CodePrinter.xlsm

www.github.com/alexofrhodes

--- Table Of Contents ---

```
--- ThisWorkbook ---
```

```
   --- PrintPage - Sheet9 ---
```

```
--- PrintSettings - Sheet12 ---
```

--- README - Sheet22 ---

--- Home ---

```
Sub Main()
    uCodePrinter.Show
End Sub

Sub CodePrinterButtonClicked(control As IRibbonControl)
    Main
End Sub
```

--- uCodePrinter ---

```
' Author    Anastasiou Alex
' Project   CodePrinter
' Purpose   Export active project's code as PDF. Code bl
'ocks linked by shape. Keywords colored. Oddlines colored.
' Website   https://github.com/alexofrhodes
' Copyright MIT License 2021 Anastasiou Alex
'
' Revision History:
' #  yyyy-mm-dd  COMMENTS
' 1  2021-08-05  Initial Release
'
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
Private SaveLocation As String
Public PrintFileName As String
Public Found1 As String
Public Found2 As String
Dim rng As Range
Public cell As Range
Public s As Shape
Public counter As Long
Dim mafChrWid(32 To 127) As Double
Dim msFontName As String

Public Function PrintProject(wb As Workbook)
    If ProtectedVBProject(wb) = True Or HasProject(wb) = False Then
        MsgBox "Project Empty or Protected"
        Exit Function
    End If
    'ThisWorkbook.Application.Visible = False
    'ThisWorkbook.IsAddin = False
    Dim vbComp As VBComponent
    ResetPrinter
    Dim tmpString As Variant
    Dim i As Long
    Dim Procedures As Collection
    Set Procedures = New Collection
    Dim ws As Worksheet
    Dim wsName As String
    'Table of contents
    Procedures.Add "--- Table Of Contents ---" & vbNewLine & vbNewLine
    'document
    For Each vbComp In wb.VBProject.VBComponents
        If vbComp.Type = vbext_ct_Document Then
            For Each ws In wb.Worksheets
                If ws.CodeName = vbComp.Name Then wsName = ws.Name
            Next ws
            If vbComp.Name <> "ThisWorkbook" Then
                Procedures.Add "(" & ComponentTypeToString(vbComp.Type) & _
                ")" & " " & wsName & " - " & vbComp.Name
            Else
```

```
                Procedures.Add "(" & ComponentTypeToString(vbComp.Type) & _
                ")" & " " & vbComp.Name
            End If
            wsName = ""
        End If
    Next vbComp
    'class
    For Each vbComp In wb.VBProject.VBComponents
        If vbComp.Type = vbext_ct_ClassModule Then
            Procedures.Add "(" & ComponentTypeToString(vbComp.Type) & ")" _
            & " " & vbComp.Name
        End If
    Next vbComp
    'module
    For Each vbComp In wb.VBProject.VBComponents
        If vbComp.Type = vbext_ct_StdModule Then
            Procedures.Add "(" & ComponentTypeToString(vbComp.Type) & ")" _
            & " " & vbComp.Name
        End If
    Next vbComp
    'userform
    For Each vbComp In wb.VBProject.VBComponents
        If vbComp.Type = vbext_ct_MSForm Then
            Procedures.Add "(" & ComponentTypeToString(vbComp.Type) & ")" _
            & " " & vbComp.Name
        End If
    Next vbComp
    'Code of components
    'document
    For Each vbComp In wb.VBProject.VBComponents
        If vbComp.Type = vbext_ct_Document Then
            'get sheet name
            For Each ws In wb.Worksheets
                If ws.CodeName = vbComp.Name Then wsName = ws.Name
            Next ws
            If vbComp.Name <> "ThisWorkbook" Then
                Procedures.Add "--- " & wsName & " - " & vbComp.Name & " - _
                --"
            Else
                Procedures.Add "--- " & vbComp.Name & " ---"
            End If
            wsName = ""
            If vbComp.CodeModule.CountOfLines > 0 Then
                tmpString = Split(GetCompText(vbComp), vbNewLine)
                For i = LBound(tmpString) To UBound(tmpString)
                    Procedures.Add " " & tmpString(i)
                Next i
            End If
        End If
    Next vbComp
    'class
    For Each vbComp In wb.VBProject.VBComponents
```

```vb
        If vbComp.Type = vbext_ct_ClassModule Then
            Procedures.Add "--- " & vbComp.Name & " ---"
            If vbComp.CodeModule.CountOfLines > 0 Then
                tmpString = Split(GetCompText(vbComp), vbNewLine)
                For i = LBound(tmpString) To UBound(tmpString)
                    Procedures.Add " " & tmpString(i)
                Next i
            End If
        End If
    Next vbComp
    'module
    For Each vbComp In wb.VBProject.VBComponents
        If vbComp.Type = vbext_ct_StdModule Then
            Procedures.Add "--- " & vbComp.Name & " ---"
            If vbComp.CodeModule.CountOfLines > 0 Then
                tmpString = Split(GetCompText(vbComp), vbNewLine)
                For i = LBound(tmpString) To UBound(tmpString)
                    Procedures.Add " " & tmpString(i)
                Next i
            End If
        End If
    Next vbComp
    'userform
    For Each vbComp In wb.VBProject.VBComponents
        If vbComp.Type = vbext_ct_MSForm Then
            Procedures.Add "--- " & vbComp.Name & " ---"
            If vbComp.CodeModule.CountOfLines > 0 Then
                tmpString = Split(GetCompText(vbComp), vbNewLine)
                For i = LBound(tmpString) To UBound(tmpString)
                    Procedures.Add " " & tmpString(i)
                Next i
            End If
        End If
    Next vbComp
    tmpString = CollectionToArray(Procedures)
    ThisWorkbook.Sheets("PrintPage").Range("B1:B" & UBound(tmpString) + 1) _
    .Value = WorksheetFunction.Transpose(tmpString)
    If CodePrinter Then PrintPDF
ErrorHandler:
End Function

Function CodePrinter() As Boolean
    ThisWorkbook.Sheets("PrintPage").Cells.Font.Name = "Consolas"
    RemoveBreaks
    BreakText
    NumberLinesPrinter
    ChgTxtColor
    GreenifyComments
    BoldPrinterComponents
    If findPairs = False Then
        CodePrinter = False
        Exit Function
    End If
    PrinterPageSetup
    ThisWorkbook.Sheets("PrintPage").rows(1).EntireRow.Insert
    copyLOGOPrinter
    ShapesCompareLeft
    PageBreaksInPrinter
    CodePrinter = True
End Function

Function findPairs() As Boolean
    Dim ShapeTypeNumber As Long
    ShapeTypeNumber = 29
    Dim CloseTXT As String
    Dim X As Variant
    Dim ws As Worksheet
    Set ws = ThisWorkbook.Sheets("PrintPage")
    Dim trimCell As String
    Dim counter As Long
    For Each cell In ThisWorkbook.Sheets("PrintPage").Range("B:B"). _
    SpecialCells(xlCellTypeConstants)
        trimCell = Trim(cell.Text)
        If IsBlockStart(trimCell) Then
            Select Case openPair(trimCell)
            Case Is = "Case", "Else"
                GoTo Skip
            Case Is = "If", "#If"
                If Right(trimCell, 4) = "Then" Then        'Or  _
                Right(trimCell, 1) = "_" Then
                'ok
            Else
                GoTo Skip
            End If
        Case Is = "skip"
            GoTo Skip
        Case Else
            '
        End Select
        CloseTXT = closePair(trimCell)
        counter = Len(cell) - Len(trimCell)
        Found1 = cell.Address
        If FOUND2FOUND(ws, WorksheetFunction.Rept(" ", counter) &  _
        CloseTXT) = False Then
            GoTo Skip
            '                 MsgBox "Cod
            'e not properly indented." & vbNewLine & _
            '                        "Error w
            'ith closing pair of " & vbNewLine & cell.Text
            '                 findPairs = False
            '                 Exit Function
        End If
        Found2 = ws.Range("B1:B" & ws.Cells(rows.Count, 2).End(xlUp).Row)  _
        _
        .Find(WorksheetFunction.Rept(" ", counter) & CloseTXT & "*",  _
        After:=cell, LookAt:=xlWhole).Address
        X = StrWidth(Application.WorksheetFunction.Rept("A", counter),  _
        "Consolas", 11)
        ws.Shapes.AddShape ShapeTypeNumber, ws.Range(Found1).Left + X -  _
        10, ws.Range(Found1).Top + (cell.Height / 2), 5, Range(Found1,  _
        Found2).Height - cell.Height
    End If
Skip:
Next cell
findPairs = True
End Function

Function FOUND2FOUND(ws As Worksheet, str As String) As Boolean
    FOUND2FOUND = True
    Dim tmp As Range
    Set tmp = ws.Range("B1:B" & ws.Cells(rows.Count, 2).End(xlUp).Row) _
    .Find(str & "*", After:=cell, LookAt:=xlWhole)
    If tmp Is Nothing Then FOUND2FOUND = False
End Function

Function openPair(strLine As String) As String
    Dim nPos As Integer
    Dim strTemp As String
    strTemp = Trim(strLine)
    nPos = InStr(1, strTemp, " ") - 1
    If nPos < 0 Then nPos = Len(strLine)
    strTemp = Left$(strLine, nPos)
    Select Case strTemp
    Case Is = "Private", "Public"
        strTemp = Trim(strLine)
        strTemp = Replace(strTemp, "Private ", "")
        strTemp = Replace(strTemp, "Public ", "")
        nPos = InStr(1, strTemp, " ") - 1
        If nPos < 0 Then nPos = Len(strTemp)
        strTemp = Left$(strTemp, nPos)
        If strTemp = "Function" Then
            openPair = "Function"
        ElseIf strTemp = "Sub" Then
            openPair = "Sub"
        Else
            GoTo Skip
        End If
    Case Is = "With"
        openPair = "With"
    Case Is = "For"
        openPair = "For"
    Case Is = "Do"
        openPair = "Do"
    Case Is = "While"
        openPair = "While"
    Case Is = "Select"
```

```
            openPair = "Select"
        Case Is = "Case"
            openPair = "Case"
        Case Is = "Sub"
            openPair = "Sub"
        Case Is = "Function"
            openPair = "Function"
        Case Is = "Property"
            openPair = "Property"
        Case Is = "Enum"
            openPair = "Enum"
        Case Is = "Type"
            openPair = "Type"
        Case "If", "#If"
            openPair = "If"
        Case "ElseIf", "#ElseIf", "Else", "Else:", "#Else", "#Else:"
            openPair = "Else"
        Case Else
    Skip:
            openPair = "skip"
        End Select
    End Function

    Function closePair(strLine As String) As String
        Dim nPos As Integer
        Dim strTemp As String
        nPos = InStr(1, strLine, " ") - 1
        If nPos < 0 Then nPos = Len(strLine)
        strTemp = Left$(strLine, nPos)
        Select Case strTemp
        Case Is = "Private", "Public"
            strTemp = Trim(strLine)
            strTemp = Replace(strTemp, "Private ", "")
            strTemp = Replace(strTemp, "Public ", "")
            nPos = InStr(1, strTemp, " ") - 1
            If nPos < 0 Then nPos = Len(strTemp)
            strTemp = Left$(strTemp, nPos)
            If strTemp = "Function" Then
                closePair = "End Function"
            ElseIf strTemp = "Sub" Then
                closePair = "End Sub"
            Else
                '
            End If
        Case Is = "With"
            closePair = "End With"
        Case Is = "For"
            closePair = "Next"
        Case Is = "Do", "While"
            closePair = "Loop"
        Case Is = "Select"        ', "Case"
            closePair = "End Select"
        Case Is = "Sub"
            closePair = "End Sub"
        Case Is = "Function"
            closePair = "End Function"
        Case Is = "Property"
            closePair = "End Property"
        Case Is = "Enum"
            closePair = "End Enum"
        Case Is = "Type"
            closePair = "End Type"
        Case "If", "#If", "ElseIf", "#ElseIf", "Else", "Else:", "#Else",  _
        "#Else:"
            closePair = "End If"
        Case Else
            '
        End Select
    End Function

    Sub PageBreaksInPrinter()
        ThisWorkbook.Sheets("PrintPage").ResetAllPageBreaks
        Dim rng As Range
        Set rng = Nothing
        Dim cell As Range
        With ThisWorkbook.Sheets("PrintPage")
            For Each cell In .Range("B1:B" & .Range("B" & .rows.Count). _
            End(xlUp).Row)
                If Left(Trim(cell.Value), 3) = "---" Then
                    If rng Is Nothing Then
                        Set rng = cell
                    Else
                        Set rng = Union(rng, cell)
                    End If
                End If
            Next
            For Each cell In rng
                .HPageBreaks.Add Before:=.rows(cell.Row)
                .rows(cell.Row).PageBreak = xlPageBreakManual
            Next
        End With
    End Sub

    Sub FormatColourFormatters()
        Dim ws As Worksheet
        Set ws = ThisWorkbook.Sheets("PrintSettings")
        LBLcolourCode.ForeColor = ws.Range("J1").Value
        LBLcolourKey.ForeColor = ws.Range("J3").Value
        LBLcolourOdd.BackColor = ws.Range("J2").Value
        LBLcolourComment.ForeColor = ws.Range("J4").Value
    End Sub

    Sub ColorPaletteDialog(rng As Range, Lbl As MSForms.Label)
        If Application.Dialogs(xlDialogEditColor).Show(10, 0, 125, 125) =  _
        True Then
            'user pressed OK
            lcolor = ActiveWorkbook.Colors(10)
            rng.Value = lcolor
            rng.Offset(0, 1).Interior.Color = lcolor
            Lbl.ForeColor = lcolor
        End If
        ActiveWorkbook.ResetColors
    End Sub

    Sub RemoveBreaks()
        'remove line break loop
        Dim cell As Range
        Dim rng As Range
        With ThisWorkbook.Sheets("PrintPage")
            Set rng = .Range("B1:B" & .Range("B" & rows.Count).End(xlUp).Row)
        End With
        Dim coll As Collection
        Set coll = New Collection
        For Each cell In rng
            coll.Add CleanTrim(cell.Value)
        Next cell
        Dim arr As Variant
        arr = CollectionToArray(coll)
        rng.Value = WorksheetFunction.Transpose(arr)
    End Sub

    Function CleanTrim(ByVal s As String, Optional ConvertNonBreakingSpace As  _
    Boolean = True) As String
        'remove line break function
        Dim X As Long, CodesToClean As Variant
        CodesToClean = Array(0, 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, 127, 129, 141, 143, 144,  _
        157)
        If ConvertNonBreakingSpace Then s = Replace(s, Chr(160), " ")
        For X = LBound(CodesToClean) To UBound(CodesToClean)
            If InStr(s, Chr(CodesToClean(X))) Then
                s = Replace(s, Chr(CodesToClean(X)), vbNullString)
            End If
        Next
        CleanTrim = s
        '  CleanTrim = WorksheetFunction.Trim(S)
    End Function

    Sub GreenifyComments()
        Dim cell As Range
        Dim sh As Worksheet
        Set ws = ThisWorkbook.Sheets("PrintPage")
        Set rng = Nothing
        For Each cell In ws.Range("B1:B" & ws.Range("B" & rows.Count). _
        End(xlUp).Row)
```

```
            If Left(Trim(cell.Value), 1) = "'" Or Left(Trim(cell.Value), 3) =  _
            "Rem" Then
                cell.Font.Color = ThisWorkbook.Sheets("PrintSettings"). _
                Range("J4").Value
            End If
        Next
    End Sub

    Sub SpaceProcsInPrinter()
        'add empty line between end of sub/fun and start of next
        Dim cell As Range
        Dim rng As Range
        With ThisWorkbook.Sheets("PrintPage")
            Set rng = .Range("B2:B" & .Range("B" & rows.Count).End(xlUp).Row)
        End With
        With rng
            Set cell = .Find("*End Sub", LookIn:=xlValues)
            If Not cell Is Nothing Then
                firstAddress = cell.Address
                Do
                    With cell.Borders(xlEdgeBottom)
                        .LineStyle = xlContinuous
                        .Weight = xlThin
                        .Color = vbBlack
                    End With
                    Set cell = .FindNext(cell)
                Loop While Not cell Is Nothing And cell.Address <>  _
                firstAddress
            End If
        End With
        With rng
            Set cell = .Find("*End Function", LookIn:=xlValues)
            If Not cell Is Nothing Then
                firstAddress = cell.Address
                Do
                    With cell.Borders(xlEdgeBottom)
                        .LineStyle = xlContinuous
                        .Weight = xlThin
                    End With
                    Set cell = .FindNext(cell)
                Loop While Not cell Is Nothing And cell.Address <>  _
                firstAddress
            End If
        End With
    End Sub

    Sub NumberLinesPrinter()
        Dim lRow
        Dim cell As Range
        With ThisWorkbook.Sheets("PrintPage")
            lRow = .Range("B" & .rows.Count).End(xlUp).Row
            For Each cell In .Range("B1:B" & lRow)
```

```
                If cell.Row Mod 2 = 0 Then
                    Range(cell.Offset(0, -1), cell.Offset(0, 1)).Interior. _
                    Color = _
                    ThisWorkbook.Sheets("PrintSettings").Range("J2").Value
                End If
            Next cell
            '.Columns(1).HorizontalAlignment = xlLeft
        End With
    End Sub

    Sub BoldPrinterComponents()
        'format printer lines with component names
        Dim rng As Range
        Set rng = Nothing
        Dim cell As Range
        With ThisWorkbook.Sheets("PrintPage")
            For Each cell In .Range("B1:B" & .Range("B" & .rows.Count). _
            End(xlUp).Row)
                If Left(Trim(cell.Value), 3) = "---" Then
                    If rng Is Nothing Then
                        Set rng = cell
                    Else
                        Set rng = Union(rng, cell)
                    End If
                End If
            Next
        End With
        If rng Is Nothing Then Exit Sub
        rng.Font.size = 18
        rng.Font.Bold = True
        rng.Font.Color = vbBlack
    End Sub

    Sub copyLOGOPrinter()
        ThisWorkbook.Sheets("PrintSettings").Shapes("LOGO").Copy
        ThisWorkbook.Sheets("PrintPage").Paste ThisWorkbook. _
        Sheets("PrintPage").Range("B1")
        Dim shp As Shape
        Set shp = ThisWorkbook.Sheets("PrintPage").Shapes("LOGO")
        With ThisWorkbook.Sheets("PrintPage")
            shp.Left = .Range("B1").Left + ((.Range("B1").Width - shp.Width) / _
            2)
            shp.Top = .Range("B1").Top
            .rows(1).RowHeight = shp.Height + 50
            .Range("A2:C2").Interior.ColorIndex = 0
            With .Range("B1")
                .HorizontalAlignment = xlCenter
                .VerticalAlignment = xlVAlignBottom
                .Value = vbNewLine & vbNewLine & PrintFileName & vbNewLine &  _
                "www.github.com/alexofrhodes"
                .Characters.Font.size = 18
                .Characters.Font.Bold = True
                .Characters.Font.Underline = False
                .Characters.Font.ColorIndex = 10
                .Characters.Font.Name = "Comic Sans MS"
            End With
        End With
    End Sub

    Sub PrinterPageSetup()
        With ThisWorkbook.Sheets("PrintPage").PageSetup
            'narrow margins
            .LeftMargin = Application.InchesToPoints(0.25)
            .RightMargin = Application.InchesToPoints(0.25)
            .TopMargin = Application.InchesToPoints(0.25)
            .BottomMargin = Application.InchesToPoints(0.75)
            'left footer filename
            Dim FileName As String
            FileName = PrintFileName
            .LeftFooter = FileName
            '.LeftFooter =  "&F"    'Filename?
            'center footer page of pages
            .CenterFooter = "Page &P of &N"
            'right footer date
            .RightFooter = "&D"
            'fit all columns in one page width
            .FitToPagesWide = 1
            .FitToPagesTall = False
        End With
    End Sub

    Sub ShapesCompareLeft()
        'if code block connector lines spill to the next page,
        'we can easily follow the one we want if each line has it's own colour
        Dim rnd As Long
        Dim n As Variant
        Dim i As Long
        Dim s As Shape
        Dim sNames
        Set sNames = CreateObject("System.Collections.ArrayList")
        'rename lines to their .left position
        For Each s In ThisWorkbook.Sheets("PrintPage").Shapes
            If Not s.Name Like "LOGO" Then
                s.Name = s.Left
                'create a unique array of names
                If Not sNames.CONTAINS(s.Name) Then
                    sNames.Add s.Name
                End If
            End If
        Next s
        'assign unique colour to lines by level (left)
        For Each n In sNames
            rnd = RandomRGB
            For Each s In ThisWorkbook.Sheets("PrintPage").Shapes
```

```
                If Not s.Name Like "LOGO" Then
                    If s.Name = n Then
                        With s.line
                            .ForeColor.RGB = rnd
                            .Weight = 1.5
                        End With
                    End If
                End If
            Next s
        Next n
        Set sNames = Nothing
    End Sub

    Function RandomRGB()
        RandomRGB = RGB(Int(rnd() * 255), Int(rnd() * 255), Int(rnd() * 255))
    End Function

    Function StrWidth(s As String, sFontName As String, fFontSize As Double)  _
    As Double
        ' Returns the approximate width in points of a text string
        ' in a specified font name and font size
        ' Does not account for kerning
        Dim i As Long
        Dim j As Long
        If Len(sFontName) = 0 Then
            Exit Function
        End If
        If sFontName <> msFontName Then
            If Not InitChrWidths(sFontName) Then
                Exit Function
            End If
        End If
        For i = 1 To Len(s)
            j = Asc(Mid(s, i, 1))
            If j >= 32 Then
                StrWidth = StrWidth + fFontSize * mafChrWid(j)
            End If
        Next i
    End Function

    Function InitChrWidths(sFontName As String) As Boolean
        Dim i As Long
        Select Case sFontName
        Case "Consolas"
            For i = 32 To 127
                Select Case i
                Case 32 To 127
                    mafChrWid(i) = 0.5634
                End Select
            Next i
            '    Case "Arial"
            '        For i = 32 To 127
```

```
'            Select Case i
'            Case 39, 106, 108
'                mafChrWid(i) = 0.1902
'            Case 105, 116
'                mafChrWid(i) = 0.2526
'            Case 32, 33, 44,
'46, 47, 58, 59, 73, 91 To 93, 102, 124
'                mafChrWid(i) = 0.3144
'            Case 34, 40, 41, 45, 96, 114, 123, 125
'                mafChrWid(i) = 0.3768
'            Case 42, 94, 118, 120
'                mafChrWid(i) = 0.4392
'            Case 107, 115, 122
'                mafChrWid(i) = 0.501
'            Case 35, 36, 48 To 57, 63, 74, 76
', 84, 90, 95, 97 To 101, 103, 104, 110 To 113, 117, 121
'                mafChrWid(i) = 0.5634
'            Case 43, 60 To 62, 70, 126
'                mafChrWid(i) = 0.6252
'            Case 38, 65, 66, 69,
'72, 75, 78, 80, 82, 83, 85, 86, 88, 89, 119
'                mafChrWid(i) = 0.6876
'            Case 67, 68, 71, 79, 81
'                mafChrWid(i) = 0.7494
'            Case 77, 109, 127
'                mafChrWid(i) = 0.8118
'            Case 37
'                mafChrWid(i) = 0.936
'            Case 64, 87
'                mafChrWid(i) = 1.0602
'            End Select
'        Next i
'
'    Case "Calibri"
'        For i = 32 To 127
'            Select Case i
'            Case 32, 39, 44, 46, 73, 105, 106, 108
'                mafChrWid(i) = 0.2526
'            Case 40, 41, 45,
'58, 59, 74, 91, 93, 96, 102, 123, 125
'                mafChrWid(i) = 0.3144
'            Case 33, 114, 116
'                mafChrWid(i) = 0.3768
'            Case 34, 47, 76, 92, 99, 115, 120, 122
'                mafChrWid(i) = 0.4392
'            Case 35, 42, 43, 60 To 63, 69, 70, 83
', 84, 89, 90, 94, 95, 97, 101, 103, 107, 118, 121, 124, 126
'                mafChrWid(i) = 0.501
'            Case 36, 48 To 57, 66, 67, 7
'5, 80, 82, 88, 98, 100, 104, 110 To 113, 117, 127
'                mafChrWid(i) = 0.5634
'            Case 65, 68, 86
'                   mafChrWid(i) = 0.6252
'               Case 71, 72, 78, 79, 81, 85
'                   mafChrWid(i) = 0.6876
'               Case 37, 38, 119
'                   mafChrWid(i) = 0.7494
'               Case 109
'                   mafChrWid(i) = 0.8742
'               Case 64, 77, 87
'                   mafChrWid(i) = 0.936
'               End Select
'           Next i
'       Case "Tahoma"
'           For i = 32 To 127
'               Select Case i
'               Case 39, 105, 108
'                   mafChrWid(i) = 0.2526
'               Case 32, 44, 46, 102, 106
'                   mafChrWid(i) = 0.3144
'               Case 33, 45, 58, 59, 73, 114, 116
'                   mafChrWid(i) = 0.3768
'               Case 34, 40, 41, 47, 74, 91 To 93, 124
'                   mafChrWid(i) = 0.4392
'               Case 63, 76, 99, 107, 115, 118, 120 To 123, 125
'                   mafChrWid(i) = 0.501
'               Case 36, 42, 48 To 57, 70, 80
', 83, 95 To 98, 100, 101, 103, 104, 110 To 113, 117
'                   mafChrWid(i) = 0.5634
'               Case 66, 67, 69, 75, 84, 86, 88, 89, 90
'                   mafChrWid(i) = 0.6252
'               Case 38, 65, 71, 72, 78, 82, 85
'                   mafChrWid(i) = 0.6876
'               Case 35, 43, 60 To 62, 68, 79, 81, 94, 126
'                   mafChrWid(i) = 0.7494
'               Case 77, 119
'                   mafChrWid(i) = 0.8118
'               Case 109
'                   mafChrWid(i) = 0.8742
'               Case 64, 87
'                   mafChrWid(i) = 0.936
'               Case 37, 127
'                   mafChrWid(i) = 1.0602
'               End Select
'           Next i
'       Case "Lucida Console"
'           For i = 32 To 127
'               Select Case i
'               Case 32 To 127
'                   mafChrWid(i) = 0.6252
'               End Select
'           Next i
'
'       Case "Times New Roman"
```

```
        '        For i = 32 To 127
        '            Select Case i
        '            Case 39, 124
        '                mafChrWid(i) = 0.1902
        '            Case 32, 44, 46, 59
        '                mafChrWid(i) = 0.2526
        '            Case 33, 34, 47, 58, 73, 91 To 93, 105, 106, 108, 116
        '                mafChrWid(i) = 0.3144
        '            Case 40, 41, 45, 96, 102, 114
        '                mafChrWid(i) = 0.3768
        '            Case 63, 74, 97, 115, 118, 122
        '                mafChrWid(i) = 0.4392
        '            Case 94, 98 To 101, 103, 1
        '04, 107, 110, 112, 113, 117, 120, 121, 123, 125
        '                mafChrWid(i) = 0.501
        '            Case 35, 36, 42, 48 To 57, 70, 83, 84, 95, 111, 126
        '                mafChrWid(i) = 0.5634
        '            Case 43, 60 To 62, 69, 76, 80, 90
        '                mafChrWid(i) = 0.6252
        '            Case 65 To 67, 82, 86, 89, 119
        '                mafChrWid(i) = 0.6876
        '            Case 68, 71, 72, 75, 78, 79, 81, 85, 88
        '                mafChrWid(i) = 0.7494
        '            Case 38, 109, 127
        '                mafChrWid(i) = 0.8118
        '            Case 37
        '                mafChrWid(i) = 0.8742
        '            Case 64, 77
        '                mafChrWid(i) = 0.936
        '            Case 87
        '                mafChrWid(i) = 0.9984
        '            End Select
        '        Next i
    Case Else
        MsgBox "Font name """ & sFontName & """ not available!",  _
        vbCritical, "StrWidth"
        Exit Function
    End Select
    msFontName = sFontName
    InitChrWidths = True
End Function

Public Sub ChgTxtColor()
    With ThisWorkbook.Sheets("PrintPage").Cells.Font
        .Color = ThisWorkbook.Sheets("PrintSettings").Range("J1").Value
        .FontStyle = "Normal"
    End With
    Dim rng As Range
    Set rng = ThisWorkbook.Sheets("PrintPage").UsedRange
    Dim cell As Range
    Dim NumChars As Long
    Dim StartChar As Long
    Dim cellChar As Long
    Dim EndWords As Long
    Dim keywords As Range
    On Error Resume Next
    For Each cell In rng
        cellChar = Len(cell)
        For Each keywords In ThisWorkbook.Sheets("PrintSettings"). _
        Range("A1").CurrentRegion.Offset(1).Resize(, 1). _
        SpecialCells(xlCellTypeConstants)
            StartChar = InStrExact(1, cell.Text, keywords.Text)
            Do Until StartChar >= cellChar Or StartChar = 0
                NumChars = Len(keywords.Text)
                EndWords = StartChar + NumChars
                If Mid(cell.Text, StartChar - 1, 1) = " " Or StartChar =  _
                1 Then
                    If Mid(cell.Text, EndWords, 1) = " " Or EndWords >=  _
                    cellChar Then
                        With cell.Characters(Start:=StartChar,  _
                        Length:=NumChars).Font
                            'format matches
                            .FontStyle = "Bold"
                            .Color = ThisWorkbook.Sheets("PrintSettings"). _
                            Range("J3").Value
                        End With
                    End If
                End If
                StartChar = InStr(EndWords, cell.Text, keywords.Text)
            Loop
        Next
    Next
End Sub

Sub ResetPrinter(Optional keepText As Boolean = False)
    '    OptOn
    With ThisWorkbook.Sheets("PrintPage")
        .ResetAllPageBreaks
        If keepText = False Then
            .[A:C].Clear
        Else
            .[A:C].ClearFormats
            .Cells.Font.ColorIndex = vbBlack
            .Cells.Font.Bold = False
        End If
        .Columns("A:A").ColumnWidth = 3        '3
        .Columns("C:C").ColumnWidth = 1
        For Each s In ThisWorkbook.Sheets("PrintPage").Shapes
            'If Left(s.name, 2) <> "cp" Then
            s.Delete
            'End If
        Next
        .Cells.Font.Name = "Consolas"
        If .PageSetup.Orientation = xlPortrait Then
```

```vb
                .Columns("B:B").ColumnWidth = 90
            Else
                .Columns("B:B").ColumnWidth = 120
            End If
            .Cells.WrapText = False
            .Cells.UseStandardHeight = True
            '        .Cells.UseStandardWidth = True
        End With
        '    Application.ScreenUpdating = True
    End Sub

    Sub BreakText()
        'Coded by Anastasiou Alex
        'Version 1
        '20/1/2021
        '    Dim l As Long
        '    l = Timer()
        'to get things right, use a monospace font like Consolas
        Dim cell      As Range
        Dim TmpStr    As String
        Dim Splitter  As Integer
        Dim counter   As Integer
        Dim Limit     As Integer
        'how many characters fit your cell width (find manually)
        If ThisWorkbook.Sheets("PrintPage").PageSetup.Orientation =  _
        xlPortrait Then
            Limit = 75
        Else
            Limit = 100        '80
        End If
        'For which range to run
        Dim rng As Range
        With ThisWorkbook.Sheets("PrintPage")
            Set rng = .Range("B1:B" & .Range("B" & .rows.Count).End(xlUp).Row)
        End With
        Dim coll As Collection
        Set coll = New Collection
        On Error Resume Next
        For Each cell In rng
            TmpStr = cell.Text
            'remove unnecessary spaces (not trimming)
            If Right(cell.Offset(-1, 0), 1) = "_" Then
                counter = Len(cell.Offset(-1, 0)) - Len(Trim(cell.Offset(-1,  _
                0)))
                TmpStr = Application.WorksheetFunction.Rept(" ", counter) &  _
                Trim(cell.Text)
                cell.Value = TmpStr
            End If
            'create collection
            'if len of cell text <= limit then take as is
    REPEATME:
            If Len(TmpStr) > Limit Then
                counter = Len(TmpStr) - Len(Trim(TmpStr))
                'if comment
                'BreakText and add first part to collection. Repeat
                If Left(Trim(TmpStr), 1) = "'" Or Left(Trim(TmpStr), 3) =  _
                "Rem" Then
                    Splitter = Len(cell) / 2
                    coll.Add Left(TmpStr, Splitter)        '& " _"
                    TmpStr = Application.WorksheetFunction.Rept(" ", counter)  _
                    & _
                    "'" & Trim(Mid(TmpStr, Splitter + 1))
                    GoTo REPEATME
                    'if not comment
                Else
                    'find which symbol is closest to the limit and before it
                    Splitter = InStrRev(TmpStr, WhichFirst(TmpStr, ".`,`/`-`  _
                    `)", "`", Limit), Limit)
                    coll.Add Left(TmpStr, Splitter) & " _"
                    TmpStr = Application.WorksheetFunction.Rept(" ", counter)  _
                    & _
                    Trim(Mid(TmpStr, Splitter + 1))
                    GoTo REPEATME
                End If
            Else
                coll.Add (TmpStr)
            End If
        Next cell
        'replace sheet printer cells with broken text from collection
        Dim arr
        arr = CollectionToArray(coll)
        With ThisWorkbook.Sheets("PrintPage")
            .Cells.Clear
            .Range("B1:B" & UBound(arr) + 1).Value = WorksheetFunction. _
            Transpose(arr)
            .Cells.Font.Name = "Consolas"
        End With
        '    Debug.Print Timer() - 1
    End Sub

    Sub testWhichFirst()
        If ActiveCell = vbNullString Then
            Exit Sub
        End If
        WhichFirst ActiveCell, ".`,`/`-`_` `)", "`", Len(ActiveCell)
    End Sub

    Function WhichFirst(st As String, items As String, delim As String,  _
    AfterPosition As Integer)
        'Coded by Anastasiou Alex
        'Version 1
        '20/1/2021
        '
        'PARAMETERS
```

```vba
        'st : which string to parse
        'items : which characters are we looking for
        'delim : delimeter to split passed items
        'AfterPosition :
        Dim i As Long
        Dim varr As Variant
        varr = Split(items, delim)
    lp:
        On Error Resume Next
        'WhichFirst set to last varr item so it will be looped again?
        WhichFirst = varr(UBound(varr))
        For i = LBound(varr) To UBound(varr)
            'Debug.Print varr(i) & InStrRev(st, varr(i), AfterPosition)
            'find the item closest to the limit
            If InStrRev(st, varr(i), AfterPosition) > InStrRev(st, WhichFirst, _
            AfterPosition) Then
                WhichFirst = varr(i)
            End If
        Next i
        '    Debug.Print "Limit", AfterPosition & vbNewLine & _
        "Closest Item", WhichFirst & vbNewLine & _
        "Found At", InStrRev(st, WhichFirst, AfterPosition)
    End Function

    Function HasProject(wb As Workbook) As Boolean
        Dim WbProjComp As Object
        On Error Resume Next
        Set WbProjComp = wb.VBProject.VBComponents
        If Not WbProjComp Is Nothing Then HasProject = True
    End Function

    Sub OptOn()
        Application.ScreenUpdating = False
        Application.DisplayStatusBar = False
        Application.Calculation = xlCalculationManual
        Application.EnableEvents = False
        ' Note: this is a sheet-level setting.
        ActiveSheet.DisplayPageBreaks = False
    End Sub

    Sub OptOff()
        Application.ScreenUpdating = True
        Application.DisplayStatusBar = True
        Application.Calculation = xlCalculationAutomatic
        Application.EnableEvents = True
        ' Note: this is a sheet-level setting.
        ActiveSheet.DisplayPageBreaks = False
    End Sub

    Public Function ActiveProjName() As String
        'name of active project in vbeditor
        ActiveProjName = Mid(Application.VBE.ActiveVBProject.FileName,  _
        InStrRev(Application.VBE.ActiveVBProject.FileName, "\") + 1)
    End Function

    Sub PrintPDF()
        ThisWorkbook.Sheets("PrintPage").ExportAsFixedFormat _
        Type:=xlTypePDF, _
        FileName:=SaveLocation & Left(PrintFileName, InStr(1, PrintFileName,  _
        ".") - 1)
    End Sub


    Sub FoldersCreate(FolderPath As String)
        Dim individualFolders() As String
        Dim tempFolderPath As String
        Dim arrayElement As Variant
        individualFolders = Split(FolderPath, "\")
        For Each arrayElement In individualFolders
            tempFolderPath = tempFolderPath & arrayElement & "\"
            If FolderExists(tempFolderPath) = False Then
                MkDir tempFolderPath
            End If
        Next arrayElement
    End Sub

    Function ProtectedVBProject(ByVal wb As Workbook) As Boolean
        If wb.VBProject.Protection = 1 Then
            ProtectedVBProject = True
        Else
            ProtectedVBProject = False
        End If
    End Function

    Sub FollowLink(FolderPath As String)
        Dim oShell As Object
        Dim Wnd As Object
        Set oShell = CreateObject("Shell.Application")
        For Each Wnd In oShell.Windows
            If Wnd.Name = "File Explorer" Then
                If Wnd.Document.Folder.Self.Path = FolderPath Then Exit Sub
            End If
        Next Wnd
        Application.ThisWorkbook.FollowHyperlink Address:=FolderPath,  _
        NewWindow:=True
    End Sub

    Function FolderExists(ByVal strPath As String) As Boolean
        On Error Resume Next
        FolderExists = ((GetAttr(strPath) And vbDirectory) = vbDirectory)
        On Error GoTo 0
    End Function

    Function ComponentTypeToString(componentType As VBIDE.vbext_ComponentType) _
```

```
As String
    Select Case componentType
    Case vbext_ct_ActiveXDesigner
        ComponentTypeToString = "ActiveX Designer"
    Case vbext_ct_ClassModule
        ComponentTypeToString = "Class Module"
    Case vbext_ct_Document
        ComponentTypeToString = "Document Module"
    Case vbext_ct_MSForm
        ComponentTypeToString = "UserForm"
    Case vbext_ct_StdModule
        ComponentTypeToString = "Code Module"
    Case Else
        ComponentTypeToString = "Unknown Type: " & CStr(componentType)
    End Select
End Function

Private Sub goToFolder_MouseDown(ByVal Button As Integer, ByVal Shift As  _
Integer, ByVal X As Single, ByVal Y As Single)
FollowLink SaveLocation
End Sub

Private Sub CommandButton1_Click()
OptOff
PrintProject ActiveWorkbook
OptOn
FollowLink SaveLocation
End Sub

Function GetCompText(vbComp As VBComponent) As String
    Dim codeMod As CodeModule
    Set codeMod = vbComp.CodeModule
    If codeMod.CountOfLines = 0 Then GetCompText = "": Exit Function
    GetCompText = codeMod.Lines(1, codeMod.CountOfLines)
End Function


Function CollectionToArray(c As Collection) As Variant
    Dim A() As Variant: ReDim A(0 To c.Count - 1)
    Dim i As Long
    For i = 1 To c.Count
        A(i - 1) = c.Item(i)
    Next
    CollectionToArray = A
End Function



Function InStrExact(Start As Long, SourceText As String, WordToFind As  _
String, _
Optional CaseSensitive As Boolean = False, _
Optional AllowAccentedCharacters As Boolean = False) As Long
```

```
        Dim X As Long, Str1 As String, Str2 As String, Pattern As String
        Const UpperAccentsOnly As String = "HIP"
        Const UpperAndLowerAccents As String = "HIPηιρ"
        If CaseSensitive Then
            Str1 = SourceText
            Str2 = WordToFind
            Pattern = "[!A-Za-z0-9]"
            If AllowAccentedCharacters Then Pattern = Replace(Pattern, "!",  _
            "!" & UpperAndLowerAccents)
        Else
            Str1 = UCase(SourceText)
            Str2 = UCase(WordToFind)
            Pattern = "[!A-Z0-9]"
            If AllowAccentedCharacters Then Pattern = Replace(Pattern, "!",  _
            "!" & UpperAccentsOnly)
        End If
        For X = Start To Len(Str1) - Len(Str2) + 1
            If Mid(" " & Str1 & " ", X, Len(Str2) + 2) Like Pattern & Str2 &  _
            Pattern _
            And Not Mid(Str1, X) Like Str2 & "'[" & Mid(Pattern, 3) & "*" Then
            InStrExact = X
            Exit Function
        End If
    Next
    End Function

    Public Function IsBlockEnd(strLine As String) As Boolean
        Dim bOK As Boolean
        Dim nPos As Integer
        Dim strTemp As String
        nPos = InStr(1, strLine, " ") - 1
        If nPos < 0 Then nPos = Len(strLine)
        strTemp = Left$(strLine, nPos)
        Select Case strTemp
        Case "Next", "Loop", "Wend", "End Select", "Case", "Else", "#Else",  _
        "Else:", "#Else:", "ElseIf", "#ElseIf", "End If", "#End If"
            bOK = True
        Case "End"
            bOK = (Len(strLine) > 3)
        End Select
        IsBlockEnd = bOK
    End Function

    Public Function IsBlockStart(strLine As String) As Boolean
        Dim bOK As Boolean
        Dim nPos As Integer
        Dim strTemp As String
        nPos = InStr(1, strLine, " ") - 1
        If nPos < 0 Then nPos = Len(strLine)
        strTemp = Left$(strLine, nPos)
        Select Case strTemp
        Case "With", "For", "Do", "While", "Select", "Case", "Else", "Else:",  _
        "#Else", "#Else:", "Sub", "Function", "Property", "Enum", "Type"
            bOK = True
        Case "If", "#If", "ElseIf", "#ElseIf"
            bOK = (Len(strLine) = (InStr(1, strLine, " Then") + 4))
        Case "public", "Public", "Friend"
            nPos = InStr(1, strLine, " Static ")
            If nPos Then
                nPos = InStr(nPos + 7, strLine, " ")
            Else
                nPos = InStr(Len(strTemp) + 1, strLine, " ")
            End If
            Select Case Mid$(strLine, nPos + 1, InStr(nPos + 1, strLine, " ")  _
            - nPos - 1)
            Case "Sub", "Function", "Property", "Enum", "Type"
                bOK = True
            End Select
        End Select
        IsBlockStart = bOK
    End Function

    Private Sub UserForm_Initialize()
    PrintFileName = ActiveWorkbook.Name
    SaveLocation = Environ("USERprofile") & "\Documents\" &  _
    "vbArc\CodePrinter\"
    FoldersCreate SaveLocation
    FormatColourFormatters
    End Sub


    Private Sub cInfo_MouseDown(ByVal Button As Integer, ByVal Shift As  _
    Integer, ByVal X As Single, ByVal Y As Single)
    uDEV.Show
    End Sub


    Private Sub LBLcolourCode_Click()
    ColorPaletteDialog ThisWorkbook.Sheets("PrintSettings"). _
    Range("GeneralFontBackground"), LBLcolourCode
    End Sub

    Private Sub LBLcolourComment_Click()
    ColorPaletteDialog ThisWorkbook.Sheets("PrintSettings"). _
    Range("ColourComments"), LBLcolourComment
    End Sub

    Private Sub LBLcolourKey_Click()
    ColorPaletteDialog ThisWorkbook.Sheets("PrintSettings"). _
    Range("ColourKeywords"), LBLcolourKey
    End Sub

    Private Sub LBLcolourOdd_Click()
    ColorPaletteDialog ThisWorkbook.Sheets("PrintSettings").Range("OddLine"),  _
```

```
LBLcolourOdd
End Sub
```

--- uDEV ---

```
Private Sub LFaceBook_Click()
FollowLink ("https://www.facebook.com/VBA-Code-Archive-110295994460212")
End Sub

Private Sub LGitHub_Click()
FollowLink ("https://github.com/alexofrhodes")
End Sub

Private Sub LYouTube_Click()
FollowLink ("https://bit.ly/2QT4wFe")
End Sub

Private Sub LBuyMeACoffee_Click()
FollowLink ("http://paypal.me/alexofrhodes")
End Sub

Private Sub LEmail_Click()
If OutlookCheck = True Then
    MailDev
Else
    Dim out As String
    out = "anastasioualex@gmail.com"
    CLIP out
    MsgBox ("Outlook not found" & Chr(10) & _
    "DEV's email address" & vbNewLine & out & vbNewLine & "copied to  _
    clipboard")
End If
End Sub

Sub MailDev()
    'For Tips see: http://www.rondebruin.nl/win/winmail/Outlook/tips.htm
    'Working in Office 2000-2016
    Dim OutApp As Object
    Dim OutMail As Object
    Dim strBody As String
    Set OutApp = CreateObject("Outlook.Application")
    Set OutMail = OutApp.CreateItem(0)
    '    strbody = "Hi there" & vbNewLine & vbNewLine & _
    "This is line 1" & vbNewLine & _
    "This is line 2" & vbNewLine & _
    "This is line 3" & vbNewLine & _
    "This is line 4"
    On Error Resume Next
    With OutMail
        .To = "anastasioualex@gmail.com"
        .CC = vbNullString
        .BCC = vbNullString
        .Subject = "DEV REQUEST OR FEEDBACK FOR -CODE ARCHIVE-"
        .body = strBody
        'You can add a file like this
        '.Attachments.Add ("C:\test.txt")
        '.Send
        .Display
    End With
    On Error GoTo 0
    Set OutMail = Nothing
    Set OutApp = Nothing
End Sub
```