BOOK操作

# セルの操作

## 行の選択

Range("B2").EntireRow.Select　‘２行目

Rows(2).Select ' 2 行目

Range("2:3").Select ' 2 ～ 3 行目を選択

Range("1:1, 3:4").Select　‘離れた行の選択

## 列の選択

Columns(3).Select　 '3列目を選択

Range(Columns(3), Columns(5)).Select '　3～5列目を選択

Columns(3).Delete　 '3列目を削除

Columns(3).Insert 　'3列目に挿入

Columns(3).Copy Columns(5)　 '3列目を5列目にコピー

## コピーモード解除

Application.CutCopyMode = False

# ブックの保存

ThisWorkBook.SaveAs ThisWorkBook.Path & “Test1.xlsx”

## ブックを閉じる（上書き保存/保存せずに）

ActiveWorkBook.SaveAs ActiveWorkBook.Path & “Test2.xlsx”

Workbooks(“ブック名”).Close True '上書き保存して閉じる

Workbooks(“ブック名”).Close True , ファイルパス '名前を付けて保存してから閉じる

ブック.Close SaveChanges:=True／False　‘ブックを閉じるときに表示される確認ダイアログ非表示

'ブック名を指定した上書き保存

Workbooks(“ブック名”).SaveAs Filename:= Path & “Test3.xlsx

## ブック名変更（ブック閉じてから）

Oldpathname ＝ “C:\Users\US525182\Desktop\テスト” &”\” & “古いファイル名.xlsx”

Newpathname= “C:\Users\US525182\Desktop\テスト” &”\” & “新ファイル名.xlsx”

Name Oldpathname As Newpathname

## ブック名採番して名前変更

If Dir(変更後) <> "" Then

flag = False

q = 0

Do Until flag = True

q = q + 1

On Error Resume Next

ファイル名変更 = 格納場所① & "【未着手ファイル\_" & q & "】 " & 取込データ(u, 4)

If Dir(ファイル名変更) = "" Then

flag = True

Name Newpathname As ファイル名変更

End If

Loop

End If

## ‘名前を付けて新しく保存

Workbooks(CSVファイル(i)).SaveAs Filename:=Newpathname, FileFormat:=xlCSV

## 不正文字チェック

Function CheckName(ByVal strName As String) As Boolean

Dim strWrong As Variant

Dim i As Integer

strWrong = Array("\", "/", ":", "\*", "<", ">", "|")

For i = LBound(strWrong) To UBound(strWrong)

If InStr(strName, strWrong(i)) > 0 Then

CheckName = False

Exit Function

End If

Next

CheckName = True

End Function

'--------------- 不正文字確認 ------------------------------------------

Dim strName As String

strName = 保存ファイル名

If CheckName(strName) = False Then

MsgBox "保存ファイル名に使用できない文字があります。"

Exit Sub

End If

## シートの保護

Private Sub Auto\_open() 'ブックを閉じるときに全シート保護

Dim sh As Worksheet

Set メインST = ThisWorkbook.Worksheets("メイン")

メインST.Unprotect

メインST.Columns("D:E").EntireColumn.Hidden = True

メインST.Columns("A:A").EntireColumn.Hidden = True

メインST.Protect

End Sub

# ブックを開く

Set メインST = ThisWorkbook.Worksheets("メイン")

filepath = メインST.Range("対象ファイルパス")　＆　ブック名

If Dir(filepath) = "" Then　　　’ファイルの存在チェック

MsgBox "対象ファイルが見つかりません。ファイルパスを確認して下さい"

Exit Sub

Else

’通常

Workbooks.Open Filename:=filepath, UpdateLinks:=0

’通常変数代入

Set 対象ファイル = Workbooks.Open(新親ファイル名, UpdateLinks:=0)

'読み取り専用

Workbooks.Open fileName:="C:\Book1.xls", ReadOnly:=True

# ブック開いてるかチェック

flag = False

For Each wb In Workbooks

If wb.Name =ブック名 Then

flag = True

Exit For

End If

Next wb

If flag = False Then Workbooks.Open Filename:=格納場所 & ブック名

Set 対象ST = Workbooks(ブック名).Sheets(1)

‘-----------------------------------------------------------

Set newBook = Workbooks.Add '新しいファイルを作成

If Right(ファイル保存場所, 1) <> "\" Then ファイル保存場所 = ファイル保存場所 & "\"

'Application.DisplayAlerts = False '上書きダイアログの強制非表示

newBook.SaveAs Filename:=ファイル保存場所 & 作成BOOK名 & ".xlsm", FileFormat:=xlOpenXMLWorkbookMacroEnabled

'Application.DisplayAlerts = True '上書きダイアログの強制非表示（解除）

ThisWorkbook.Sheets(シート名).Copy After:=newBook.Sheets(newBook.Sheets.Count)

販売店進捗一覧ST.Copy before:=newBook.Sheets(1)

フォーマットST.Copy before:=newBook.Sheets(1)

Application.DisplayAlerts = False

newBook.Sheets("Sheet1").Delete

Application.DisplayAlerts = True

前回ログST.Copy before:=newBook.Sheets(1)

ログST.Copy before:=newBook.Sheets(1)

newBook.Sheets("申請ログ").Visible = True

newBook.Sheets("【前月分】申請ログ").Visible = False

# '名前の定義の確認

cnt = 0

For Each MyName In 作業用①.Names

セル名 = MyName.NameLocal

If セル名 Like "\*Print\_Area\*" = False Then

adrs = MyName.RefersToLocal

adrs = Right(adrs, Len(adrs) - InStr(adrs, "!"))

End If

Next

## シート内の全ての図を選択　画像　Shapes

　Dim shp As Shape

　For Each shp In ActiveSheet.Shapes

　　shp.Select Replace:=False

　Next

# 画面更新ストップ

Application.ScreenUpdating = False

# テキスト系

# セルの書式設定

' 取得

Dim s As String

s = Range("A1").NumberFormatLocal

' 設定

Range("A:A").NumberFormatLocal = "yyyy/m/d"

Range("A:A").NumberFormatLocal = "#,##0.0"

Range("B:B,E:F"). = "@" ‘飛び飛び複数列

'----------------------------------------------------------

‘置換　（Replace (文字列 , 検索文字列 , 置換文字列 [, 開始位置] [, 置換回数] [, 比較方法])）

strVal = "文字列"

strRet = Replace(strVal, "字", "●")　‘”文●列”

'半角全角

StrConv(対象, vbNarrow) '半角へ変換

StrConv(対象, vbwide) '全角へ変換

' 文字の配置

Range("A1").HorizontalAlignment = xlGeneral ' 横位置

Range("A1").VerticalAlignment = xlCenter ' 縦位置

Range("A1").AddIndent = False ' 前後にスペースを入れる

Range("A1").IndentLevel = 0 ' インデント

xlGeneral 1 標準

xlLeft -4131 左詰め

xlCenter -4108 中央揃え

xlRight -4152 右詰め

xlTop ‘上詰め

xlFill 5 繰り返し

xlJustify -4130 両端揃え

xlCenterAcrossSelection 7 選択範囲内で中央

xlDistributed -4117 均等割り付け

' 文字の制御

Range("A1").WrapText = False ' 折り返して全体を表示する

Range("A1").ShrinkToFit = False ' 縮小して全体を表示する

Range("A1").MergeCells = true ' セルを結合する

' 右から左

Range("A1").ReadingOrder = xlContext ' 文字の方向

Range("A1").Orientation = 0 ' 方向の角度

## 行列の幅/高さ変更

Columns("B:C").ColumnWidth = 15

Range("B:C").ColumnWidth = 20

Rows("3").AutoFit '行の幅

Columns("B").AutoFit '列の幅

d = Columns(2).ColumnWidth

d = Columns("B").Width

d = Range("A1").EntireColumn.ColumnWidth

d = Range("A1").EntireColumn.Width

d = Range("B:C").ColumnWidth ' B ～ C 列の幅を取得

d = Range(Columns(2), Columns(3)).ColumnWidth ' B ～ C 列の幅を取得

d = Range("D:D").ColumnWidth ' D 列の幅を取得

d = Range("B:C").Width ' B ～ C 列の幅の合計を取得

Rows("2:3").RowHeight = 20

Range("A1").EntireRow.RowHeight = 20

d = Rows(2).RowHeight

d = Range("2:3").RowHeight ' 2 ～ 3 行目の高さを取得

d = Range("4:4").RowHeight ' 4 行目の高さを取得

d = Range("2:3").Height ' 2 ～ 3 行目の高さの合計を取得

## フォントの変更

Set f = Range("A1").Font

' 設定

Range("A1").Font.Color = RGB(255, 0, 0) ' 文字色

Range("A1").Font.Name = "ＭＳ Ｐゴシック" ' 名前

Range("A1").Font.Size = 11 ' サイズ

Range("A1").Font.Bold = True ' 太字

## '罫線を引く

| **引数** | **線の場所** |
| --- | --- |
| — | 上下左右 |
| xlDiagonalDown | 斜線(左上から右下) |
| xlDiagonalUp | 斜線(右上から左下) |
| xlEdgeBottom | 下 |
| xlEdgeLeft | 左 |
| xlEdgeRight | 右 |
| xlEdgeTop | 上 |
| xlInsideHorizontal | 内側(横) |
| xlInsideVertical | 内側(縦) |

Range("A1").Borders.LineStyle = xlContinuous '通常線で格子を引く

Range("A1"). Borders().Weight = xlHairline 　'極細線で格子を作成

Range("B2:F11").BorderAround Weight:=xlMedium　‘外枠を中太線

Range("A1").Borders(xlEdgeTop).LineStyle = xlDouble

Range("A1").Borders(xlEdgeLeft).LineStyle = xlLineStyleNone

Range("A1").Borders.LineStyle = xlContinuous ' 種類

Range("A1").Borders.Weight = xlThin ' 太さ

Range("A1").Borders(xlEdgeLeft).LineStyle = xlDashDot

Selection.Borders(xlInsideHorizontal) .LineStyle = xlDash’選択した範囲の中間横線に点線を引く

(xlEdgeBottom)

## ‘罫線の種類

xlContinuous 実線(細)

xlDash 破線

xlDashDot 一点鎖線

xlDashDotDot 二点鎖線

xlDot 点線

xlDouble 二重線

xlSlantDashDot 斜め斜線

xlLineStyleNone 無し

xlHairline 極細

xlThin 細

xlMedium 中

xlThick 太

xlNone ‘消す

## '塗りつぶし

' 取得

Dim l As Long

Dim i As Integer

l = Range("A1").Interior.Color ' 背景色

i = Range("A1").Interior.ColorIndex ' 背景色番号

i = Range("A1").Interior.Pattern ' パターン

l = Range("A1").Interior.Pattern.Color ' パターンの色

i = Range("A1").Interior.Pattern.ColorIndex ' パターンの色番号

' 設定

Range("A1").Interior.Color = RGB(255, 0, 0) ' 背景色

Range("A1").Interior.ColorIndex = 3 ' 背景色番号

Range("A1").Interior.Pattern = xlPatternGray50 ' パターン

Range("A1").Interior.Pattern.Color = RGB(0, 255, 0) ' パターンの色

Range("A1").Interior.Pattern.ColorIndex = 4 ' パターンの色番号

## 行・列の再表示/非表示

'--- 全ての行を再表示する（Rowsを使うケース） ---'

ws.Rows.Hidden = False

'--- 全ての行を再表示する（Cells.EntireRowを使うケース） ---'

ws.Cells.EntireRow.Hidden = False

'--- 全ての列を再表示する（Columnsを使うケース） ---'

ws.Columns.Hidden = False

'--- 全ての列を再表示する（Cells.EntireColumnを使うケース） ---'

ws.Cells.EntireColumn.Hidden = False

## 結合されたセルの場所取得

Sub test()

If Cells(5, i).MergeCells And InStr(Range(tt).Name, "科目") > 0 Then

With Cells(5, i).MergeArea

msg = msg & "結合範囲：" & .Address & vbCrLf

msg = msg & "大きさ：" & .Count & vbCrLf

msg = msg & "行数：" & .Rows.Count & vbCrLf

msg = msg & "列数：" & .Columns.Count & vbCrLf

msg = msg & "左上セル：" & .Item(1).Address & vbCrLf

msg = msg & "右下セル：" & .Item(.Count).Address

End With

End If

End Sub

## ウィンドウ枠の固定

‘選択セルをアクティブにしてから

ActiveWindow.FreezePanes = True

# 条件付き書式設定

## 条件の設定

Sub FormatCollectionsAddStringTest()

Dim r1 As Range, r2 As Range, r3 As Range

Dim f1 As FormatCondition, f2 As FormatCondition, f3 As FormatCondition

最終行 = Cells(Rows.Count, 3).End(xlUp).Row + 2

'// 対象範囲指定

Set r1 = Range("$P$4:$Q$" & 最終行)

Set r2 = Range("$D$4:$O$" & 最終行)

Set r3 = Range("$G$4:$N$" & 最終行)

'// 条件付き書式の追加（A2セルが入力有り　＆　D2セルが空白の場合）

Set f3 = r3.FormatConditions.Add(Type:=xlExpression, Formula1:=" =AND($A2<>"""",$D2="""")")

'// フォント太字、文字色、背景色

f3.Font.Bold = True

f3.Font.Color = RGB(192, 0, 0)　　‘濃い赤

f3.Interior.Color = RGB(255, 204, 204)　‘ピンク

f3.Borders.LineStyle = xlContinuous　‘罫線

End sub

‘例②------------------------------------------------------------------

'// 条件付き書式の追加（セルが空白でない場合）Sinaps入力

Set f3 = r3.FormatConditions.Add(Type:=xlExpression, Formula1:="=AND($A2<>"""",$D2="""")")

'// フォント太字、文字色、背景色

f3.Font.Bold = True

f3.Font.Color = RGB(192, 0, 0)

f3.Interior.Color = RGB(255, 204, 204)

f3.Borders.LineStyle = xlContinuous

‘例①------------------------------------------------------------------

For m = 3 To 2 + xCount Step 2

xRange = linkST.Range(Cells(m, 5), Cells(m + 1, 5)).Address

xRange① = Right(Left(xRange, InStr(xRange, ":") - 1), Len(Left(xRange, InStr(xRange, ":") - 1)) - 1)

xRange② = Right(Right(xRange, Len(xRange) - InStr(xRange, ":")), Len(Right(xRange, Len(xRange) - InStr(xRange, ":"))) - 1)

linkST.Range(xRange②).FormatConditions.Add Type:=xlExpression, Formula1:="=" & xRange① & "<>" & xRange②

linkST.Range(xRange②).FormatConditions(linkST.Range(xRange②).FormatConditions.Count).SetFirstPriority

With linkST.Range(xRange②).FormatConditions(1).Interior

.PatternColorIndex = xlAutomatic

.ThemeColor = xlThemeColorAccent4

.TintAndShade = 0.799981688894314

End With

linkST.Range(Cells(m + 1, 2), Cells(m, 14)).Borders(xlEdgeBottom).Weight = xlThick '太い線で格子を引く

linkST.Range(Cells(m, 4), Cells(m, 14)).Interior.Color = RGB(219, 242, 249) '上段薄青

linkST.Range(Cells(m + 1, 4), Cells(m + 1, 4)).Interior.Color = RGB(255, 242, 204) '下段タイトルのみ薄い黄色

linkST.Range(xRange).Copy

linkST.Range(Cells(m, 5), Cells(m + 1, 14)).PasteSpecial Paste:=xlPasteFormats '書式のコピペ

Application.CutCopyMode = False

Next

## 条件付き書式の削除

Cells.FormatConditions.Delete　‘全削除

Range(A1).FormatConditions(1).Delete　‘一部削除 ※条件上から(1)

# 保護・プロテクト

## セルのロック

Range（”［セル範囲］”）.Locked = True/False

' 取得

Dim b As Boolean

b = ActiveSheet.ProtectContents ' シートが保護されているか

b = Range("A1").Locked

' 設定

Call ActiveSheet.Protect(UserInterfaceOnly:=True) ' シートを保護する

ActiveSheet.Unprotect ' シートの保護を解除する

Range("A1").Locked = True ' セルをロックする

Rows(2).Hidden = Ture '行の非表示

Columns("B:B").Hidden = True ' B 列を非表示 '行の非表示

‘--------------------------------------------

Sub 全シートプロテクト()

For Each ws In ThisWorkbook.Sheets

ws.Protect

Next

End Sub

## ‘シート保護特定条件

ws.Protect UserInterfaceOnly:=True, AllowFormattingCells:=True, AllowFormattingColumns:=True, AllowFormattingRows:=True, AllowInsertingHyperlinks:=True, AllowSorting:=True, AllowFiltering:=True, AllowUsingPivotTables:=True

## ‘----------イミディエイトウィンドウ-------------

for each ws in sheets:ws.protect:next

‘--------------------------------------------

Password 保護を解除するためのパスワードを設定します。半角英数時を指定します。

Contents オブジェクトの内容を保護させるには、True を指定します。対象はワークシートの場合はロックされているセルです。 False

Scenarios シナリオを保護するには、True を指定します。 True

UserInterfaceOnly:=True を指定すると、画面上からの変更は保護されますが、マクロからの変更は保護されません。この引数を省略すると、マクロからも、画面上も変更することができなくなります。 False

AllowFormattingCells:=True を指定すると、セルの書式設定ができます。 False

AllowFormattingColumns:=True を指定すると、列の書式設定ができます。 False

AllowFormattingRows:=True を指定すると、行の書式設定ができます。 False

AllowInsertingColumns:=True を指定すると、列を挿入できます。 False

AllowInsertingRows:=True を指定すると、行を挿入できます。 False

AllowInsertingHyperlinks:=True を指定すると、ハイパーリンクを挿入できます。 False

AllowDeletingColumns:=True を指定すると、列を削除でき、削除される列のセルはすべてロック解除されます。 False

AllowDeletingRows:=True を指定すると、行を削除でき、削除される行のセルはすべてロック解除されます。 False

AllowSorting:=True を指定すると、並べ替えができます。並べ替え範囲内のセルは、ロックと保護が解除されている必要があります。 False

AllowFiltering:=True を指定すると、フィルタを設定できます。ユーザーは、フィルタ条件を変更できますが、オート フィルタの有効と無効を切り替えることはできません。 False

AllowUsingPivotTables:=True を指定すると、ピボットテーブル レポートを使用できます。 False

# ハイパーリンク

## '同じドキュメント内のシート

ActiveSheet.Hyperlinks.Add anchor:=Range("A" & i), Address:="", SubAddress:=Worksheets(i).Name & "!A1", TextToDisplay:=Worksheets(i).Name

## 'セルへのハイパーリンク設定

Set hyplink = ActiveSheet.Hyperlinks.Add(Anchor:=Range("C" & i), Address:=Range("D" & i))

# 配列関係

## 多次元配列の1行のみを一括代入

Array2D = Range("A1:E5")

'■二次元配列の指定行(2行目)を一次元配列に格納する

Array1D = WorksheetFunction.Index(Array2D, 2)

'■二次元配列の指定列(A列(1列目))を一次元配列に格納する

Array1D = WorksheetFunction.Index(WorksheetFunction.Transpose(Array2D), 1)

Array1D = WorksheetFunction.Transpose(Array1D)

Range(Cells(8, 1), Cells(18, 1)) = Array1D

# シート

## シートの追加

Worksheets.Add(After:=Worksheets(Worksheets.Count)).Name =“新規シート”

## シートのコピー

Set フォーマット = ブック.Worksheets("フォーマット")

フォーマット.Copy after:=ブック.Worksheets(ブック.Sheets.Count)

ActiveSheet.Name = 名前

## シート名重複してたら採番

On Error Resume Next

作業ST.Name = “シート名”

On Error GoTo 0

If 作業ST.Name <> “シート名" Then

flag = False

q = 1

Do Until flag = True

q = q + 1

On Error Resume Next

シート名変更 = “シート名(" & q & ")"

作業ST.Name = シート名変更

On Error GoTo 0

If 作業ST.Name = シート名変更 Then flag = True

Loop

End If

## シートの再表示/非表示

Set 作業ST = 集計表BK.Worksheets(集計表BK.Sheets.Count)

作業ST.Visible = True　‘再表示

## シート名変更不可設定

Private Sub worksheet\_SelectionChange(ByVal Target As Excel.Range)

'Updateby Extendoffice

If ActiveSheet.Name <> "シート名" Then

ActiveSheet.Name = "シート名"

End If

End Sub

## ‘確認メッセージ無視

Application.DisplayAlerts = False

　　ActiveSheet.Delete

Application.DisplayAlerts = True

## ソート

メインST.Activate

最終行 =メインST .Cells(Rows.Count, 2).End(xlUp).Row

Set 範囲 = メインST.Range(Cells(メイン項目行, 1), Cells(最終行, 9))

範囲.Sort key1:=Range("D2"), order1:=xlAscending, key2:=Range("E2"), order2:=xlAscending, Header:=xlGuess

・xlAscending　‘昇順

・xlDescending ‘降順

## オートフィルタ

‘------------------ 設定 On/Off ------------------------------

If メインST.AutoFilterMode = True Then メインST.AutoFilterMode = False

メインST.Rows(2).AutoFilter

‘------------------ フィルタ解除(全データ表示) -------------------

If メインST.FilterMode = True Then メインST. ShowAllData

‘-------------- フィルタ ------------------------------------

設定したい表のセル(一か所).AutoFilter Field:=対象の番号, Criteria1:="条件"　‘フィルタ

Range("A2").AutoFilter Field:=行番号, Criteria1:=">=80"

‘省略ver

Set 範囲 = メインST.Range(Cells(4, 1), Cells(最終行, 17))

範囲.AutoFilter 17,月

‘-----------フィルタ設定ON/OFF判定 ---------------------------

If ActiveSheet.AutoFilterMode = True Then

MsgBox "設定されています"

Else

MsgBox "設定されていません"

End If

‘------------フィルタ絞り込みされてるか判定 -----------------------

If ActiveSheet.FilterMode = True Then

MsgBox "絞り込まれています"

Else

MsgBox "絞り込まれていません"

End If

‘---------- 絞り込み解除（全データ表示） ---------------------------

If ActiveSheet.AutoFilterMode = True Then　 ActiveSheet.ShowAlldata

## コピー＆ペースト

Range("B1").CurrentRegion.Copy

Application.CutCopyMode = False　‘解除

Range("E1").PasteSpecial Paste:=xlPasteValues '値

Application.CutCopyMode = False

-----------------------------------------

xlPasteAll ‘すべて（既定） -4104

xlPasteFormulas ‘数式 -4123

xlPasteValues　‘値 -4163

xlPasteFormats ‘書式 -4122

xlPasteComments 　‘コメント　-4144

xlPasteValidation 6 ‘入力規則 2002以降(※)

罫線を除く全て xlPasteAllExceptBorders 7

列幅 xlPasteColumnWidths 8 2002以降(※)

数式と数値の書式 xlPasteFormulasAndNumberFormats 11 2002以降

値と数値の書式 xlPasteValuesAndNumberFormats 12 2002以降

コピー元のテーマを使用してすべて貼り付け xlPasteAllUsingSourceTheme 13 2007以降

すべての結合されている条件付き書式 xlPasteAllMergingConditionalFormats 14 2010以降

## クリア

result = MsgBox("「データ貼り付け」シートの内容をクリアしますか？", vbYesNo)

If result = vbNo Then Exit Sub

Set データST = ThisWorkbook.Worksheets("データ貼り付け")

最終行 = データST.Cells(Rows.Count, 2).End(xlUp).Row

If 最終行 > 2 Then データST.Range(Rows(3), Rows(最終行 + 1)).Delete

## 印刷範囲

With ActiveSheet

'B2のアクティブセル領域を印刷範囲に設定(A1形式の文字列で指定)

.PageSetup.PrintArea = Range("B2").CurrentRegion.Address

.PrintPreview

'印刷範囲の設定を解除。シート全体が印刷範囲になるbot

.PageSetup.PrintArea = ""

.PrintPreview

End With

## 配列内での最大値取得

Redim 配列arr(3,1) ‘配列内のデータを削除

配列arr(0, 0) = x数量

配列arr(0, 1) = x店名

fmaxbmi = ""

fmaxbmi = Application.WorksheetFunction.Max(配列arr)

For pp = 0 To 2

If 配列arr(pp, 0) = fmaxbmi Then

最大数店舗名= 配列arr(pp, 1)

Exit For

Endif

Next

# データ取得

## VLOOKUP

Set SerchRange = メインST.Range("A:F")

SerchKey = 出荷伝票.Range("販売店コード") & 出荷伝票.Cells(行, 1)

On Error Resume Next

単価 = WorksheetFunction.VLookup(SerchKey, SerchRange, 6, False)

On Error GoTo 0

## ファイルの最終更新日時取得

Dim a As Date

a = FileDateTime("C:\Windows\Win.ini")

MsgBox a & vbCrLf & \_

Year(a) & "年" & vbCrLf & \_

Month(a) & "月" & vbCrLf & \_

Day(a) & "日" & vbCrLf & \_

Hour(a) & "時" & vbCrLf & \_

Minute(a) & "分" & vbCrLf & \_

Second(a) & "秒" & vbCrLf & \_

Format(a, "aaaa") & vbCrLf & \_

Format(a, "ggge年m月d日")

‘---------------------------------

Set f = "D:\Tips.txt"

d = f.DateCreated ' 作成日時を取得

d = f.DateLastModified ' 更新日時を取得

d = f.DateLastAccessed ' アクセス日時を取得

## フォルダパス・ファイル名取得

‘フルパスからファイル名抜く

フォルダパス = Left(ThisWorkbook.FullName, InStrRev(ThisWorkbook.FullName, "\"))

ファイル名 = Mid(ThisWorkbook, InstrRev(ThisWorkbook, "\") + 1)

‘-----------------------------------------------------

Dim A

A = "C:\Users\User\Desktop\TEST\TEST.xlsm"

Dim FSO

Set FSO = CreateObject("Scripting.FileSystemObject")

'ファイルのフォルダパス

フォルダパス= FSO.GetParentFolderName(A)

## ファイルパス（拡張子抜き）

Dim A

A = "C:\Users\User\Desktop\TEST\TEST.xlsm"

Dim FSO

Set FSO = CreateObject("Scripting.FileSystemObject")

'拡張子を除くファイル名

ファイル名= FSO.GetBaseName(A)

## 最終行・最終列

最終列 = Cells(●, Columns.Count).End(xlToLeft).Column

最終行 = Cells(Rows.Count, ●).End(xlUp).Row

## 配列内でのソート

‘中身が数字の一次元の場合昇順で並べ替える

For i = 0 To 9

srtArray(i) = WorksheetFunction.Small(orgArray, i + 1)

Next i

## 指定のセルの内容が変更されたら

Private Sub Worksheet\_Change(ByVal Target As Range)

If Intersect(Target, Range("Q2")) Is Nothing Then

Exit Sub

Else

MsgBox "セルの値が変更されました"

End If

End Sub

## 外部リンク先の取得

aLinks = ActiveWorkbook.LinkSources(xlExcelLinks)

If Not IsEmpty(aLinks) Then

For i = 1 To UBound(aLinks)

MsgBox "Link " & i & ":" & Chr(13) & aLinks(i)

Next i

End If

# テキストボックス

‘標準モジュール(「ActiveSheet」で選択)

値 = ActiveSheet.TextBox1

# 文言チェック

If 項目名 <> "" Then

項目名 = StrConv(項目名, vbNarrow)

項目名 = Replace(項目名, " ", "")

項目名 = Replace(項目名, "　", "")

項目名 = WorksheetFunction.Clean(項目名)

カウント項目(i - 2, ii - 1) = 項目名

End If

# セル名から列番号の取得

項目名①列 = Range(メインST.txt項目名①セル.Text).Column

項目名②列 = Range(メインST.txt項目名②セル.Text).Column

adr = Cells(１, 1).Address ‘A1

# 存在確認

## ファイル存在確認

n = 1

tmp= folderPath & "\" & ファイル名 & “.xlsx”

If Dir(tmp) <> "" Then

Do While Dir(tmp) <> ""

n = n + 1

tmp = folderPath & "\" & ファイル名 & "(" & n & ")" & ".xlsx”

Loop

End If

# フォルダ作業

## フォルダを開く

格納場所 = メインST.txt格納先.Text

Shell "C:\Windows\Explorer.exe " & 格納場所, vbNormalFocus

## フォルダ存在確認

Function FolderExists(folder\_path As String) As Boolean

If Dir(folder\_path, vbDirectory) = "" Then

FolderExists = False

Else

FolderExists = True

End If

End Function

'------------------------------------------------

Dim 格納場所 As String

格納場所 = メインST.TextBox2.Text

'If Right(格納場所, 1) <> "\" Then 格納場所 = 格納場所 & "\"

If FolderExists(格納場所) = False Then

MsgBox "ファイル保存先パスが不正です。確認してください。"

Exit Sub

End If

'------ ファイル存在確認 ------------------------------------

filepath = "C:\Users\US525182\Desktop\Book1.xlsx"

If Dir(filepath) = "" Then

MsgBox "対象ファイルが見つかりません。ファイルパスを確認して下さい"

Exit Sub

End If

## フォルダ内のファイル名を取得する

Dim strFilename As String

cnt = 0

格納場所 = メインST.txtCSV格納場所.Text

If Right(格納場所, 1) <> "\" Then 格納場所 = 格納場所 & "\"

strFilename = Dir(格納場所, vbNormal)

' ファイルが見つからなくなるまで繰り返す

Do While strFilename <> ""

If InStr(strFilename, ".csv") > 0 Then CSVファイル(cnt) = strFilename

strFilename = Dir()

cnt = cnt + 1

Loop

## フォルダのコピー

Sub フォルダコピー ()

Dim FSO As Object

Set FSO = CreateObject("Scripting.FileSystemObject")

''C:\Workフォルダ内の全ファイルをC:\Tmpフォルダにコピーします

FSO.GetFolder("C:\Work").Copy "C:\Tmp"

''C:\WorkフォルダをC:\Tmpフォルダのサブフォルダとしてコピーします（フォルダごとコピー）

FSO.GetFolder("C:\Work").Copy "C:\Tmp\"

Set FSO = Nothing

End Sub

# '警告メッセージを表示/非表示

Application.DisplayAlerts = True　 '警告メッセージを表示

# メッセージボックス

## エラー有の時のmsgbox

If エラー <> "" Then

Unload Me

メインST.txtエラー = "処理エラー有り" & Format(Now, "yyyy/m/dd h:mm") & vbCrLf & エラー

MsgBox "エラーあり。作業完了"

Else

Unload Me

メインST.txtエラー = "正常に処理終了" & Format(Now, "yyyy/m/dd h:mm")

MsgBox "正常に完了"

End If

## 改行コード

& vbLf &　　　‘セル内改行

& vbCrLf &　　‘msgbox内改行

## YesNo

result = MsgBox("データ抽出開始しますか？", vbYesNo)

If result = vbNo Then Exit Sub

## マーク表示

ans = MsgBox("実行しますか？", vbOKCancel + vbQuestion, "テスト")

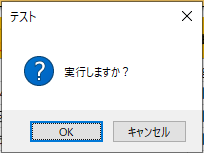
If ans = vbOK Then

Range("A1").Value = "OKが押されました"

Else

Range("A1").Value = "キャンセルが押されました"

End If



|  |  |  |
| --- | --- | --- |
| **定義済み定数** | **実際の値** | **説明** |
| vbCritical | 16 | 警告 |
| vbQuestion | 32 | 問い合わせ |
| vbExclamation | 48 | 注意 |
| vbInfomation | 64 | 情報 |

# 外部リンク解除

Dim astrLinks As Variant

このBOOK = ThisWorkbook.Name

新しいBOOK = newBook.Name

newBook.ChangeLink Name:=このBOOK, NewName:=新しいBOOK, Type:=xlExcelLinks

' 変数を Excel リンク タイプとして定義します。

astrLinks = ActiveWorkbook.LinkSources(Type:=xlLinkTypeExcelLinks)

' アクティブ ブックリンクを解除します。

'ActiveWorkbook.BreakLink Name:=astrLinks(1), Type:=xlLinkTypeExcelLinks

# 自動実行

**★【Thisworkbook】シート上に記述する**

## ブックを閉じるとき

Private Sub Workbook\_BeforeClose(Cancel As Boolean)

　　'処理を記述

End Sub

## ブックを開いたとき

Workbook\_Open

　　'処理を記述

End Sub

# ユーザーフォーム操作

Userform1.show

'------------

Private Sub UserForm\_Activate()

Me.Repaint

Unload Me

# ファイル情報取得【FileSystemObject】

## ファイル日時取得

DateCreated 作成日時を取得します。

DateLastModified 更新日時を取得します。

DateLastAccessed アクセス日時を取得します。

‘--------------------------------------

Dim FSO As Object

Dim MyFile As Object

Dim FilePath As String

FilePath = "C:\Sample\Test01.xlsx" 'フォルダとファイルを指定

Set FSO = CreateObject("Scripting.FileSystemObject") 'FSOをセット

Set MyFile = FSO.GetFile(FilePath) 'ファイルを取得

Debug.Print (MyFile.DateCreated) '作成日時

Debug.Print (MyFile.DateLastModified) '更新日時

Debug.Print (MyFile.DateLastAccessed) 'アクセス日時

Set FSO = Nothing

‘--------------------------------------

## 拡張子の取得

Dim fso As Object

Dim extension As String

Set fso = CreateObject("Scripting.FileSystemObject")

fileFullPath = "C:\Users\user\Desktop\aiueo.txt"　'ファイルのパスを指定

extension = fso.GetExtensionName(fileFullPath)　 '拡張子を取得

Set fso = Nothing　'後片付け

# クリップボードにコピー

Private Sub CommandButton4\_Click()

Text = ThisWorkbook.Worksheets("メイン").TextBox3.Text

CopyText (Text)

End Sub

‘--------------------------------------

Function CopyText(ByVal Text As String) As Long

Dim MSFDO As MSForms.DataObject

Set MSFDO = New MSForms.DataObject ' (1)

If Text <> "" Then ' (2)

MSFDO.SetText Text ' (3)

'If MsgBox(MSFDO.GetText & " をクリップボードにコピーしますか？", \_

' vbInformation + vbYesNo, "確認") = vbYes Then ' (4)

MSFDO.PutInClipboard ' (5)

CopyText = 1

End If

'End If

Set MSFDO = Nothing

End Function

# テキストボックス　カンマ自動入力

Private Sub txt金額\_Exit(ByVal Cancel As MSForms.ReturnBoolean)

If Me.txt金額.Text = "" Then

Exit Sub

End If

If IsNumeric(Me.txt金額.Text) Then

Me.txt金額.Text = Format(Me.txt金額.Text, "#,##0")

Else

MsgBox "「前回請求金額」には半角数値のみを入力して下さい。"

Cancel = True

End If

End Sub

# コンボボックス　自動追加

Private Sub UserForm\_Initialize()

Dim i As Long

'年のコンボボックス　去年から10年間

For i = Year(Date) - 1 To Year(Date) + 10

ComboBox1.AddItem i

Next

'初期値は前月

If Month(Date) - 1 = 0 Then

ComboBox2.Value = 12

'初期値は現在の年

ComboBox1.Value = Year(Date) - 1

Else

ComboBox2.Value = Month(Date)

'初期値は現在の年

ComboBox1.Value = Year(Date)

End If

End Sub

# フォルダ選択ダイアログ

Private Sub CommandButton2\_Click(ByVal Cancel As MSForms.ReturnBoolean)

With Application.FileDialog(msoFileDialogFolderPicker)

.InitialFileName = txtPath.Value

If .Show = True Then

Me.txtPath.Value = .SelectedItems(1)

End If

End With

End Sub

# ユーザーフォーム

## コンボボックスへ項目追加

Private Sub UserForm\_Initialize()

Dim i As Integer

i = 4

cnt = 0

For Each ii In MS.Names

If InStr(ii.Name, "科目") > 0 Then

'Debug.Print (ii.Name)

If MS.Range(ii.Name) <> "" Then

ComboBox1.AddItem MS.Range(ii.Name)

End If

End If

Next ii

‘--------------------------------------------------------

Sub main ()

If ComboBox1.Text = "" Then

MsgBox "作成する「勘定科目」をプルダウンより一つ選択して下さい"

Exit Sub

End If

End sub