



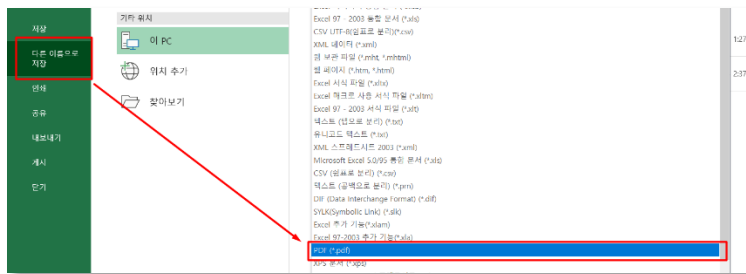
1. 엑셀을 PDF 로 저장하는 방법 (매크로 없이)

엑셀은 PDF 저장기능을 기본적으로 제공하며, 크게 2 가지 방법이 있습니다.

- PDF 파일 형식으로 [다른 이름으로 저장]하기
- PDF 형식이 지원되는 프린터로 출력하기 (Microsoft Print to PDF, ALPDF, 등)

1-1. PDF 파일 형식으로 '다른 이름으로 저장'

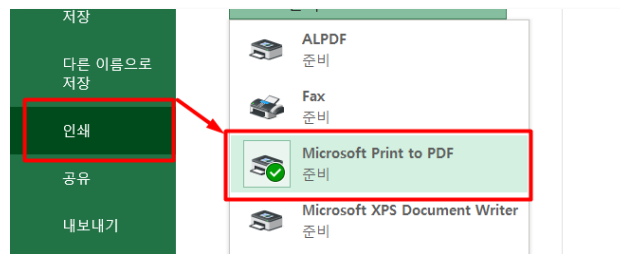
[파일] - [다른 이름으로 저장]을 선택합니다. 이후 파일형식을 'PDF' 로 선택한 뒤 원하는 경로에 파일을 저장합니다.



[다른 이름으로 저장] 에서 엑셀을 PDF 형식으로 저장 할 수 있습니다.

1-2. PDF 출력이 지원되는 프린터로 저장하기 (Microsoft Print to PDF 이용)

Microsoft 에서 지원하는 Microsoft Print to PDF 프린터를 사용하여 엑셀을 PDF 로 저장할 수 있습니다. Microsoft Print to PDF 프린터는 윈도우 7 이상 사용자라면 기본으로 설치되어 있으며, 만약 프린터목록에서 보이지 않을 경우 관련링크에 자세히 적어드렸으니 프린터 설치 후 진행하시기 바랍니다.



PDF 출력이 지원되는 프린터를 사용하면 엑셀을 PDF 로 보다 쉽게 변환할 수 있습니다.

프린터로 PDF 를 출력할 경우 머리말과 꼬릿말, 그리고 한페이지에 열 맞추기 등 다양한 페이지 레이아웃을 설정할 수 있는 장점이 있습니다.



엑셀은?

오빠두

퀵 VBA 6강 - 엑셀 선택범위 PDF 변환 매크로

www.oppadu.com

오빠두엑셀 무료 E-Book

2. 이번 강의 사용된 보조 명령문

이번 강의에서는 총 4 개의 보조 명령문과 1 개의 보조 모듈이 사용되었습니다. 각 보조 명령문에 대한 자세한 설명은 관련 포스트를 참고하세요.

➤ 사용된 VBA 보조 명령문

FileExists 함수 :: 사용자지정함수

설명 : **지정한 파일경로의 파일존재여부를 확인합니다.**

구문 : = FILEEXISTS (파일경로)

[오빠두 함수 상세설명 링크](#)

GetDesktopPath 함수 :: 사용자지정함수

설명 : **사용자 컴퓨터의 바탕화면경로를 출력합니다.**

구문 : = GETDESKTOPPATH ([백슬래쉬표시])

[오빠두 함수 상세설명 링크](#)

FileSequence 함수 :: 사용자지정함수

설명 : **파일이름이 중복되지 않도록 순번을 매깁니다.**

구문 : = FILESEQUENCE(파일경로, [시작번호])

[오빠두 함수 상세설명 링크](#)

ValidFileName 함수 :: 사용자지정함수

설명 : **지정한 파일이름의 사용가능 여부를 확인합니다.**

구문 : = VALIDFILENAME (파일경로)

[오빠두 함수 상세설명 링크](#)

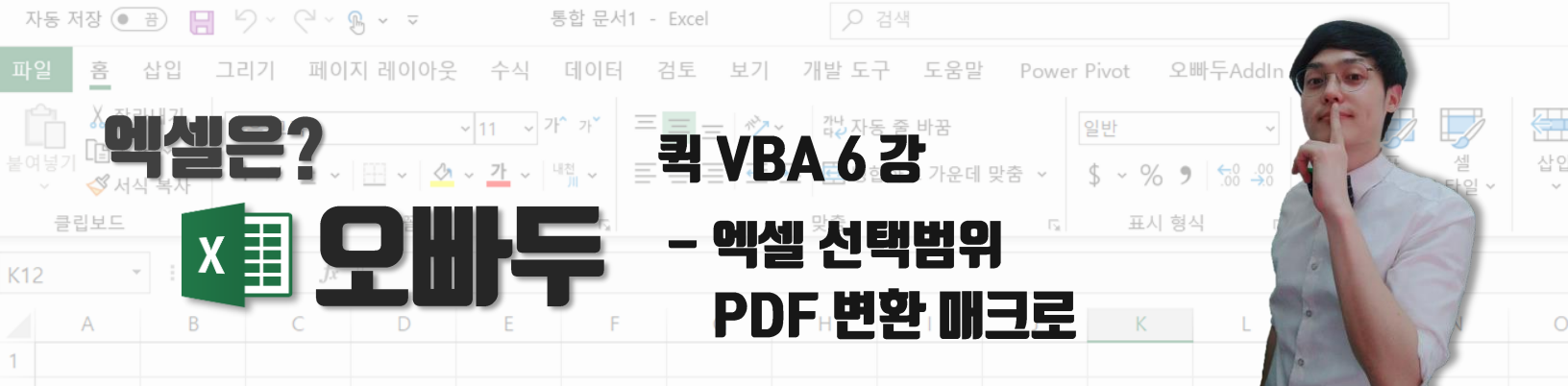
➤ 사용된 VBA 보조 모듈

PAGE_SETUP 모듈 :: VBA 모듈

설명 : **시트의 인쇄 레이아웃을 설정합니다.**

구문 : PAGE_SETUP (머리말, 꼬릿말, 여백, 출력여부, 정렬여부, 용지크기, 등)

[오빠두 엑셀 상세설명 링크](#)



www.oppadu.com

오빠두엑셀 무료 E-Book

✓ 사용된 VBA 보조 명령문 전체 코드

```
'#####
'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

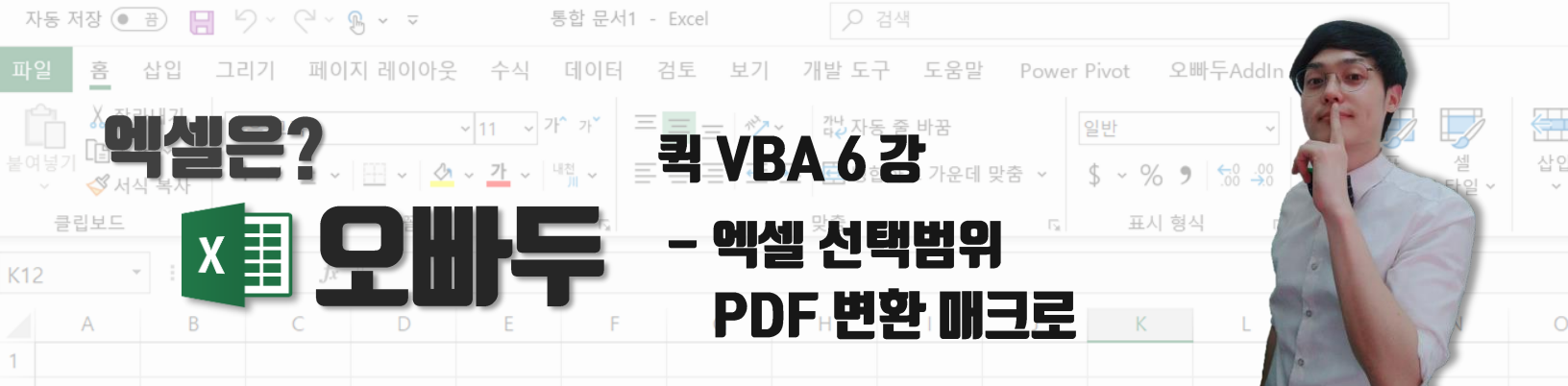
Function FileSequence(FilePath As String, Optional Sequence As Long = 1)

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

```

✓ 사용된 VBA 보조 모듈 전체코드 (PAGE_SETUP)

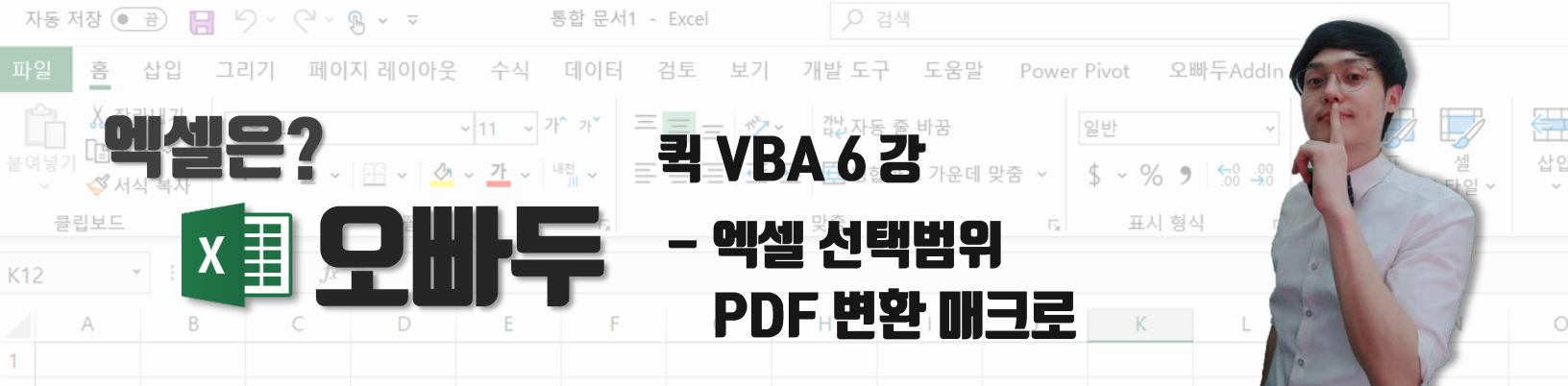
```

'#####
'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.
varMargin = getPrintMargin(eMargin)

'// 인쇄용지 방향을 설정합니다.
```



엑셀은?



오빠두

퀵 VBA 6강

- 엑셀 선택범위
PDF 변환 매크로

www.oppadu.com

오빠두엑셀 무료 E-Book

```

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 없음으로 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 & ")"

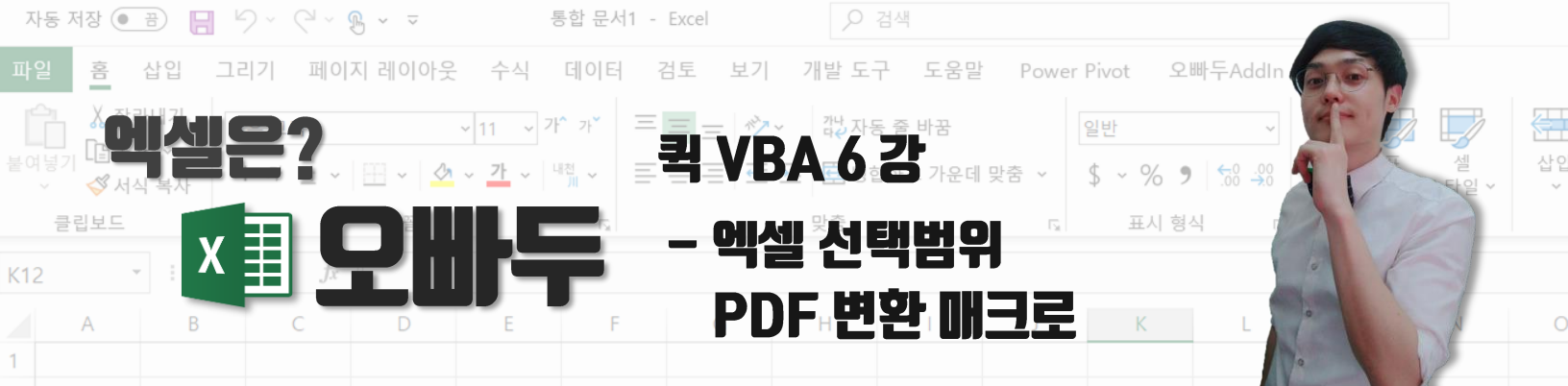
Application.ExecuteExcel4Macro pSetup

'// ExecuteExcel4Macro 에서는 '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
```



3. 엑셀 PDF 저장 매크로 본 명령문 작성

엑셀을 PDF 로 저장하기 위한 본 명령문을 작성합니다. 명령문의 이름은 Rng_To_Pdf 로 진행하겠습니다. VBA 편집창의 [삽입] - [모듈] 을 클릭하여 새로운 모듈을 추가한 뒤 명령문을 작성합니다.

3-1. 인수 설정

본 명령문에는 총 7 개의 인수가 들어갑니다. 각 인수는 아래와 같습니다.

인수	형식	설명	기본값
rngSelect	Range	PDF 로 출력할 범위입니다.	
FileName	String	(Optional) 저장할 파일명입니다.	"pdf 출력"
SavePath	String	(Optional) 저장 경로입니다.	"
DocProperty	Boolean	(Optional) 문서속성 저장 여부입니다.	True
PrintArea	Boolean	(Optional) 기존 인쇄영역 무시 여부입니다.	False
OpenPdf	Boolean	(Optional) 출력 후 PDF 파일 실행 여부입니다.	False
AddSequence	Boolean	(Optional) 파일이름 중복 시 순번으로 저장 여부입니다.	True

```

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)

End Sub
    
```




3-2. 변수 설정

대부분의 변수는 명령문의 인수로 이미 받아왔으므로, 2 개의 변수만 추가로 생성합니다. 생성할 변수는 아래와 같습니다.

변수명	형식	설명
WS	Worksheet	PDF 로 출력할 범위의 대상 워크시트입니다.
FilePath	String	명령문의 인수로 받아온 저장경로와 파일명이 합쳐진 완성된 파일 저장경로입니다.

```
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
```

WS 는 명령문에서 받아온 rngSelect 범위의 상위개체인 워크시트로 설정합니다.

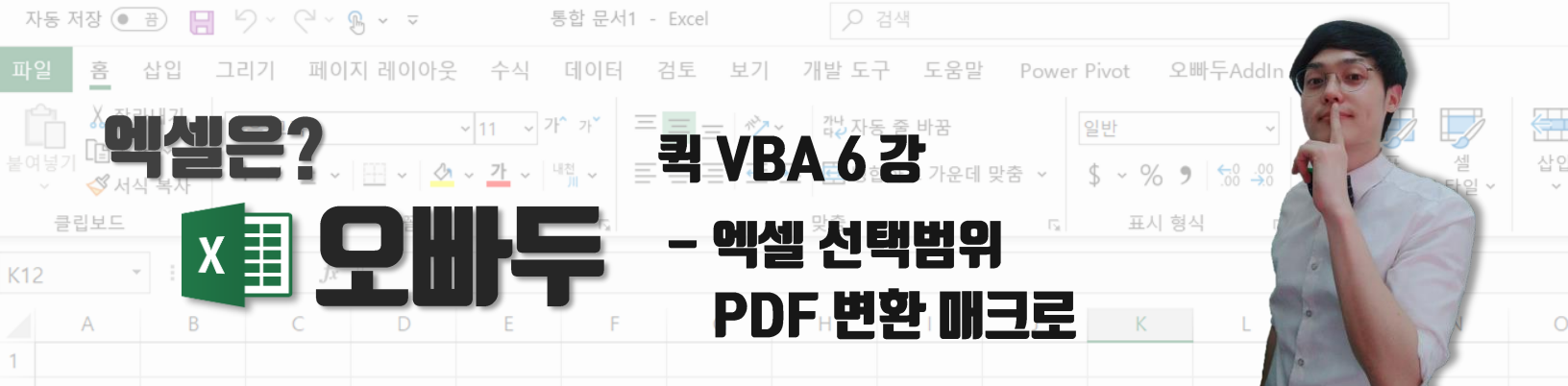
FilePath 의 경우 ValidFileName 보조명령문을 사용하여 파일경로에 윈도우에서 사용할 수 없는 특수기호가 사용되었는지 여부를 검토한 뒤 다음단계로 넘어갑니다.

3-3. 파일 저장경로 순번 지정

명령문의 AddSequence 값이 True 로 입력되면, 저장경로에 중복된 파일이 있을 경우 파일이름 뒤에 순번을 넣어서 저장합니다. (반대로 False 일 경우 파일을 덮어쓰기 합니다.)

따라서, AddSequence 값이 True 일 경우 FileSequence 함수를 통해 PDF 파일 저장경로에 순번을 지정합니다.

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



3-4. 선택범위 PDF 출력

선택된 범위를 ExportAsFixedFormat 함수를 사용하여 PDF 파일로 출력합니다. ExportAsFixedFormat 에 대한 자세한 설명은 링크를 참고하세요.

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

ExportAsFixedFormat 에 들어가는 인수는 아래와 같습니다. 각각의 인수는 위에서 이미 받아왔으므로 그대로 대입해주기만 하면 간단히 마무리됩니다.

인수	형식	설명
Type	XIFixedFormatType	범위를 PDF 형식 또는 XPS 형식으로 추출할지 결정
FileName	String	저장할 파일 경로입니다.
Quality	XIFixedFormatQuality	PDF 또는 XPS 의 인쇄 품질입니다.
IncludeDocProperties	Boolean	문서 속성 포함 여부 입니다.
IgnorePrintAreas	Boolean	기존 설정된 인쇄영역 무시 여부입니다.
From	Long	저장을 시작할 페이지 번호입니다.
To	Long	저장을 종료할 페이지 번호입니다.
OpenAferPublish	Boolean	추출 후 파일 실행여부입니다.
FixedFormatExtClassPtr	Object	다른 형식으로 다시 저장될 수 있도록 FixedFormatExt 클래스로 내보내는 선택 인수입니다.



www.oppadu.com

오빠두엑셀 무료 E-Book

3-5. PDF 저장 매크로 전체 명령문

```

'#####
'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

```

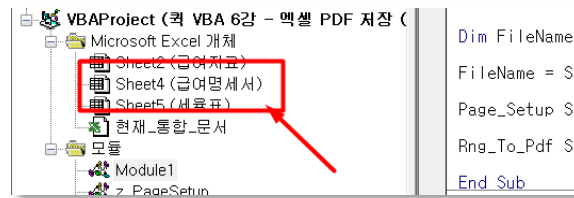


www.oppadu.com

오빠두엑셀 무료 E-Book

4. 작성한 명령문 테스트하기

선택범위를 PDF 로 추출하는 명령문이 잘 동작하는지 테스트합니다. 아래와 같이 테스트 명령문을 작성합니다. 본 강의에 사용된 예제파일에서 [급여명세서] 시트는 Sheet4 에 해당합니다. 따라서 Sheet4 를 기준으로 인쇄 레이아웃을 설정한 뒤 선택한 범위를 PDF 로 추출한다는 가정하에 명령문을 작성합니다.



강의에서 사용된 예제파일의 [급여명세서] 시트의 시트 ID 는 'Sheet4' 입니다.

Page_Setup (인쇄 레이아웃 설정) 모듈에 대한 자세한 설명은 관련 링크를 참고하세요.

```

Sub Test ()

Dim FileName As String

FileName = Sheet4.Range("E3").Value & "-" & Sheet4.Range("H3").Value & "년 " &
Sheet4.Range("H4").Value & "월"

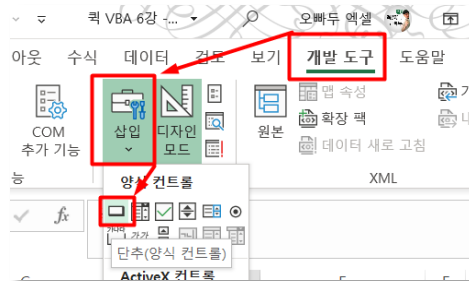
Page_Setup Sheet4, FileName, HCenter:=False

Rng_To_Pdf Selection, FileName, AddSequence:=False

End Sub

```

명령문을 작성한 뒤 엑셀의 [급여명세서] 시트로 이동합니다. 이후 상단 [개발도구] - [삽입]에서 '양식컨트롤'의 [단추]를 추가합니다.



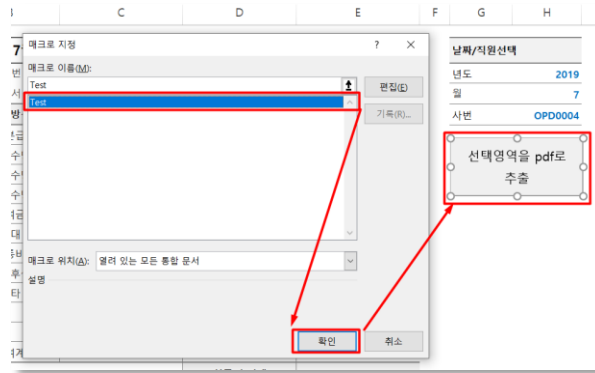
[개발도구] - [삽입] 에서 양식컨트롤 단추를 추가합니다.



www.oppadu.com

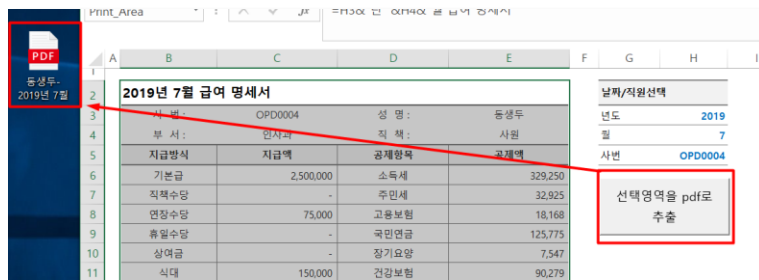
오빠두엑셀 무료 E-Book

새로운 팝업창 목록에서 우리가 방금 작성한 Test 명령문을 선택합니다. 이후 단추의 이름을 본인이 원하는 단추이름으로 변경합니다. 예제에서는 '선택영역을 pdf로 추출'이라고 변경하였습니다.



방금 작성한 Test 명령문을 단추에 연결한 뒤, 단추이름을 원하는 문장으로 변경합니다.

[급여명세서] 시트에서 범위를 선택 후 버튼을 클릭하면 바탕화면에 직원명과 년도 및 월이 적힌 파일명으로 PDF 파일이 저장됩니다.



버튼을 클릭하면 선택한 범위가 바탕화면에 PDF 파일로 저장됩니다.

PDF 파일을 실행합니다. 좌측 상단에는 직원명과 해당기간, 우측상단에는 PDF 를 저장한 시간이 출력됩니다. 머리말 및 꼬릿말은 Page_Setup 모듈을 사용하여 본인이 원하는 형태로 마음껏 변경해서 사용할 수 있습니다.



머리말과 꼬릿말 등의 다양한 인쇄 레이아웃 설정도 변경할 수 있습니다.



5. 시트 전체 또는 고정된 범위를 PDF 로 저장하는 매크로 명령문 (응용)

5-1. 시트에서 사용 중인 전체 범위를 PDF 로 출력하기

워크시트의 UsedRange 속성을 사용하여 시트에서 사용중인 범위를 자동으로 받아와 PDF 로 저장합니다. UsedRange 에 대한 설명은 마이크로소프트 홈페이지 관련 링크를 참고하세요.

```
Sub Test ()

Dim FileName As String

FileName = Sheet4.Range("E3").Value & "-" & Sheet4.Range("H3").Value & "년 " &
Sheet4.Range("H4").Value & "월"

Page_Setup Sheet4, FileName, HCenter:=False

'# Rng_To_Pdf 의 저장범위를 Sheet4 에서 사용중인 전체 범위로 입력합니다.
'# 시트 UsedRange 에 대한 설명은 아래 링크를 참고하세요.
'# https://docs.microsoft.com/en-us/office/vba/api/excel.worksheet.usedrange

Rng_To_Pdf Sheet4.UsedRange, FileName, AddSequence:=False

End Sub
```

5-2. 고정된 범위를 PDF 로 출력하기

시트의 고정된 범위를 PDF 로 저장할 수도 있습니다. 아래 예제는 [급여명세서] 시트의 'B2:E18' 범위를 지정하여 PDF 로 출력하는 명령문입니다.

```
Sub Test ()

Dim FileName As String

FileName = Sheet4.Range("E3").Value & "-" & Sheet4.Range("H3").Value & "년 " &
Sheet4.Range("H4").Value & "월"

Page_Setup Sheet4, FileName, HCenter:=False

'# 시트의 특정 범위를 지정하여 PDF 로 저장할 수도 있습니다.
Rng_To_Pdf Sheet4.Range("B2:E18"), FileName, AddSequence:=False

End Sub
```