



1. 메일 보내기 명령문 보조 모듈 추가

이전 강의에서 제작한 PDF 변환 명령문 및 그 외 보조 모듈을 추가합니다. 각 명령문에 대한 자세한 설명은 이전 강의 [PDF 변환 매크로 만들기 \(퀵 VBA 6 강\) 강의](#)를 참고하세요.

A. 파일명 설정 및 파일 저장을 위한 보조 명령문

```
'#####  
'z_SubModule 이라는 모듈을 생성한 뒤 복사/붙여넣기 하세요  
'#####  
Option Explicit  
  
Public Function FileExists(ByVal path_ As String) As Boolean  
  
    FileExists = (Dir(path_, vbDirectory) <> "")  
  
End Function  
  
Public Function GetDesktopPath(Optional BackSlash As Boolean = True)  
  
Dim oWShell As Object  
  
Set oWShell = CreateObject("WScript.Shell")  
  
If BackSlash = True Then  
    GetDesktopPath = oWShell.SpecialFolders("Desktop") & "\"  
Else  
    GetDesktopPath = oWShell.SpecialFolders("Desktop")  
End If  
  
Set oWShell = Nothing  
  
End Function  
  
Function FileSequence(filePath As String, Optional Sequence As Long = 1)  
  
Dim Ext As String: Dim Path As String: Dim newPath As String  
Dim Pnt As Long  
  
Pnt = InStrRev(filePath, ".")  
Path = Left(filePath, Pnt - 1)  
Ext = Right(filePath, Len(filePath) - Pnt + 1)
```



```

newPath = Path & Sequence & Ext

Do Until FileExists(newPath) = False
    Sequence = Sequence + 1
    newPath = Path & Sequence & Ext
Loop

FileSequence = newPath

End Function

Function ValidFileName(ByVal FileName As String) As Boolean

Dim Arr As Variant: Dim Val As Variant
Dim Pnt As Long

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

If InStr(1, FileName, ":\") > 0 Then
    Pnt = InStrRev(FileName, "\")
    FileName = Right(FileName, Len(FileName) - Pnt)
    Debug.Print FileName
End If

ValidFileName = True

For Each Val In Arr
    If InStr(1, FileName, Val) > 0 Then ValidFileName = False: Exit Function
Next

End Function
    
```

B. 페이지 레이아웃 설정을 위한 보조 모듈 (Page_Setup)

```

'#####
'z_PageSetup 이라는 모듈을 생성한 뒤 복사/붙여넣기 하세요
'#####

Public Enum ePrintMargin
xlNone = 0
xlNarrow = 1
xlNormal = 2
    
```



```
xlWide = 3
End Enum

Public Enum ePaperSize
xlA4 = 9
xlA3 = 8
xlLetter = 1
xlA5 = 11
End Enum

Function getPrintMargin(eValue As ePrintMargin) As Variant

'// 설정된 eNum 값으로 페이지 여백설정을 위한 값을 배열로 나열합니다.

Select Case eValue
Case 0
getPrintMargin = Array(0.05, 0.05, 0.05, 0.05, 0.1, 0.1)
Case 1
getPrintMargin = Array(0.25, 0.25, 0.75, 0.75, 0.3, 0.3)
Case 2
getPrintMargin = Array(0.7, 0.7, 0.75, 0.75, 0.3, 0.3)
Case 3
getPrintMargin = Array(1, 1, 1, 1, 0.5, 0.5)
End Select

End Function

Sub Page_setup(WS As Worksheet, Optional LHead As String = "", Optional RHead
As String = "&D / &T", _
Optional LFoot As String = "본 페이지의 무단복제를 금합니다.", Optional RFoot As
String = "&P / &N 페이지", _
Optional eMargin As ePrintMargin = xlNarrow, _
Optional HFit As Boolean = True, Optional VFit As Boolean = False, _
Optional HCenter As Boolean = True, Optional VCenter As Boolean = False, _
Optional eOrient As XlPageOrientation = xlPortrait, Optional eSize As
ePaperSize = xlA4)

Dim pSetup As String
Dim varMargin As Variant
Dim lngOrient As Integer

'// 인쇄설정 업데이트 중단 (속도증가)
Application.PrintCommunication = False

'// 인쇄여백값을 받아옵니다.
```



엑셀은?

퀵 VBA 7강

- 엑셀 메일 보내기 자동화 매크로 (아웃룩 연동)



오빠두

www.OPPADU.com

오빠두엑셀 무료 E-Book

```

varMargin = getPrintMargin(eMargin)

'// 인쇄용지 방향을 설정합니다.
If eOrient = xlPortrait Then
lngOrient = 1
Else
lngOrient = 2
End If

'// ExecuteExcel4Macro 의 Page.Setup 명령문 실행을 위한 문구를 입력합니다.
Head = ""&"&L" & LHead & "&R" & RHead & """" ' // 페이지 머릿말입니다.
Foot = ""&"&L" & LFoot & "&R" & RFoot & """" ' // 페이지 꼬릿말입니다.
pLeft = varMargin(0) ' // 왼쪽여백
pRight = varMargin(1) ' // 오른쪽여백
Top = varMargin(2) ' // 윗여백
Bot = varMargin(3) ' // 아래여백
Head_margin = varMargin(4) ' // 머릿말여백
Foot_margin = varMargin(5) ' // 꼬릿말여백
Hdng = 0 ' // 행/열반복 출력여부 0 =
반복출력안함 1 = 반복출력

Grid = False ' // 눈금선출력여부
Notes = False ' // 메모출력여부
H_cntr = HCenter ' // 가운데정렬
V_cntr = VCenter ' // 중앙정렬
Orient = lngOrient ' // 문서방향, 1 = 세로 2 = 가로
Paper_size = eSize ' // 용지크기
Pg_num = 1 ' // 페이지 시작번호
Pg_order = 1 ' // 페이지번호 순서, 1 = 위-아래-우
2 = 좌-우-아래
Quality = "" ' // 인쇄품질 (dot-per-inch 로
입력) (공백 = 자동)
bw_cells = False ' // 흑백인쇄여부, TRUE =
글자/테두리 검정, 배경 흰색 FALSE = 색깔
pScale = 100 ' // 축소/확대비율 또는 TRUE (Fit
to Page)

'// 여백을 없으므로 설정할 경우 머릿말/꼬릿말을 삭제하여 인쇄영역과 겹치지 않도록 합니다.
If eMargin = xlNone Then
Head = ""
Foot = ""
End If

```



```

'// ExecuteExcel4Macro 명령문을 실행합니다.
pSetup = "PAGE.SETUP(" & Head & ", " & Foot & ", " & pLeft & ", " & pRight & ",
" & Top & ", " & Bot & ", "
pSetup = pSetup & Hdng & ", " & Grid & ", " & H_cntr & ", " & V_cntr & ", " &
Orient & ", "
pSetup = pSetup & Paper_size & ", " & pScale & ", "
pSetup = pSetup & Pg_num & ", " & Pg_order & ", " & bw_cells & ", " & Quality &
", "
pSetup = pSetup & Head_margin & ", " & Foot_margin & ", " & Notes & ") "

Application.ExecuteExcel4Macro pSetup

'// ExecuteExcel4Macro에서는 '한 페이지에 행/열 맞추기' 기능이 지원되지 않습니다.
'// 따라서 시트의 PageSetup 속성으로 '페이지 행/열 맞추기 기능을 설정합니다.
With WS.PageSetup
If HFit = True Then
.FitToPagesWide = 1
Else
.FitToPagesWide = False
End If

If VFit = True Then
.FitToPagesTall = 1
Else
.FitToPagesTall = False
End If
End With

'// 인쇄설정 업데이트
Application.PrintCommunication = True

End Sub

```

C. 선택범위 PDF 저장 모듈 (Rng_To_PDF)

```

'#####
'모듈을 추가한 뒤 복사/붙여넣기 하세요.
'#####

Sub Rng_To_Pdf(rngSelect As Range, _
Optional FileName As String = "pdf 출력", _
Optional SavePath As String = "", _

```



```

Optional DocProperty As Boolean = True, _
Optional PrintArea As Boolean = False, _
Optional OpenPdf As Boolean = False, _
Optional AddSequence As Boolean = True)

Dim WS As Worksheet
Dim FilePath As String

Set WS = rngSelect.Parent

If SavePath = "" Then SavePath = GetDesktopPath

FilePath = SavePath & FileName & ".pdf"

If ValidFileName(FilePath) = False Then MsgBox ("올바른 파일명을 사용하세요"): Exit Sub

If AddSequence = True Then
    FilePath = FileSequence(FilePath, 1)
End If

rngSelect.ExportAsFixedFormat xlTypePDF, FilePath, xlQualityStandard,
DocProperty, PrintArea, , , OpenPdf

End Sub

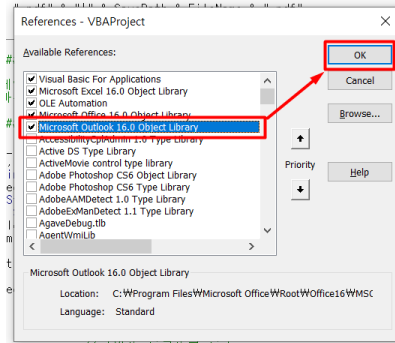
```

2. 메일 보내기 본 명령문 작성 (Send_Email 명령문)

이번 강의에서는 엑셀을 아웃룩을 연동하여 이메일을 보내는 방법을 알아봅니다. 새로운 모듈을 추가한 뒤, 모듈의 이름을 "Email_Module"로 변경합니다.

A. Outlook 라이브러리 추가

VBA 편집창 상단 '도구' - '참조' 에서 Outlook 라이브러리(Microsoft Outlook 16.0 Object Library) 를 추가합니다.



아웃룩을 참조하기 위해 Outlook 라이브러리를 추가합니다.

B. 명령문 인수 작성

명령문에 들어갈 인수를 작성합니다. Send_Email 명령문에는 총 8 개의 인수가 들어갑니다.

인수명	데이터구분	설명	기본값
MailTo	String	받는 사람의 메일 주소입니다.	
Subject	String	메일 제목입니다.	
HTMLString	String	HTML 형식의 메일 본문입니다.	
PasteSelection	Boolean	[선택인수] 선택 범위를 메일 본문에 붙여넣기 할지 여부입니다.	False
CCTo	String	[선택인수] 참조 메일 주소입니다.	""
BCCTo	String	[선택인수] 숨은참조 메일 주소입니다.	""
AttachFilePath	String	첨부파일의 파일 경로입니다.	""
PathDelimiter	String	여러 개의 첨부파일이 있을 시, 파일 경로를 구분할 구분자입니다.	" "



C. 명령문 변수 작성 및 설정

명령문에 들어갈 변수를 작성한 뒤 각 변수를 설정합니다.

- ✓ 변수 중 wdPasteDefault 는 워드에디터에서 복사한 값을 붙여넣기 할 방식을 결정하는 WdRecoveryType 의 한 항목이며, 원칙적으로 wdRecoveryType 항목은 Word Object Library 를 참고하여 지정하는 것이 맞습니다.
- ✓ 하지만 아웃룩에서 사용하는 워드에디터는 기본 워드 프로그램으로 참조하는 Library 와 별개로 작성되므로, Word Object Library 를 사용할 수 없습니다. 따라서 **Variant(임의값)** 으로 변수를 지정하여 명령문을 작성해주세요.

변수명	데이터구분	설명
AppOutlook	Outlook.Application	실행할 아웃룩 프로그램입니다.
newEmail	Outlook.MailItem	아웃룩에서 작성할 메일입니다.
pageInspector	Outlook.Inspector	아웃룩 워드에디터를 불러오기 위한 Inspector 항목입니다.
pageEditor	Object	아웃룩에서 메일을 편집할 워드 에디터입니다.
varFilePath	Variant	String 형태로 받아온 첨부파일 주소를 배열형태로 변환합니다.
FileCount	Long	첨부파일의 개수입니다.
i	Long	For 반복문의 연번입니다.
wdPasteDefault	Variant	엑셀에서 선택한 범위를 메일에 붙여넣기 할 방식입니다.



```

Dim AppOutlook As Outlook.Application
Dim newEmail As Outlook.MailItem
Dim pageInspector As Outlook.Inspector
Dim pageEditor As Object
Dim varFilePath As Variant
Dim FileCount As Long
Dim i As Long
Dim wdPasteDefault As Variant

Set AppOutlook = New Outlook.Application
Set newEmail = AppOutlook.CreateItem(olMailItem)

```

D. 메일항목 설정 #1

보낼 메일의 각 항목을 설정합니다. 설정에 사용되는 모든 항목은 명령문의 인수/변수에서 받아왔으므로, 각 항목을 호출해주면 됩니다.

- To : 받는 사람의 메일 주소입니다.
- CC : 참조 메일 주소입니다.
- BCC : 숨은 참조 메일 주소입니다.
- Subject : 메일 제목입니다.
- Attachments.Add : 첨부파일을 추가합니다.
- HTMLBody : HTML 형식의 메일 본문을 작성합니다.
- DefferedDeliveryTime : 예약발송 또는 지연발송을 위한 시간을 설정합니다.
 - ✓ DateSerial 함수 - 연/월/일로 입력한 날짜에 해당하는 일련번호를 반환합니다.
 - ✓ TimeSerial 함수 - 시간/분/초로 입력한 시간에 해당하는 일련번호를 반환합니다.
 - ✓ DateAdd 함수 - 지정한 시간/날짜로부터 선택한 '단위' 만큼 지연된 값을 반환합니다.

예제 1) 2019년 12월 25일 오후 9시에 발송 : DateSerial(2019, 12, 25) + TimeSerial(9,0,0)

예제 2) 현재시간으로부터 30분 뒤에 발송 : DateAdd("n", 30, Now)



E. 메일항목 설정 #2

엑셀의 선택한 범위를 메일 본문에 붙여넣기 할 경우, 아웃룩의 워드에디터 추가 설정이 필요합니다. 본 강의에서는 엑셀의 선택한 범위를 메일 본문 마지막 부분에 기본 표 형식으로 붙여넣기 하는 방법을 알아봅니다.

- pageEditor.Application.Selection.Start : 엑셀에서 복사한 범위를 붙여넣기 할 시작지점입니다.
- pageEditor.Application.Selection.PasteAndFormat : 붙여넣기 할 방식입니다. 본 강의에서는 붙여넣기 기본값을 사용하였습니다.

✓ 워드에디터 붙여넣기 형식 전체목록 보러가기 :

<https://docs.microsoft.com/en-us/office/vba/api/word.wdrecoverytype>

```
If PasteSelection = True Then
    .Display
    Set pageInspector = newEmail.GetInspector
    Set pageEditor = pageInspector.WordEditor
    pageEditor.Application.Selection.Start = Len(.Body)

    Selection.Copy
```



```
pageEditor.Application.Selection.PasteAndFormat wdPasteDefault
Else
    .Display
End If
```

F. 메일 보내기 및 변수 초기화

Send 함수를 통해 메일을 발송합니다. 여러 개의 메일을 발송할 경우, 할당된 변수를 초기화하여 매크로 동작속도를 더욱 빠르게 합니다.

변수 초기화는 마지막으로 할당된 변수부터 역순으로 하는 것에 주의하세요.

```
' .Send      '// 메일을 보내려면 주석처리를 해제하세요.

Set pageEditor = Nothing
Set pageInspector = Nothing
Set newEmail = Nothing
Set AppOutlook = Nothing
```

G. Send_Email 전체 명령문

```
'#####
' 명령문      : Send_Email
' 설명       : 아웃룩과 연동하여 메일보내기를 자동화하는 모듈입니다.
'#####

Sub Send_Email (MailTo As String, _
                Subject As String, _
                HTMLString As String, _
                Optional PasteSelection As Boolean = False, _
                Optional CCTo As String = "", _
                Optional BCCTo As String = "", _
                Optional AttachFilePath As String = "", _
                Optional PathDelimiter As String = "|")

Dim AppOutlook As Outlook.Application      '// 아웃룩 프로그램
Dim newEmail As Outlook.MailItem          '// 아웃룩 새로 메일을 보내기 위해 생성한
메일
Dim pageInspector As Outlook.Inspector    '// 아웃룩 워드에디터 가져오기위한 항목
Dim pageEditor As Object                  '// 아웃룩 이메일 편집창
Dim varFilePath As Variant                '// 파일경로를 배열형태로 만들어준 변수
```



```

Dim FileCount As Long           '// 첨부파일의 개수
Dim i As Long                   '// For 문 반복문의 변수
Dim wdPasteDefault As Variant

Set AppOutlook = New Outlook.Application
Set newEmail = AppOutlook.CreateItem(olMailItem)

If AttachFilePath <> "" Then
    varFilePath = Split(AttachFilePath, PathDelimiter)
End If

With newEmail
    .To = MailTo
    .CC = CCTo
    .BCC = BCCTo
    .Subject = Subject
    If AttachFilePath <> "" Then
        For i = 1 To UBound(varFilePath) + 1
            .Attachments.Add varFilePath(i - 1), 1, i
        Next
    End If
    .HTMLBody = HTMLString

    '.DeferredDeliveryTime = DateAdd("n", 5, Now)
    .DeferredDeliveryTime = DateSerial(2030, 1, 1) + TimeSerial(8, 0, 0)

    If PasteSelection = True Then
        .Display
        Set pageInspector = newEmail.GetInspector
        Set pageEditor = pageInspector.WordEditor
        pageEditor.Application.Selection.Start = Len(.Body)

        Selection.Copy
        pageEditor.Application.Selection.PasteAndFormat wdPasteDefault
    Else
        .Display
    End If

    '.Send           '// 메일을 보내려면 주석처리를 해제하세요.

End With

Set pageEditor = Nothing
Set pageInspector = Nothing
    
```



```
Set newEmail = Nothing
Set AppOutlook = Nothing

End Sub
```

3. 이메일 보내기 명령문 테스트

작성한 메일 보내기 명령문을 테스트하기 위한 명령문을 작성합니다. 아래 명령문을 복사/붙여넣기 한 뒤 명령문을 실행해주세요. 테스트 명령문은 아래 동작을 수행합니다.

1. 직원명_부서명_년도-월 의 파일명으로 PDF 파일을 바탕화면에 저장
2. 저장된 PDF 파일을 메일에 첨부 (파일경로를 세로바(|)로 구분, 2 개 동시첨부 가능)
3. HTML 형식으로 작성된 메일 본문 및 급여명세서의 B2:E18 범위를 메일 본문에 붙여넣기
4. test@oppadu.com 로 메일 발송

테스트 명령문에서 받는 사람 및 메일 본문, PDF 변환 범위 등은 필요에 따라 변경 후 사용 가능합니다.

```
#####
' 오피아두엑셀 퀵 vba 7 강, 메일 보내기 자동화 완성파일
' 명령문에 대한 자세한 설명은 아래 링크에서 확인하세요.
' https://www.oppadu.com/엑셀-메일-보내기-아웃룩-매크로/
#####

Sub Test ()

Dim FileName As String
Dim SavePath As String

FileName = Sheet4.Range("E3").Value & "_" & Sheet4.Range("C4").Value & "_" &
Sheet4.Range("H3").Value & "년 " & Sheet4.Range("H4").Value & "월"
SavePath = GetDesktopPath

Rng_To_Pdf Sheet4.Range("B2:E18"), FileName, SavePath, OpenPdf:=False,
AddSequence:=False
'// Rng_To_Pdf 명령문의 첫번째 인수를 Selection 으로 변경하면 선택한 범위를 PDF 파일로
추출하여 메일에 첨부합니다.

Sheet4.Range("B2:E18").Select
```



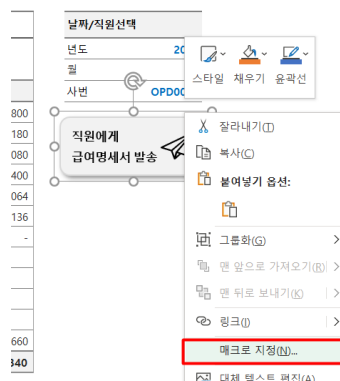
'// 선택된 범위가 아닌 원하는 부분을 지정해서 메일에 첨부하는 방법이 궁금하신분은 Send_Email 명령문 관련 포스트를 참고하세요.

```
Send_Email "test@oppadu.com", _
    FileName, _
    "<p><span style=""font-family: NanumGothic, 나눔고딕, sans-serif; font-size: 9pt;"><b><u><span style=""font-size: 9pt;">오빠두
대리</span></u></b> 님께&nbsp;</span></p><p><br></p><p><span style=""font-
family: NanumGothic, 나눔고딕, sans-serif; font-size: 9pt;">귀하의 <b><u><span
style=""font-size: 9pt;">2019년 10월 급여명세서</span></u></b>를
송부드립니다.&nbsp;</span></p><p><span style=""font-family: NanumGothic, 나눔고딕,
sans-serif; font-size: 9pt;">오빠두엑셀을 위한 귀하의 노고에 깊은 감사드리며 더욱
발전된 모습으로 귀하에 노고에 보답하겠습니다.</span></p>", _
    True, _
    "", _
    SavePath & FileName & ".pdf" & "|" & SavePath & FileName & ".pdf"

End Sub
```

4. 시트에 매크로 실행 버튼 추가

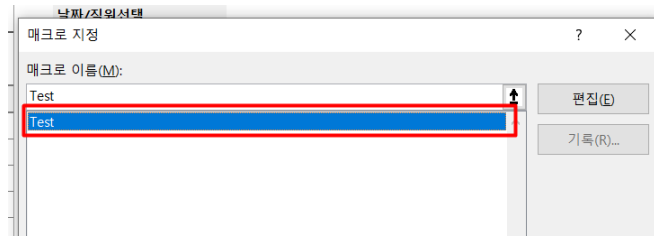
작성한 매크로를 쉽게 실행하기 위한 버튼을 시트위에 추가합니다. 시트에 도형을 생성한 뒤, 원하는 문구/아이콘을 추가합니다. 생성한 도형을 우클릭하여 매크로를 연결합니다.



도형을 우클릭 한 뒤, [매크로지정]을 선택합니다.

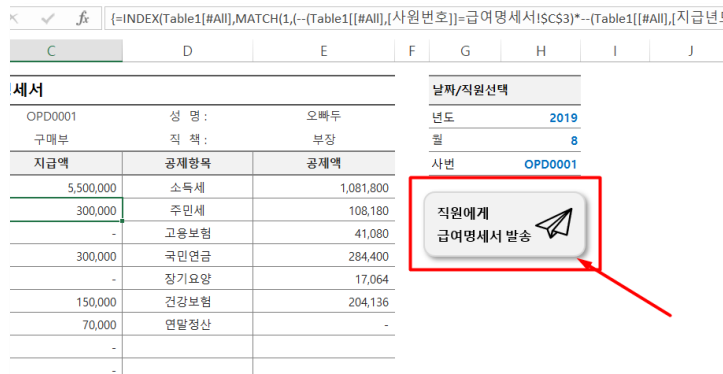


매크로 지정을 선택하면 새로운 팝업창이 나타납니다. 팝업창에서 방금 추가 한 'Test' 명령문을 선택합니다.



팝업창에서 Test 명령문을 선택한 뒤, 확인을 눌러 마무리합니다.

이후 도형을 클릭하면 메일 보내기 명령문이 실행됩니다.



도형을 클릭하면 메일 보내기 명령문이 실행됩니다.