

VBA

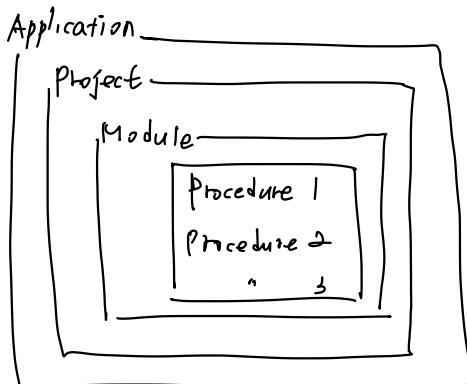
Alt + F11 (작업장)

결과: 도구 → 블록

```

Sub
:
End Sub
    ] Sub procedure. (ex. 54주20주)
  
```

보기 → 작업장을 통한 예가 ⇒ 코드 만드는 학습이면서 작업 가능.



작업: '모임 (작업장에서)

프로젝트: 셀 → 프로젝트.

1. Sub Procedure 작성

Public Sub

Range("a1").Select

ActiveCell.Font.Size=20

ActiveCell.Font.Color=vbBlue

End Sub

2. Function Procedure 작성

Public Function 나의계산(Num1, Num2)

‘함수’

 나의계산 = (Num1+Num2) * 10

End Function

(엑셀창에서) -> 함수마법사 -> 함수삽입

함수선택하고 확인 버튼

3. Event Procedure

Sheet1모듈 클릭

왼쪽상단목록에서 (일반) 말고 Worksheet 선택

-> 이벤트선택

Private Sub Worksheet_Activate()

 MsgBox Sheet1.Name

End Sub

변수

- Dim, Private, Public, Static

Dim 변수명 as 데이터형식 (Dim I As Integer)

Option Explicit 문 모듈 상단에 있는 경우 변수는 반드시
선언 후에 사용해야 함

데이터형식

- 논리: Boolean
- 숫자: Byte, Integer, Long, Currency, Decimal, Single, Double
- 날짜: Date
- 문자: String
- 개체: Object
- 기타: Variant

상수

- 항상 같은 값을 저장하는 공간
- # Const 상수명 As 데이터형식 = 값
Const 할인율 As Single = 0.3

VBA 개체모델

개체: 엑셀vba를 구성하는 요소

- Application, Workbook, Worksheet, Range 개체 등

속성: 개체가 가지는 특성 . (ex. 셀의 주소, 글꼴색, 워크시트이름 등)

- 개체.속성 = 속성 값(Sheet1.Name=“1월”, Range(“a1”).Value=10)

메서드: 개체가 실행할 수 있는 동작 (ex. 워크시트삽입, 삭제, 삭제, 셀복사, 이동 등)

- 개체.메서드 (Sheets.Add, Range(“a1”).Select)

개체변수

: 변수에 값이 아닌 개체를 할당(셀범위, 워크시트, 차트 등)

Dim 변수명 As 개체

Set 변수명 = 개체

Dim Sht As Worksheet

Set Sht = Workbook(“Sheet1”)

Dim 매출실적 As Range

Set 매출실적 = Activesheet.Range(“A1:A10”)

```
Dim rng As Range  
Set rng = Range("a5:a10")  
rng.Value = 10  
rng.Font.Color = vbRed
```

‘ 이렇게 쓰는 것보다는 위처럼 변수 선언해서 쓰는 것이 좋음
Range("a5:a10").Value = 10

Application 객체

```
Sub Application객체1()  
    ‘Application.StatusBar = “실행중…”  
    Application.StatusBar = False # 초기화  
  
    Range("a1:a10").Copy  
    Range("c1").PasteSpecial  
    Application.CutCopyMode = False # 복붙 후 깜빡거리는 걸 없애기  
  
    Application.ScreenUpdating = False  
    Application.DisplayAlerts = False # 알럿 안보이게 하기  
    Sheet1.Delete  
    Application.ScreenUpdating = True  
    Application.DisplayAlerts = True  
  
End Sub
```

WorkSheet Function 엑셀 워크시트함수를 VBA에서 사용

```
Sub applicationObjectTest()
```

```
    Range("c2").Value =  
    Application.WorkSheetFunction.Sum(Range("a1:a10"))
```

```
End Sub
```

```
# input box, match 실습
```

```
Sub 조회()
```

```
    Dim 조건 As String  
    Dim RowNo As Integer  
    조건 = Application.Inputbox ("사번을 입력하세요", "사원조회", "디폴트  
값")  
    RowNo = Application.WorkSheetFunction.Match(조건, Range("b:b"),  
0)  
    Range("b" & RowNo).Select
```

```
End Sub
```

단축접근자

```
# 직접실행창  
?ActiveCell.Value  
>>55  
?ActiveWorkbook.Name  
>>Application객체.xlsm
```

```
?Activesheet.name  
>>Sheet1
```

```
range("a5").select  
?selection.value  
>>5
```

VBA 주요객체 다루기

```
# Workbook 객체
```

```
Sub 파일취합()  
    Workbooks.open filename:="C\~\~\파일명.xlsx", ReadOnly:=True  
    Workbooks.open filename:= ThisWorkbook.Path & "\파일명.xlsx" # 현재  
    워크북경로에 위치한 파일 열때  
  
End Sub
```

Worksheet 개체

Option Explicit

```
Sub sample_01()
    Sheet1.Select
    # Sheets(1).Select
    #Worksheets(1).Select
    #Sheets("연습").Select
End Sub
```

```
Sub sample_02()
    Sheets.Add After:=ActiveSheet # 현시트뒤에 시트추가
    Sheets.Add After:=Worksheets("연습")
    Sheets.Add After:=Worksheets(Worksheets.Count) #맨뒤에

    Worksheets("연습").Activate
    Worksheets(Array("sheet1", "sheet2", "sheet3")).Select
    Worksheets("sheet2").Activate
```

```
Application.DisplayAlerts=False
Worksheets("sheet5").Delete # 삭제
Application.DisplayAlerts=True
```

```
Sheets("sheet1").Visible=True
End Sub
```

```
Sub sample_04()
    ActiveSheet.Protect Password:="1111"
    ActiveSheet.Unprotect Password:="1111"
End Sub
```

Range 개체

```
Sub sample_00()
    Range("b3:e10").Select
    # Cells(3,1).Select
    # Range("b3:d10").Cells(1,1).Select # 이렇게도 가능
    # Range("a4").Activate

    Columns("A:E").Autofit #열너비 자동지정
    Rows.Select # 전체행 선택

    Worksheets("고객명부").Select
    Range("a10").Select
    ActiveCell.EntireRow.Select
    ActiveCell.ColumnWidth = 30
    Range("a3").CurrentRegion.Select
    Range("a3").End(xlDown).Select
    Range("a1048576").End(xlUp).Offset(1,0).Select

    ActiveSheet.Paste
    # 선택하여 붙여넣기
    Selection.PasteSpecial Paste:=xlPasteValues #값붙여넣기
    Selection.PasteSpecial Paste:=xlPasteAll #전체붙여넣기
    Selection.PasteSpecial Paste:=xlPasteAll, Transpose:=True #행열바꿈

    Worksheets("고객명부").Select
    Range("A:A").Find("최소라").Select
    Columns(1).Find("최소라").Select

End Sub

# F5 : 전체 실행
# F8 : 하나씩 실행
# ctrl+j : 가질수있는 프로퍼티, 메서드 확인용
```

With문과 조건문

With 개체

- . 구성원 = 속성

- . 구성원 메서드

End With

Sub sample_01()

 Range("a1").Value = "Excel VBA"

 Range("a1").Font.Name = "나눔고딕"

 Range("a1").Font.Bold = True

 Range("a1").Interior.Color = vbYellow # 내장상수

 #Range("a1").Interior.Color = rgb(255,0,0)

End Sub

Sub sample_02()

 With Range("a1")

 .Value = "Excel VBA")

 .Font.Name = "나눔고딕"

 .Font.Bold = True

 .Interior.Color = vbYellow

 End With

End Sub

Sub sample_03()

 With Sheet1

 .Activate

 .Protect

 .Copy after:=Worksheets("조건문1")

 End With

End Sub

```
# 조건 1개일때  
IF 조건식 Then  
    조건식판단결과가 True일때 실행문  
End IF
```

```
Sub sample_01()  
    If range("e4").Value >= 80 Then  
        Range("f4").Value = "합격"  
    End If  
End Sub
```

```
# 조건 판단결과에 따라 다른값  
IF 조건식 Then  
    조건식판단결과가 True일때 실행문  
Else  
    조건식판단결과가 False일때 실행문  
End IF
```

```
Sub sample_02()  
    If range("e4").Value >= 80 Then  
        Range("f4").Value = "합격"  
        Range("f4").Font.Color = vbBlue  
    Else  
        Range("f4").Value = "불합격"  
        Range("f4").Font.Color = vbRed  
    End If  
End Sub
```

```
Sub sample_03()
    If Range("e4").Value >= 90 Then
        Range("g4").Value = "A"
    ElseIf Range("e4").Value >= 80 Then
        Range("g4").Value = "B"
    ElseIf Range("e4").Value >= 70 Then
        Range("g4").Value = "C"
    ElseIf Range("e4").Value >= 60 Then
        Range("g4").Value = "D"
    Else
        Range("g4").Value = "F"
    End Sub
```

조건의 판단결과에 따른 다른값반환
조건식에 And, Or 논리연산자 사용

```
Sub sample_04()
    If Range("c4").Value >= 90 Or Range("d4").Value >= 90 Then
        Range("h4").Value = "수상"
    Else
        Range("h4").Value = "탈락"
    End If
End Sub
```

```
Sub sample_05()
    If Not Range("l4").Value = "Seoul" Then
        Range("m4").Value = 100000
    Else
        Range("m4").Value = 50000
    End If
End Sub
```

Select ~ Case 문

여러조건 처리할 경우 IF문보다 효과적

```
# Case Is >= 90  
# Case 90 To 100 # 범위조건  
# Case “서울”,“경기” # Or조건  
# Case 조건1 And 조건2
```

Sub sample_06()

```
    Select Case Range(“e4”)  
        Case Is >= 90 # Case 90 To 100  
            Range(“f4”) = “A”  
        Case Is >= 80  
            Range(“f4”) = “B”  
        Case Is >= 70  
            Range(“f4”) = “C”  
        Case Is >= 60  
            Range(“f4”) = “D”  
        Case Else  
            Range(“f4”) = “F”  
    End Select
```

End Sub

Sub sample_07()

```
    Select Case Range(“j4”).Value  
        Case “서울”  
            Range(“k4”).Value = 50000  
        Case “경기”,“인천”  
            Range(“k4”).Value = 100000  
        Case Else  
            Range(“k4”).Value = 200000  
    End Select
```

End Sub

```
Sub sample_08()
    Select Case True
        Case Range("e4").Value >= 90 and Range("e4").Value <= 100
            Range("f4").Value = "A"
        Case Range("e4").Value >= 80 and Range("e4").Value <= 90
            Range("f4").Value = "B"
        ...
    Case Else
        Range("f4").Value = "F"
    End Select
End Sub
```

반복문

```
# For ~ Next 문
#
For 카운터변수 = 시작값 To 끝값 [Step 증감값]
    반복할실행문
Next [카운터변수]
```

```
Sub sample_02()
```

```
# For ~ Next
```

```
Dim i As Integer
```

```
Dim RowCnt As Long
```

```
# RowCnt = Range("b3").End(xlDown).Row
```

```
# RowCnt = Range("a1048576").End(xlUp).Row # 중간중간데이터가 비어있을 리스트가 있는경우 사용
```

```
RowCnt = Range("b3").CurrentRegion.Rows.Count + 2
```

```
# For i = 4 To 13
```

```
For i = 4 To RowCnt
```

```
If range("e" & i).Value >= 80 Then
```

```
    Range("f" & i).Value = "합격"
```

```
    Range("f" & i).Font.Color = vbBlue
```

```
Else
```

```
    Range("f" & i).Value = "불합격"
```

```
    Range("f" & i).Font.Color = vbRed
```

```
End If
```

```
Next i
```

```
End Sub
```

For ~ Next 문 (2)

For Each Next 문

Dim 개체변수명 As 개체

For Each 개체변수 In 컬렉션

 실행문

Next

개체들의 집합(컬렉션)의 개별요소에 대한 반복작업 실행

통합문서내의 워크시트를 반복하면서 각 시트명을 셀에 입력

선택한 범위의 각 셀을 순환하면서 셀 서식 지정

Sub 합격여부()

Dim Rng합격여부 As Range

For Each Rng합격여부 In Range("f4:f13")

 If Rng합격여부.Offset(0,-1).Value >= 80 Then

 Rng합격여부.Value = "합격"

 Else

 Rng합격여부.Value = "불합격"

 End If

Next

End Sub

Sub 합격여부4()

```
Dim Sht As Worksheet  
Dim Rng합격여부 As Range  
Dim Rng As Range
```

```
Set Sht = Worksheets("반복문1")  
Set Rng합격여부 = Sht.Range("F4:F13")
```

```
For Each Rng합격여부 In Range("f4:f13")  
    If Rng합격여부.Offset(0,-1).Value >= 80 Then  
        Rng합격여부.Value = "합격"  
    Else  
        Rng합격여부.Value = "불합격"  
    End If  
Next  
End Sub
```

Sub 시트목록()

```
Dim Sht As Worksheet  
Dim i As Integer  
i = 2  
Sheets("반복문2").Select  
For Each Sht in Worksheets  
    Range("A" & i).Value = Sht.Name  
    i = i + 1  
Next  
End Sub
```

Do ~ Loop 문

조건 만족하는 동안 실행문 반복

Do Until ~ Loop

조건 만족할 때까지 실행문 반복

Sub 합격여부_0()

```
Sheets("반복문4").Select
```

```
Range("F4").Select
```

```
Do While ActiveCell.Offset(0,-1).Value <> "" #비어있지않을때까지 반복
```

```
    If ActiveCell.Offset(0,-1).Value >= 80 Then
```

```
        ActiveCell.Value = "합격"
```

```
    Else
```

```
        ActiveCell.Value = "불합격"
```

```
    End If
```

```
    ActiveCell.Offset(1,0).Select # 다음셀선택해주는거까지
```

```
Loop
```

```
End Sub
```

Sub 합격여부_0()

```
Sheets("반복문4").Select
```

```
Range("F4").Select
```

```
Do Until ActiveCell.Offset(0,-1).Value = "" #비어있지않을때까지 반복
```

```
    If ActiveCell.Offset(0,-1).Value >= 80 Then
```

```
        ActiveCell.Value = "합격"
```

```
    Else
```

```
        ActiveCell.Value = "불합격"
```

```
    End If
```

```
    ActiveCell.Offset(1,0).Select # 다음셀선택해주는거까지
```

```
Loop
```

```
End Sub
```

오류제어문과 디버깅

오류 종류

구문(Syntax)오류 (Compile Error)

- 줄의 구문이 올바르지 않은 경우 발생

컴파일 오류

- 여러라인에 걸쳐 발생
- 명령문을 실행하거나 컴파일 할 때 확인 가능
- 디버그-vbaProject 컴파일

ex. End With, End If, Next, Loop 누락, Option Explicit 선언한 상태에서 변수 선언하지 않고 사용한 경우 등

런타임 오류

- 코드가 실행되는 동안 발생하는 오류. 실행문이 잘못된 동작을 시도할 때 발생
 - 오류창에 표시되는 오류코드를 통해 오류의 원인을 파악할 수 있음.
- ex. 0으로 나누는 경우, 없는파일 여는 경우, 보호된워크시트 개체 참조하는 경우

오류제어문

On Error GoTo 0

- VBA 기본에러처리방식. On Error 구문작성하지 않는 경우 수행
- 에러발생시 해당지점에서 명령문을 멈추고 오류메시지 출력

On Error Resume Next

- 에러발생된 줄을 무시하고 다음 줄 실행

On Error GoTo 레이블

- 에러발생시 지정한 레이블로 이동

On Error GoTo -1

- 발생한 모든 오류를 초기화

On Error GoTo 레이블

실행문

...

Exit Sub

레이블 :

오류발생시 처리할 실행문

[Resume Next]

```
Sub RuntimeErrorTest1()
```

```
    Dim i As Single
```

```
    Dim j As Variant # 문자숫자아무거나 저장 타입
```

```
    j = InputBox("값을 입력하세요", , 3)
```

```
#On Error GoTo -1 # 에러를 초기화
```

```
#On Error Resume Next # 그다음줄 그냥 수행
```

```
On Error GoTo errChk
```

```
    i = 10 / j
```

```
    MsgBox i
```

```
    Exit Sub
```

errChk:

```
If Err.Number = 10 Then
```

```
    MsgBox "0으로 나누었습니다. 다른 숫자를 입력하세요", vbInformation
```

```
Else If Err.Number = 13 Then
```

```
    MsgBox "문자로 나누었습니다. 숫자를 입력하세요", vbInformation
```

```
Else
```

```
    MsgBox "오류가 발생했습니다", vbCritical
```

```
End If
```

```
End Sub
```

디버깅

- 프로그램의 비정상적 연산(버그)이나 논리적 오류를 찾아내고 수정하는 작업

단축키	설명
F5	프로시저 실행
F8	명령문을 한 줄씩 실행
Ctrl+F8	커서 지점까지 실행
F9	중단점 설정
Ctrl+Shift+F9	모든 중단점 지우기
마우스 드래그	코드 창 왼쪽 회색 바의 화살표 드래그

사용자 정의 품

- 사용자 인터페이스를 개발하고 사용자와 상호작용하는 양식(품) 작성
- 다양한 컨트롤(레이블, 텍스트박스, 콤보박스, 목록상자, 확인란, 옵션단추, 명령단추, 스피너 등)을 사용하여 품디자인
- UserForm 및 컨트롤에 기능을 수행하는 이벤트프로시저 작성

VBA - 삽입 - 사용자정의품

➊ 품을 실행하는 단추와 Procedure 작성

➋ 등록

- [등록] 버튼을 누르면
품에 입력한 데이터를 '사원명부' 시트에 저장하고 품 초기화

➌ 닫기

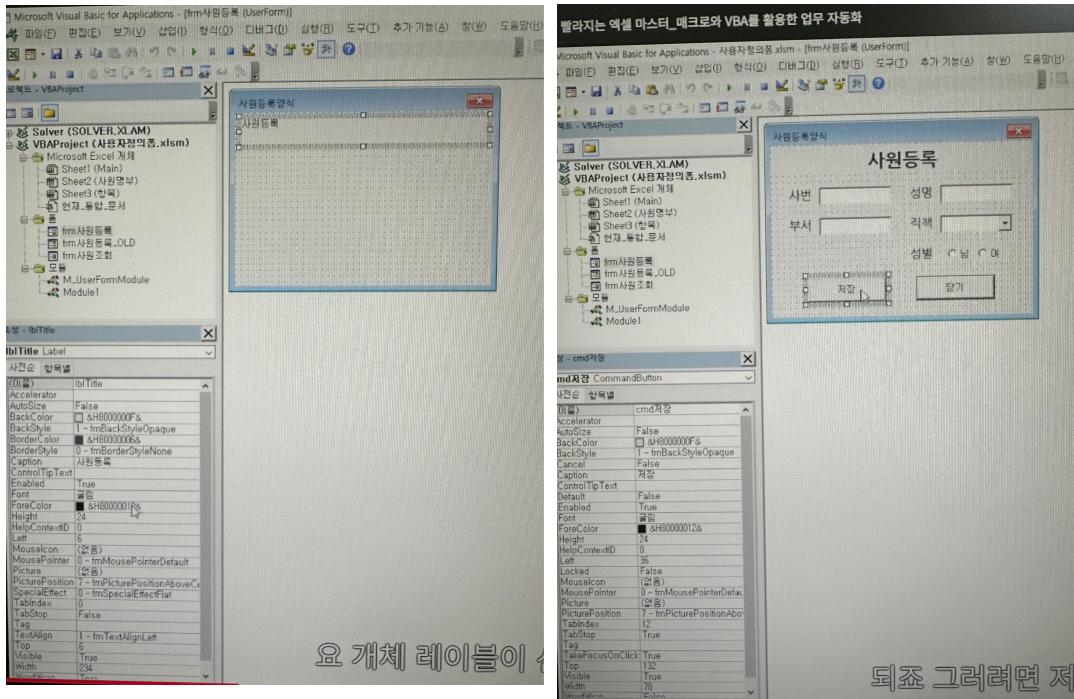
- [닫기] 버튼을 누르면 품 종료

```
Sub 사용등록실행()
    frm사원등록.Show
End Sub

Private Sub cmd등록_Click()
    Sheets("사원명부").Select
    Range("A1048576").End(xlUp).Offset(1, 0).Select
    ActiveCell.Value = txt사번.Value
    ActiveCell.Offset(0, 1).Value = txt성명.Value
    ActiveCell.Offset(0, 2).Value = txt부서.Value
    ActiveCell.Offset(0, 3).Value = cbo직책.Value
    If opt남.Value = True Then
        ActiveCell.Offset(0, 4).Value = "남"
    Else
        ActiveCell.Offset(0, 4).Value = "여"
    End If
    txt사번.Value = ""
    txt성명.Value = ""
    txt부서.Value = ""
    cbo직책.ListIndex = -1
    opt남.Value = False
    opt여.Value = True
End Sub

Sub 종료()
    ActiveWorkbook.Save
    Application.Quit
End Sub
```

그렇게 작성된 품, 유저 품에다가 등록 버튼을 누



버튼 더블클릭시 : 버튼 더블클릭시의 procedure가 자동생성

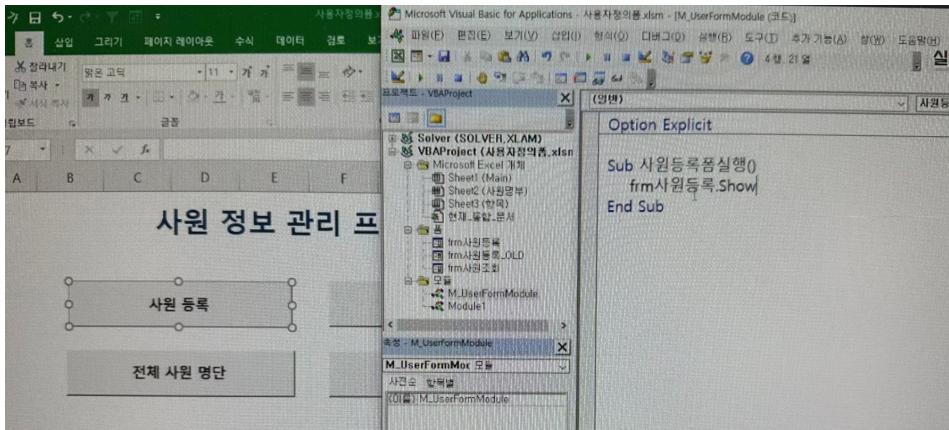
```
Private Sub cmd닫기_Click()
    Unload Me
End Sub
```

```
Private Sub cmd저장_Click()
    Worksheets("사원명부").Select
    Range("a1048576").End(xlUp).Offset(1,0).Select
    ActiveCell.Value = txt사번
    ActiveCell.Offset(0,1).Value = txt성명.Value
    ActiveCell.Offset(0,2).Value = txt부서.Value
    ActiveCell.Offset(0,3).Value = cbo직책.Value

    If opt남 = True Then
        ActiveCell.Offset(0,4).Value = "남"
    Else
        ActiveCell.Offset(0,4).Value = "여"
    End If

    txt사번 = ""
    txt성명 = ""
    txt부서 = ""
    cbo직책.ListIndex = -1
    MsgBox "데이터가 저장되었습니다", vbInformation

End Sub
```



Show

줄을 보여주는 코드

Unload Me

열려 있는 줄을 닫는 코드

me 하시면 된다라고 기억하시며 되세요 자 그럼

