次元配列をセル範囲に代入
```

```
With Range(ActiveCell, ActiveCell.Offset(strCnt - 1, 0))
```

```
.Select
```

```
.HorizontalAlignment = xlCenter
```

```
.VerticalAlignment = xlCenter
```

```
.Value = aryTblv
```

```
End With
```

```
'開始位置を選択する
```

```
Range(hStr).Select
```

```
'列方向のクリア
```

```
'途中にカラのセルがあるとクリアが出来ない場合あり
```

```
With Range(ActiveCell, ActiveCell.End(xlToRight))
```

```
.Clear
```

End With

'選択されたセルから列方向へ文字数分の範囲

'文字の配置：縦横センターに

'2次元配列をセル範囲に代入

With Range(ActiveCell, ActiveCell.Offset(0, strCnt - 1))

.Select

.HorizontalAlignment = xlCenter

.VerticalAlignment = xlCenter

.Value = aryTblh

End With

Range(tStr).Select

Exit Sub

Err_Trap:

MsgBox "エラーコード NO." & Err.Number & vbCrLf & Err.Description

End Sub

Excel 標準処理を使用する

	A	B	C	D	E
1					
2					
3		オリジナル文字列	対象データ	出力	
4		あいうえお	B4	B6	
5					
6		えいうあお			
7					
8					
9					
10					

セルに入力された文字列をランダムに並べ替えます。マクロ(VBA)で作成します

プロシージャ名は WorkSheetSort とします。但しソート処理は Excel の並べ替えとフィルターの処理を利用しま

す。

概要

1. セルのデータを配列に格納(文字列の分割)
2. 配列をセルに展開します
3. 作業シートを追加します。
4. 作業シートに配列をセル展開します。
5. ソートします。
6. 作業シートを削除します。

コードは下記のようになります。

```
'
'#-----
'Excel のソートを使用する
'・手順
' 作業用ワークシートを追加
' 追加した作業用ワークシートに文字列をコピーする
' 更に分解した文字に乱数を設定する
' 乱数の昇順にデータを並べ替える
' 並び替えたデータを文字列に連結する
' 作業用ワークシートを削除する。
'#-----
'
Sub WorkSheetSort(ByRef charTbl As Variant, ByRef randTbl As Variant)
    Dim sheetName As String
    Dim idx, jdx, cnt As Long
    Dim wRange As Range
    Dim wSheet As Sheets
    Dim sortTbl() As Variant

    On Error GoTo Err_Trap

    'デバッグフラグ
    Const dFlag = True

    '1次元配列のままでも作業可能だが、
    'Worksheet への貼り付けを考慮して
    '2次元テーブルに格納する
    ReDim sortTbl(UBound(charTbl), 2)
```

```

For idx = 0 To UBound(charTbl) - 1
    sortTbl(idx, 0) = charTbl(idx)
    sortTbl(idx, 1) = randTbl(idx)
Next idx

Dim aryLength As Integer
Dim aryRank    As Integer

'確認用(デバック)
'動的配列の要素数取得
aryLength = UBound(sortTbl)
'動的配列の次元数取得
aryRank = arrayRank(sortTbl)

'画面描画を止める-ちらつき防止のため-
If dFlag Then
    Application.ScreenUpdating = False
End If
sheetName = "ソート作業"
If dFlag Then
    sheetName = sheetName & Format(Now(), "h 時 mm 分 ss 秒")
End If

'現在開いてるワークブックの最後尾にワークシートを追加し、シートネームを付加する。
Worksheets.Add(After:=Worksheets(Worksheets.Count)).Name = sheetName

'文字数×2のエリアを設定
'2次元テールにしたので貼り付けが下記で可能になる
Range("B2").Select
Set wRange = Range(ActiveCell, ActiveCell.Offset(aryLength - 1, aryRank - 1))
With wRange
    .Select
    .Value = sortTbl
End With

'転記した範囲を選択する
wRange.Select

```



```
'乱数を昇順にソートする
Selection.Sort key1:=wRange(1, 2), order1:=xlAscending
```

```
'ソートした文字を配列に格納する
For idx = LBound(charTbl) To UBound(charTbl) - 1
    charTbl(idx) = wRange(idx + 1, 1)
Next idx
```

```
'作業用シートを削除する
'アラートメッセージを抑制設定する
If dFlag Then
    Application.DisplayAlerts = False
End If
Worksheets(sheetName).Delete
'アラートメッセージを抑制解除する
Application.DisplayAlerts = True

Exit Sub
```

```
Err_Trap:
```

```
MsgBox "エラーコード NO." & Err.Number & vbCrLf & Err.Description
```

```
End Sub
```

```
,
```

```
'#-----
```

```
'動的配列の次元数を取得する
```

```
'@parm : arraytbl
```

```
'@return : 次元数
```

```
'Version : 0.1
```

```
'#-----
```

```
,
```

```
Function arrayRank(arrayTbl As Variant) As Long
```

```
Dim i As Long
```

```
Dim TempData As Variant
```

```
On Error Resume Next
```

```

Do While Err.Number = 0
    i = i + 1
    TempData = UBound(arrayTbl, i)
Loop
On Error GoTo 0

arrayRank = i - 1

```

End Function

ドリル作成

一問一答形式のドリル問題作成です。

講座では触れません。ご活用下さい。

概要

問題と解答を設定したシートを準備してランダムに最大 50 問出題します。

問題用紙・解答用紙・模範解答用紙を作成します。

イメージ ドリル作成.xlsm

操作パネル

問題解答

色んな漢字 ～問題用紙～

問題番号	漢字	クラス	番号	氏名
001	臥			
002	安道爾			
003	布隆迪			
004	鯨			
005	斯洛文尼亚			
006	白耳義			
007	海地			
008	宇岸訖			
009	缅甸			
010	聖登米・不林刺部			
011	薩摩亜			
012	洪都拉斯			
013	救世主国・薩爾瓦多			
014	鯨			
015	白露西亞・白露・西露西亞			

色んな漢字 ～解答用紙～

問題番号	漢字	クラス	番号	氏名
001	臥			
002	安道爾			
003	布隆迪			
004	鯨			
005	斯洛文尼亚			
006	白耳義			
007	海地			
008	宇岸訖			
009	缅甸			
010	聖登米・不林刺部			
011	薩摩亜			
012	洪都拉斯			
013	救世主国・薩爾瓦多			
014	鯨			
015	白露西亞・白露・西露西亞			

問題用紙

解答用紙

色んな漢字 ～模範解答～

問題番号	漢字	クラス	番号	氏名
001	臥			
002	安道爾			
003	布隆迪			
004	鯨			
005	斯洛文尼亚			
006	白耳義			
007	海地			
008	宇岸訖			
009	缅甸			
010	聖登米・不林刺部			
011	薩摩亜			
012	洪都拉斯			
013	救世主国・薩爾瓦多			
014	鯨			
015	白露西亞・白露・西露西亞			

模範解答

問題番号	漢字	クラス	番号	氏名
001	臥			
002	安道爾			
003	布隆迪			
004	鯨			
005	斯洛文尼亚			
006	白耳義			
007	海地			
008	宇岸訖			
009	缅甸			
010	聖登米・不林刺部			
011	薩摩亜			
012	洪都拉斯			
013	救世主国・薩爾瓦多			
014	鯨			
015	白露西亞・白露・西露西亞			

模範解答用紙

Option Explicit

Private Const dataMaxRow = 50

'最大問題数

Private Const buMaxCnt = 10

'最大問題部数

Private Const QuestionLine = "B3:O52"

'問題用紙の問題文範囲

Dim selectSection(buMaxCnt) As Integer

'入力された部

' #-----

' Macro : ドリル問題作成メイン処理

' @parm : wRange

' @return : ランダム文字列

```

' Version : 0.1
' #-----
Sub mainExercise()
    Dim SheetName As String
    Dim msgStr     As String
    Dim cnt        As Integer
    Dim dataCnt    As Integer
    Dim RandCNt    As Integer
    Dim ret        As Integer
    Dim subTitle   As String
    Dim mesCnt     As Integer

    Application.ScreenUpdating = False

    Sheets("作業").Visible = True
    Sheets("乱数表").Visible = True

    dataCnt = 問題 TO 作業()

    If dataCnt = -1 Then
        Sheets("作業").Visible = False
        Sheets("乱数表").Visible = False
        Exit Sub
    End If

    RandCNt = 乱数 TO 作業(dataCnt)

    Call ソートコピー(dataCnt)

    Call 作業 TO 問題

    Call 作業 TO 解答

    Sheets("作業").Visible = False
    Sheets("乱数表").Visible = False

    SheetName = "コンパネ"
    Sheets(SheetName).Select

```

```
If dataCnt >= dataMaxRow Then
```

```
    mesCnt = dataMaxRow
```

```
Else
```

```
    mesCnt = dataCnt
```

```
End If
```

```
'サブタイトルの更新
```

```
subTitle = Range("B5").Value
```

```
subTitle = subTitle & mesCnt & "問"
```

```
Range("B5").Value = subTitle
```

```
'完了メッセージ表示
```

```
msgStr = "全 " & dataCnt & " 問から " & mesCnt & " 問シャッフルしました。"
```

```
ret = MsgBox(msgStr, vbOKOnly, "完了")
```

```
End Sub
```

```
' #-----
```

```
' Macro    : 問題用紙のセルを連結する,連結したセルに罫線を付加する
```

```
' @parm    : none
```

```
' @return : none
```

```
' Version : 0.1
```

```
' #-----
```

```
Sub セル連結開始()
```

```
    Dim SheetName As String
```

```
    Dim Row        As Integer
```

```
    Dim i          As Integer
```

```
    Dim wRange     As Range
```

```
    Application.ScreenUpdating = False
```

```
    SheetName = "問題用紙"
```

```
    Sheets(SheetName).Activate
```

```
    Set wRange = Range(QuestionLine)
```

```
    Row = wRange.Rows.Count
```

```

For i = 1 To wRange.Rows.Count
    wRange.Rows(i).Select
    Selection.Merge
    Selection.Borders.LineStyle = True
    Selection.Font.Size = 20
    Selection.HorizontalAlignment = xlGeneral
    Selection.VerticalAlignment = xlCenter
Next i

Range("A2").Select

'Application.ScreenUpdating = True

End Sub

' #-----
' Macro    : 問題用紙の連結したセルを解除する,解除したセルの罫線を削除
' @param   : none
' @return  : none
' Version : 0.1
' #-----
Sub セル連結解除()
    Dim SheetName As String
    Dim Row        As Integer
    Dim wRange     As Range

    Application.ScreenUpdating = False

    SheetName = "問題用紙"
    Sheets(SheetName).Activate

    Set wRange = Range(QuestionLine)
    wRange.Select
    Selection.UnMerge
    Selection.Borders.LineStyle = False
    Selection.Clear

```

```
'Application.ScreenUpdating = True

End Sub

' #-----
' Macro    : 解答用紙の解答群のセルを連結する,連結したセルに罫線を付加
' @param   : none
' @return  : none
' Version  : 0.1
' #-----
Sub セル連結開始 2()

    Dim SheetName As String
    Dim i As Integer
    Dim j As Integer

    Application.ScreenUpdating = False

    SheetName = "解答用紙"
    Sheets(SheetName).Activate

    For i = 1 To 2

        For j = 1 To 25

            Range(Cells(j + 2, (i - 1) * 4 + 9), Cells(j + 2, (i - 1) * 4 + 9 + 2)).Select
            Selection.Merge
            'Selection.Borders.LineStyle = True

        Next j

    Next i

    Range("A2").Select

    'Application.ScreenUpdating = True
```

End Sub

```
' #-----
' Macro    : 解答用紙の解答群の連結したセルを解除する,解除したセルの罫線を削除
' @parm    : none
' @return  : none
' Version  : 0.1
' #-----
```

Sub セル連結解除 2()

```
    Dim SheetName As String
    Dim Row        As Integer

    Application.ScreenUpdating = False

    SheetName = "解答用紙"
    Sheets(SheetName).Activate

    Range(Cells(3, 9), Cells(25 + 2, 16)).Select
    Selection.UnMerge
    Selection.Borders.LineStyle = False
    Selection.Clear

    'Application.ScreenUpdating = True
```

End Sub

```
' #-----
' Macro    : 作業シートをクリアする
' @parm    : none
' @return  : none
' Version  : 0.1
' #-----
```

Sub 作業クリア()

```
    Dim SheetName As String
    Dim Row        As Integer

    Const HeadLine = 3
```



```

Application.ScreenUpdating = False

SheetName = "作業"
Sheets(SheetName).Activate

Range(Cells(HeadLine, 3), Cells(HeadLine + 1000, 256)).Select
Selection.UnMerge
Selection.Borders.LineStyle = False
Selection.Clear

'Application.ScreenUpdating = True

End Sub

' #-----
' Macro    : 入力された部をクリアする
' @parm    : none
' @return  : none
' Version : 0.1
' #-----
Sub 部クリア()
    Dim SheetName As String

    Application.ScreenUpdating = False

    SheetName = "コンパネ"
    Sheets(SheetName).Activate
    Range("B5") = ""
    Range(Cells(7, 2), Cells(7, buMaxCnt - 1)).ClearContents

    'Application.ScreenUpdating = True

End Sub

' #-----
' Macro    : 問題を作業に転記する。
' @parm    : none
' @return  : 指定された部の総問題数

```

```

' Version : 0.1
' #-----
Function 問題 TO 作業()
    Dim cnt        As Integer
    Dim idx        As Integer
    Dim section    As Integer
    Dim retCnt     As Integer
    Dim SheetName As String
    Dim subTitle   As String
    Dim ret        As Integer

    Const HeadLine = 7

    Call 作業クリア

    Application.ScreenUpdating = False

    SheetName = "コンパネ"
    Sheets(SheetName).Select

    '配列クリア(初期値-1 設定)
    For idx = 0 To buMaxCnt - 1
        selectSection(idx) = -1
    Next

    '部入力エリアを先頭から検査
    cnt = 0
    For idx = 0 To buMaxCnt - 1
        If Cells(HeadLine, 2 + idx) <> "" Then
            selectSection(cnt) = Cells(HeadLine, 2 + idx)
            subTitle = subTitle & selectSection(cnt) & ","
            cnt = cnt + 1
        End If
    Next

    If Range("B3").Value = "" Or Range("B3").Value = Null Then
        ret = MsgBox("タイトルを入力して下さい！", vbOKOnly, "エラー")
        問題 TO 作業 = -1
    End If
End Function

```

```
Exit Function
End If

If selectSection(0) = -1 Then
    ret = MsgBox("部を入力して下さい！", vbOKOnly, "エラー")
    問題 TO 作業 = -1
    Exit Function
End If

subTitle = Mid(subTitle, 1, Len(subTitle) - 1)
subTitle = "第" & subTitle & "回から"
Range("B5").Value = subTitle

cnt = 0
For idx = 0 To buMaxCnt - 1

    If selectSection(idx) = -1 Then
        Exit For
    End If

    SheetName = "問題解答"
    Sheets(SheetName).Select

    '部の問題数取得
    retCnt = Cells(1, selectSection(idx) * 2)

    '問題を作業シートにコピー
    If retCnt > 0 Then
        Range(Cells(3, selectSection(idx) * 2 - 1), Cells(retCnt + 2, selectSection(idx) * 2)).Select
        Selection.Copy

        SheetName = "作業"
        Sheets(SheetName).Select
        Range(Cells(cnt + 3, 3), Cells(cnt + retCnt + 3, 4)).PasteSpecial xlPasteValues
        cnt = cnt + retCnt
    End If
```

```

Next
'戻り値の設定
'問題数の総数
問題 TO 作業 = cnt

'Application.ScreenUpdating = True

End Function

'
' #-----
' Macro    : 乱数を問題数にあわせて作業に転記する。
' @parm    : 乱数を付加する総問題数
' @return  : 付加した乱数の総数(※問題数より大きくなる可能性がある。)
' Version : 0.1
' #-----
Function 乱数 TO 作業(intR)

    Dim cnt        As Integer
    Dim idx        As Integer
    Dim SheetName As String
    Dim intRnd     As Integer

    Const HeadLine = 3

    Application.ScreenUpdating = False

    cnt = 0
    Do While intR > cnt
        SheetName = "乱数表"
        Sheets(SheetName).Select

        '乱数系列を初期化
        Randomize
        intRnd = Int((dataMaxRow - 1 + 1) * Rnd + 1)
        'Debug.Print "RND->" & intRnd

        Range(Cells(HeadLine, 2 + intRnd), Cells(HeadLine - 1 + dataMaxRow, 2 + intRnd)).Select
    
```

```

Selection.Copy

SheetName = "作業"
Sheets(SheetName).Select
If cnt = 0 Then
    Range(Cells(HeadLine, 5), Cells(HeadLine + dataMaxRow, 5)).PasteSpecial xlPasteValues
Else
    Range(Cells(cnt + HeadLine, 5), Cells(cnt + HeadLine + dataMaxRow, 5)).PasteSpecial
xlPasteValues
End If
cnt = cnt + dataMaxRow

Loop

'余分な乱数をクリアする
Range(Cells(HeadLine + intR, 5), Cells(HeadLine + cnt - 1, 5)).Clear

'Application.ScreenUpdating = True

乱数 TO 作業 = cnt

End Function

' #-----
' Macro    : 問題をソートし最大 50 問に制限する
' @parm    : 総問題数
' @return : none
' Version : 0.1
' #-----
Sub ソートコピー(intR)

    Dim SheetName As String

    Const HeadLine = 3

    Application.ScreenUpdating = False

```

```

SheetName = "作業"
Sheets(SheetName).Select

'選択された全てのデータをソート
Range(Cells(HeadLine, 3), Cells(HeadLine + intR - 1, 5)).Select
Selection.Sort key1:=Range("E1"), order1:=xlAscending

'最大問題数に限定する
Range(Cells(HeadLine + dataMaxRow, 3), Cells(HeadLine + intR, 5)).Clear
Range(Cells(HeadLine, 3), Cells(HeadLine + dataMaxRow - 1, 5)).Select
Selection.Copy
Range("G3").PasteSpecial xlPasteValues

'選択肢ソート
Range(Cells(HeadLine, 7), Cells(HeadLine + dataMaxRow - 1, 9)).Select
Selection.Sort key1:=Range("G1"), order1:=xlAscending

'Application.ScreenUpdating = True

```

End Sub

```

' #-----
' Macro    : 問題を問題用紙に転記
' @parm    : none
' @return  : none
' Version : 0.1
' #-----

```

Sub 作業 TO 問題()

```

Dim SheetName As String

Const HeadLine = 3

Call セル連結解除

Application.ScreenUpdating = False

```

```

SheetName = "作業"
Sheets(SheetName).Select
'問題書き戻し
Range(Cells(HeadLine, 4), Cells(HeadLine + dataMaxRow - 1, 4)).Select
Selection.Copy

```

```

SheetName = "問題用紙"
Sheets(SheetName).Select
Range("B3").PasteSpecial xlPasteValues
Call セル連結開始

```

```
End Sub
```

```

' #-----
' Macro      : 解答を解答用紙に転記
' @parm      : none
' @return    : none
' Version : 0.1
' #-----

```

```
Sub 作業 TO 解答()
```

```

    Dim SheetName As String
    Dim cnt        As Integer

```

```

    Const HeadLine = 3
    Const rowCnt = 25

```

```
Call セル連結解除 2
```

```
Application.ScreenUpdating = False
```

```
For cnt = 0 To 1
```

```

    SheetName = "作業"
    Sheets(SheetName).Select
    '解答書き戻し
    Range(Cells(HeadLine + ((cnt * 1) * rowCnt), 3), Cells(HeadLine + ((cnt * 1) * rowCnt) + rowCnt
- 1, 3)).Select
    Selection.Copy

```

```

    SheetName = "解答用紙"
    Sheets(SheetName).Select

    Cells(HeadLine, (cnt * 3) + 2).PasteSpecial xlPasteValues

Next

For cnt = 0 To 1

    SheetName = "作業"
    Sheets(SheetName).Select
    '解答書き戻し
    Range(Cells(HeadLine + ((cnt * 1) * rowCnt), 7), Cells(HeadLine + ((cnt * 1) * rowCnt) + rowCnt
- 1, 7)).Select
    Selection.Copy

    SheetName = "解答用紙"
    Sheets(SheetName).Select

    Cells(HeadLine, (cnt * 4) + 9).PasteSpecial xlPasteValues

Next

Call セル連結開始 2

End Sub

' #-----
' Macro    : 解答用紙と模範解答の切替(文字を白色にすることで空白に見せかける)
' @parm    : True:模範解答用紙,False:解答用紙
' @return  : none
' Version  : 0.1
' #-----
Sub 色反転(flag As Boolean)
    Dim SheetName As String

```



```

Application.ScreenUpdating = False

SheetName = "解答用紙"
Sheets(SheetName).Select

If flag Then
    '黒 タイトルを模範解答に変更
    Range("B3:B27").Select
    Selection.Font.ColorIndex = 0
    Range("E3:E27").Select
    Selection.Font.ColorIndex = 0
    Cells(1, 1) = "=コンパネ!B3 & "" "" &コンパネ!B11"

Else
    '白 タイトルを解答用紙に変更
    Range("B3:B27").Select
    Selection.Font.ColorIndex = 2
    Range("E3:E27").Select
    Selection.Font.ColorIndex = 2
    Cells(1, 1) = "=コンパネ!B3 & "" "" &コンパネ!B10"

End If

Range("A2").Select

Application.ScreenUpdating = True

End Sub

' #-----
' Macro    : 印刷、プレビューメイン
' @parm    : flag->True:プレビュー,False:印刷
' @parm    : copies->印刷枚数
' @return  : none
' Version : 0.1
' #-----
Sub 印刷(flag As Boolean, copies As Integer)

```

```

    If flag Then
        ActiveSheet.PrintPreview
    Else
        ActiveSheet.PrintOut copies:=copies
    End If

End Sub

' #-----
' Macro      : 問題用紙プレビュー
' @parm      : none
' @return    : none
' Version    : 0.1
' #-----

Sub 印刷 M1()
    Dim SheetName As String
    Dim prtCnt     As Integer

    SheetName = "コンパネ"
    '印刷部数取得  しかし未使用
    prtCnt = Sheets(SheetName).Range("B7").Value

    SheetName = "問題用紙"
    Sheets(SheetName).Activate

    Call 印刷(True, prtCnt)

End Sub

' #-----
' Macro      : 問題用紙 1 部印刷
' @parm      : none
' @return    : none
' Version    : 0.1
' #-----
'
Sub 印刷 M2()

```

```

Dim SheetName As String
Dim prtCnt    As Integer

SheetName = "コンパネ"
'印刷部数取得  しかし未使用
prtCnt = Sheets(SheetName).Range("B7").Value

SheetName = "問題用紙"
Sheets(SheetName).Activate

Call 印刷(False, prtCnt)

End Sub

' #-----
' Macro    : 解答用紙プレビュー
' @param   : none
' @return  : none
' Version : 0.1
' #-----
'
Sub 印刷 A1()
    Dim SheetName As String
    Dim prtCnt    As Integer

    SheetName = "コンパネ"
    '印刷部数取得  しかし未使用
    prtCnt = Sheets(SheetName).Range("B7").Value

    SheetName = "解答用紙"
    Sheets(SheetName).Activate

    Call 色反転(False)

    Call 印刷(True, prtCnt)

End Sub

```

```

' #-----
' Macro    : 解答用紙 1 部印刷
' @parm    : none
' @return  : none
' Version  : 0.1
' #-----
'
Sub 印刷 A2()
    Dim SheetName As String
    Dim prtCnt    As Integer

    SheetName = "コンパネ"
    '印刷部数取得  しかし未使用
    prtCnt = Sheets(SheetName).Range("B7").Value

    SheetName = "解答用紙"
    Sheets(SheetName).Activate

    Call 色反転(False)

    Call 印刷(False, prtCnt)

End Sub

' #-----
' Macro    : 模範解答プレビュー
' @parm    : none
' @return  : none
' Version  : 0.1
' #-----
'
Sub 印刷 B1()
    Dim SheetName As String
    Dim prtCnt    As Integer

    SheetName = "コンパネ"
    '印刷部数取得  しかし未使用
    prtCnt = Sheets(SheetName).Range("B7").Value

```

```
SheetName = "解答用紙"
Sheets(SheetName).Activate

Call 色反転(True)

Call 印刷(True, prtCnt)

End Sub

' #-----
' Macro    : 模範解答印刷
' @parm    : none
' @return  : none
' Version : 0.1
' #-----
'
Sub 印刷 B2()
    Dim SheetName As String
    Dim prtCnt     As Integer

    SheetName = "コンパネ"
    '印刷部数取得  しかし未使用
    prtCnt = Sheets(SheetName).Range("B7").Value

    SheetName = "解答用紙"
    Sheets(SheetName).Activate

    Call 色反転(True)

    Call 印刷(False, prtCnt)

End Sub
```

座席表作成

講座では触れません。ご活用下さい。

概要

教室に合わせ、列数と 1 列の席数を指定し、作成ボタンを押すと、座席表を自動で作成します。

先生用座席表を出力する設定を行うと先生側から見た座席表も出力します。

イメージ 座席表.xlsm

E11	:	<input type="text"/>	<input type="text"/>	<input type="text"/>	
列数		<input type="text" value="8"/>			
1列の席数		<input type="text" value="6"/>			
座席表のタイトル		<input type="text" value="サンبل座席表"/>			
先生用座席表も出力		<input checked="" type="checkbox" value="する"/>			
作成					
KCS鹿兒島情報専門学校					
<	>	コントロール	名簿	座席表	座席表(先生用)
		+>	:<		
準備完了					

A3	:	<input type="text"/>	<input type="text"/>	<input type="text"/>	1		
1	A	B	C	D	E F G		
2	出席番号	名前	ふりがな	アドレス	性別	誕生日	携帯
3	1	山本 希	あきやま のぞみ	akiyama_nozomi@example.com	女	1994/3/21	090-4757-7951
4	2	新野 龍吉	いづか りょうきち	iidake_ryuukichi@example.com	男	1984/10/16	090-4265-2917
5	3	石塚 ひとり	いしがき ひとり	ishigaki_hitori@example.com	男	1994/5/28	090-6095-9642
6	4	井田 まさみ	いた まさみ	ide_masaki@example.com	女	1993/12/14	090-1847-7439
7	5	岩井 育子	いわい いくこ	iwai_ikuko@example.com	女	1984/3/27	090-8627-2529
8	6	千野 早紀	ちの さき	uno_saki@example.com	女	1983/12/27	090-1560-0783
9	7	大野 真奈美	おおの まなみ	ono_manami@example.com	女	1994/8/30	090-3782-1865
10	8	大和田 未来	おおわだ みらい	oomeda_mirai@example.com	女	1994/4/21	090-1370-5652
11	9	奥 真	おく かおる	oku_kaooru@example.com	男	1994/7/16	090-9392-3163
12	10	風間 真帆	かざま まほ	kazama_maho@example.com	女	1994/1/3	090-8296-5396
13	11	香椎 さやか	かしい さやか	kashii_sayaka@example.com	女	1994/4/28	090-9561-3167
14	12	上条 隼士	かみじょう しゅんじ	kamijou_shunji@example.com	男	1993/12/9	090-3651-0693
15	13	北 丈	きた たける	kita_takeru@example.com	男	1994/8/2	090-3773-6116
16	14	黒木 桃子	くろぎ ももこ	kuroki_momoko@example.com	女	1994/2/24	090-3404-3036
17	15	河本 直人	こうもと なおと	koumoto_naoto@example.com	男	1994/3/14	090-2899-0035
18	16	小杉 まし	こすぎ まさし	kosugi_masashi@example.com	男	1994/1/2	090-4010-8223
19	17	沢村 里奈	さわむら りな	sawamura_rina@example.com	女	1994/10/4	090-7080-3596
20	18	篠山 勤	しのやま つとむ	shinoyama_tsutomu@example.com	男	1994/7/15	090-6414-3393
21	19	城田 由樹	しろた ゆき	shirota_yuki@example.com	女	1994/11/11	090-8941-5052
22	20	轟崎 寿々花	すずき すずか	suzuki_suzuka@example.com	女	1994/3/8	090-7109-9823
23	21	竹原 千佳子	たけはら ちかこ	takahara_chika@example.com	女	1994/5/10	090-6159-0676
24	22	橘 孝太郎	たちばな こうたろう	tachibana_koutarou@example.com	男	1994/7/17	090-0590-5716
25	23	田辺 里穂	たなべ りほ	tanabe_rihoh@example.com	女	1994/12/3	090-4378-6444

操作パネル

名簿

H2															H28														
教 卓															教 卓														
サンプル座席表															サンプル座席表														
教 卓															教 卓														
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120
121	122	123	124	125	126	127	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143	144	145	146	147	148	149	150
151	152	153	154	155	156	157	158	159	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175	176	177	178	179	180
181	182	183	184	185	186	187	188	189	190	191	192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207	208	209	210
211	212	213	214	215	216	217	218	219	220	221	222	223	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239	240
241	242	243	244	245	246	247	248	249	250	251	252	253	254	255	256	257	258	259	260	261	262	263	264	265	266	267	268	269	270
271	272	273	274	275	276	277	278	279	280	281	282	283	284	285	286	287	288	289	290	291	292	293	294	295	296	297	298	299	300
301	302	303	304	305	306	307	308	309	310	311	312	313	314	315	316	317	318	319	320	321	322	323	324	325	326	327	328	329	330
331	332	333	334	335	336	337	338	339	340	341	342	343	344	345	346	347	348	349	350	351	352	353	354	355	356	357	358	359	360
361	362	363	364	365	366	367	368	369	370	371	372	373	374	375	376	377	378	3											

座席表

座席表 (先生用)

Option Explicit

Public retusu As Integer ' 列数

```

Public sekisu As Integer ' 1 列の座席数
Public teach As Boolean ' true なら先生から見た座席表の作成
Public title As String ' 座席表のタイトル
Const OUT_SHEET01 As String = "座席表"
Const OUT_SHEET02 As String = "座席表（先生用）"

Public Sub 座席表作成()
    Dim wrksht As Worksheet

    ' 「コントロール」シートから列数、1 列の席数などを読み込む
    Worksheets("コントロール").Select
    retusu = Cells(3, 3).Value '列数
    sekisu = Cells(5, 3).Value ' 1 列の席数
    title = Cells(7, 3).Value '座席表のタイトル
    If Cells(9, 3).Value = "する" Then '先生用座席表
        teach = True ' 出力する
    Else
        teach = False '出力しない
    End If

    Application.DisplayAlerts = False
    For Each wrksht In Worksheets
        If wrksht.Name = OUT_SHEET01 Then
            wrksht.Delete
        ElseIf wrksht.Name = OUT_SHEET02 Then
            wrksht.Delete
        End If
    Next
    Application.DisplayAlerts = True

    ' 座席表を入れるワークシートを追加
    Worksheets.Add After:=Worksheets(Worksheets.Count)
    Worksheets(Worksheets.Count).Name = OUT_SHEET01

    ' 1 列に何席作ったかのカウンタ

```

```

Dim Cnt As Integer
Cnt = 0

' 座席の行番号
Dim s As Integer
s = 4 '4 行目から作成

' 座席の列番号
Dim r As Integer
r = 2 '2 列目から作成

' この部分で出席番号が入っている列を選択する
' 必要に応じて変更してください。
Worksheets("名簿").Select ' <-----名簿が入っているシート名に変更
Range("A3").Select ' <-----出席番号の最初のセルに変更
Range(Selection, Selection.End(xlDown)).Select ' その列全体をセレクトする「Ctrl + Shift + ↓」の動き

' 出席番号の各セルに対して繰り返し
Dim x As Variant
For Each x In Selection
    With Worksheets(OUT_SHEET01)
        .Cells(s, r).ColumnWidth = 14 '名前の入っているセル幅は広く
        .Cells(s, r - 1).ColumnWidth = 2.5 '名前の入っていないセルは狭くする
        .Cells(s, r) = x.Value ' 出席番号を格納
        .Cells(s, r).HorizontalAlignment = xlCenter ' 中央揃えにする（お好みで）
        .Cells(s + 1, r) = x.Cells(1, 3) ' ふりがなを格納
        .Cells(s + 1, r).Font.Size = 8 '文字サイズを小さく
        .Cells(s + 1, r).HorizontalAlignment = xlCenter ' 中央揃えにする（お好みで）
        .Cells(s + 2, r) = x.Cells(1, 2) ' 氏名を格納
        .Cells(s + 2, r).HorizontalAlignment = xlCenter ' 中央揃えにする（お好みで）
        .Range(.Cells(s, r), .Cells(s + 2, r)).BorderAround Weight:=xlThin ' 罫線で囲む
    End With

    Cnt = Cnt + 1 ' カウントアップ
    If Cnt = sekisu Then ' 1 列分作成したか
        s = 4 ' そうなら、次の列へ移動
        r = r + 2
    End If
End For

```



```

        Cnt = 0
    Else
        s = s + 4    ' そうでないなら、1 席分下へ移動
    End If
Next x

'教卓表示位置（中央）の計算
Dim takustart As Integer
Dim takuend As Integer

'偶数列と奇数列で表示位置が違う
If (retusu Mod 2) = 0 Then '偶数列
    takustart = retusu / 2 * 2
    takuend = ((retusu / 2) + 1) * 2
Else '奇数列
    takustart = (retusu ¥ 2) * 2
    takuend = ((retusu / 2) + 2) * 2 - 1
End If

' 計算した位置に教卓を表示
Worksheets(OUT_SHEET01).Select
Range(Cells(2, takustart), Cells(2, takuend)).Select
Selection.HorizontalAlignment = xlCenter
Selection.Merge
Range(Cells(2, takustart), Cells(2, takuend)).BorderAround LineStyle:=xlDouble, Weight:=xlThin
ActiveCell.FormulaR1C1 = "教    卓"
ActiveCell.Font.Size = 14    '文字サイズを大きく

'先頭にタイトルを格納
Cells(1, 1).Value = title
Cells(1, 1).Font.Size = 18

'先生側から見た座席表を出力
If teach = True Then
    Call 座席表作成_先生用
End If

```

End Sub

Private Sub 座席表作成_先生用()

' 座席表を入れるワークシートを追加

Worksheets.Add After:=Worksheets(Worksheets.Count)

Worksheets(Worksheets.Count).Name = OUT_SHEET02

' 1 列に何席作ったかのカウンタ

Dim Cnt As Integer

Cnt = 0

' 座席の行番号

Dim s As Integer

s = sekisu * 4 + 2 '座席表開始位置の計算

' 座席の列番号

Dim r As Integer

r = retusu * 2 '座席表開始位置の計算

' この部分で出席番号が入っている列を選択する

' 必要に応じて変更してください。

Worksheets("名簿").Select ' <-----名簿が入っているシート名に変更

Range("A3").Select ' <-----出席番号の最初のセルに変更

Range(Selection, Selection.End(xlDown)).Select ' その列全体をセレクトする「Ctrl + Shift + ↓」の

動き

' 出席番号の各セルに対して繰り返し

Dim x As Variant

For Each x In Selection

With Worksheets(OUT_SHEET02)

.Cells(s, r).ColumnWidth = 14 'セル幅の変更

.Cells(s, r - 1).ColumnWidth = 2.5 '名前の入っていないセルは狭くする

.Cells(s - 2, r) = x.Value ' 出席番号を格納

.Cells(s - 2, r).HorizontalAlignment = xlCenter ' 中央揃えにする (お好みで)

.Cells(s - 1, r) = x.Cells(1, 3) ' ふりがなを格納

.Cells(s - 1, r).Font.Size = 8 '文字サイズを小さく

.Cells(s - 1, r).HorizontalAlignment = xlCenter ' 中央揃えにする (お好みで)

```

.Cells(s, r) = x.Cells(1, 2) ' 氏名を格納
.Cells(s, r).HorizontalAlignment = xlCenter ' 中央揃えにする (お好みで)
.Range(.Cells(s, r), .Cells(s - 2, r)).BorderAround Weight:=xlThin ' 罫線で囲む
End With

Cnt = Cnt + 1 ' カウントアップ
If Cnt = sekisu Then ' 1 列分作成したか
    s = sekisu * 4 + 2 ' そうなら、次の列へ移動
    r = r - 2
    Cnt = 0
Else
    s = s - 4 ' そうでないなら、1 席分下へ移動
End If
Next x

'教卓表示位置 (中央) の計算
Dim takustart As Integer
Dim takuend As Integer

'教卓表示行
s = sekisu * 4 + 4

'偶数列と奇数列で表示位置が違う
If (retusu Mod 2) = 0 Then '偶数列
    takustart = retusu / 2 * 2
    takuend = ((retusu / 2) + 1) * 2
Else
    takustart = (retusu ¥ 2) * 2
    takuend = ((retusu / 2) + 2) * 2 - 1
End If

' 計算した位置に教卓を表示
Worksheets(OUT_SHEET02).Activate
Range(Cells(s, takustart), Cells(s, takuend)).Select
Selection.HorizontalAlignment = xlCenter
Selection.Merge
Range(Cells(s, takustart), Cells(s, takuend)).BorderAround LineStyle:=xlDouble, Weight:=xlThin
ActiveCell.FormulaR1C1 = "教 卓"

```

```
ActiveCell.Font.Size = 14      '文字サイズを大きく
```

```
'先頭にタイトルを格納
```

```
Cells(1, 1).Value = title
```

```
Cells(1, 1).Font.Size = 18
```

```
End Sub
```

改訂履歴	日 付	備 考
初版	2017 年 7 月 17 日	2 日間コース用に新規作成

メモ：
