

엑셀 VBA

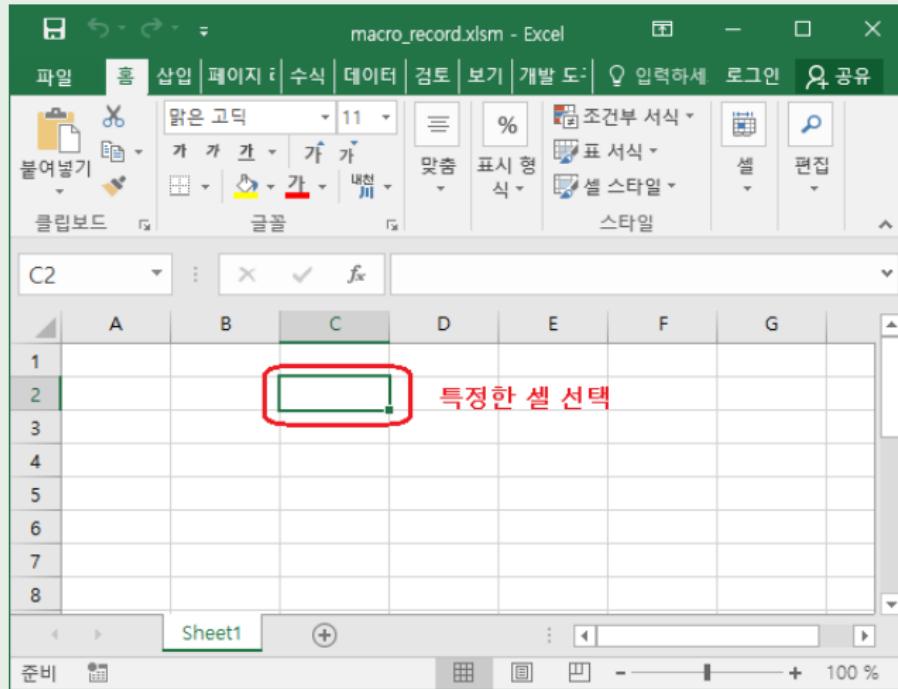
실행과정을 중심으로

hmkang98@naver.com

매크로 작성과 실행

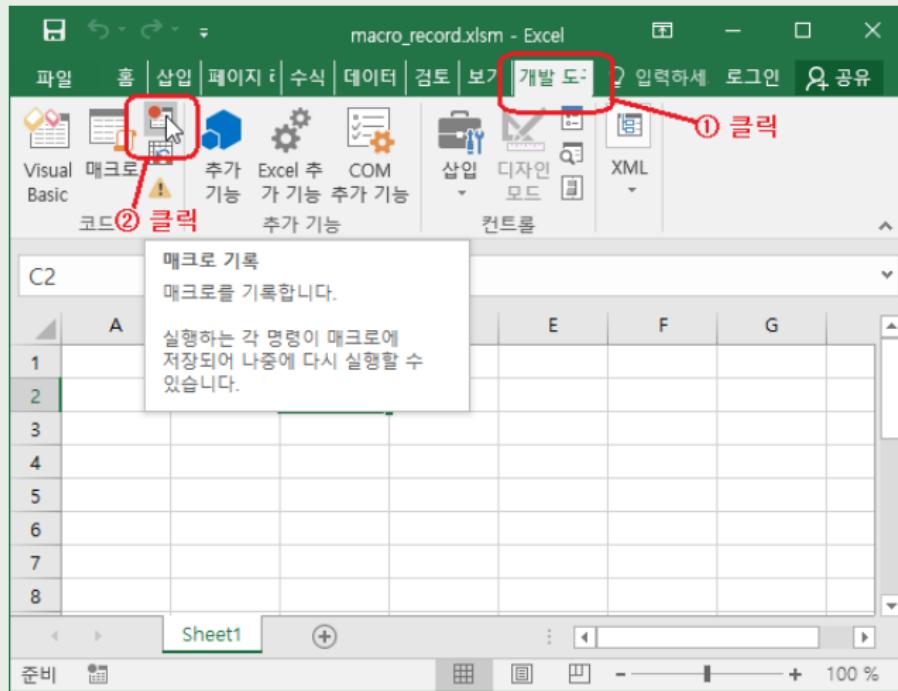
숫자를 자동으로 채우는 매크로 작성

셀 선택



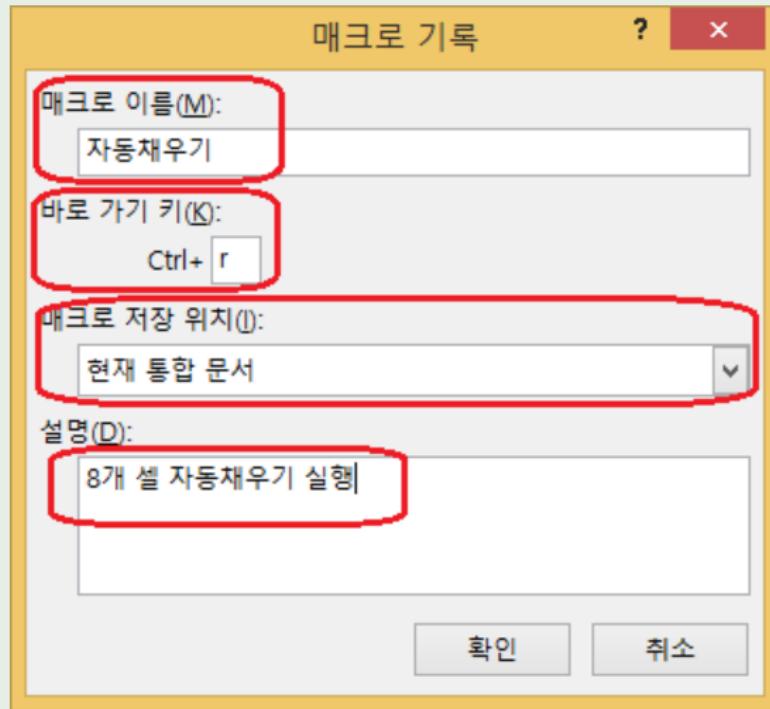
숫자를 자동으로 채우는 매크로 작성

개발도구→매크로 기록 메뉴 실행



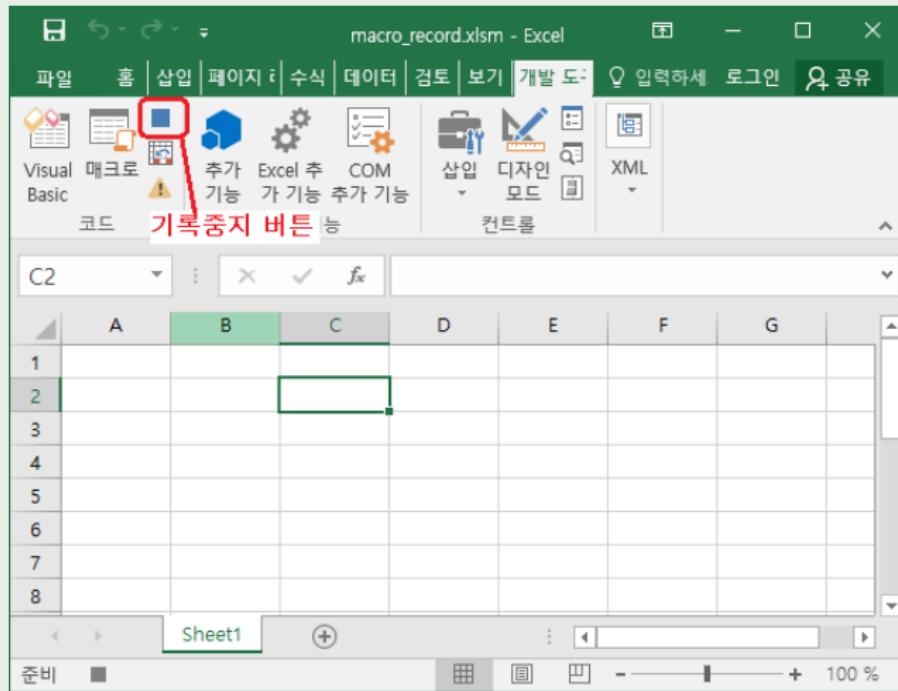
숫자를 자동으로 채우는 매크로 작성

매크로 기록 설정



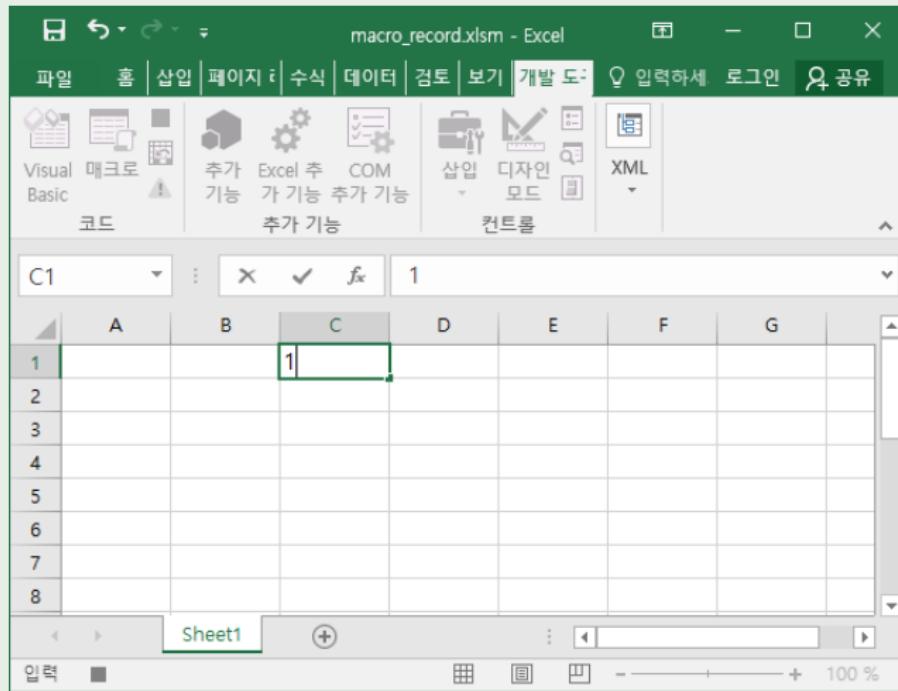
숫자를 자동으로 채우는 매크로 작성

버튼 모양이 매크로 기록에서 기록 중지로 변경



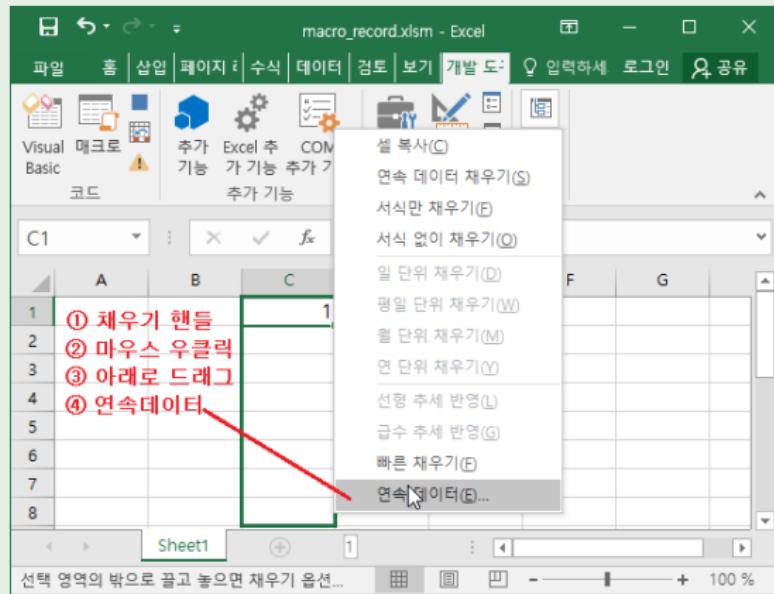
숫자를 자동으로 채우는 매크로 작성

매크로 기록 시작



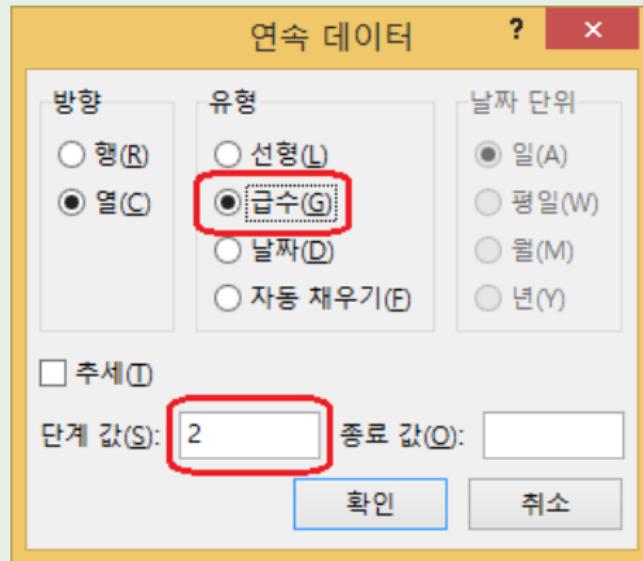
숫자를 자동으로 채우는 매크로 작성

채우기 핸들로 연속 데이터 실행



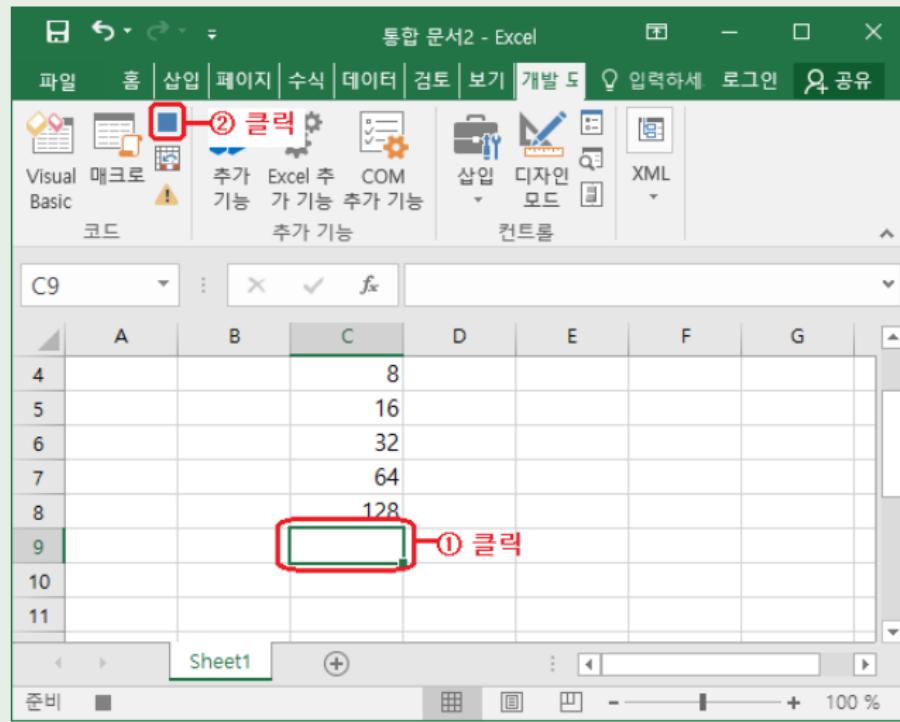
숫자를 자동으로 채우는 매크로 작성

연속 데이터에서 급수 선택하고 단계값은 2



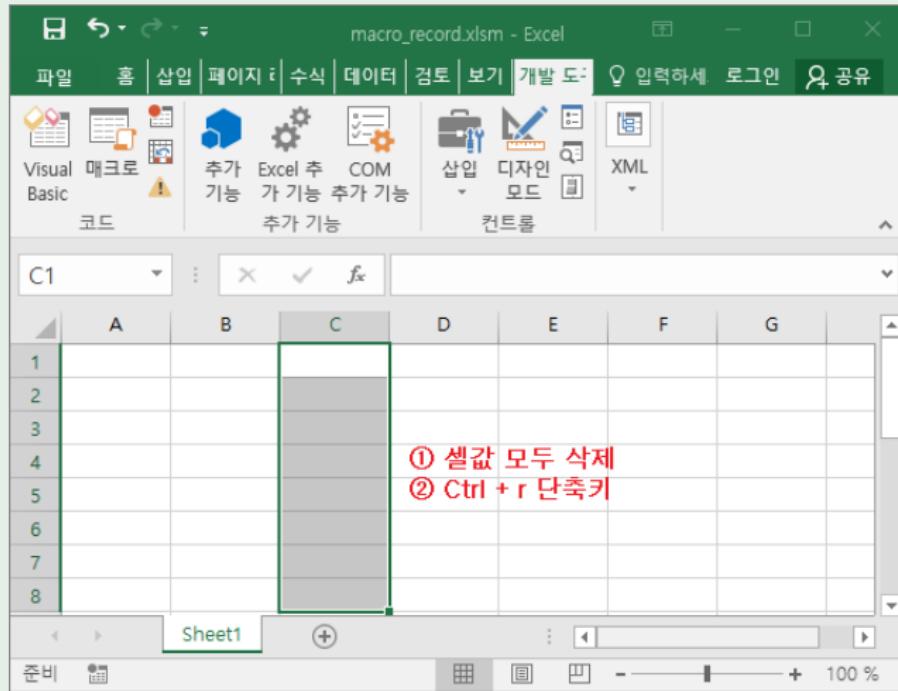
숫자를 자동으로 채우는 매크로 작성

연속 데이터 마치고 C9 셀 클릭 후 기록 정지 클릭



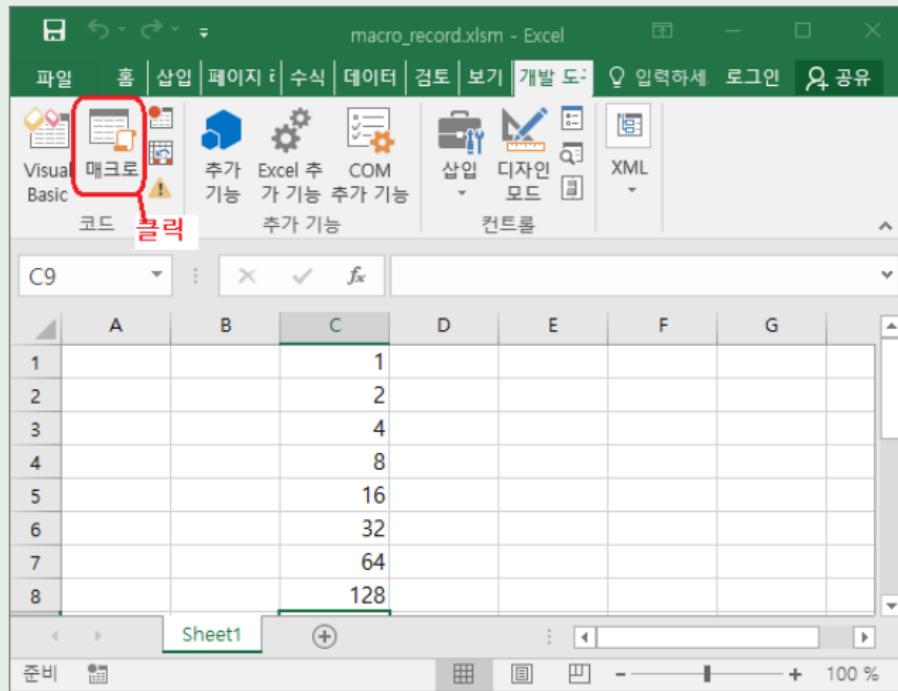
숫자를 자동으로 채우는 매크로 작성

셀 값 모두 삭제하고 매크로 단축키 실행



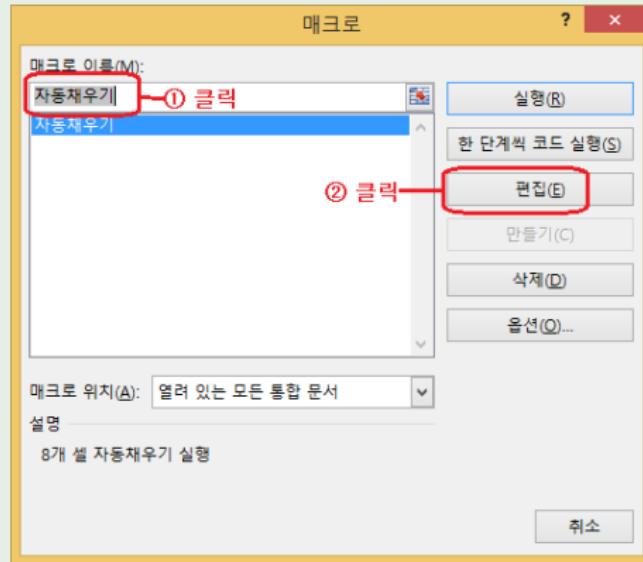
숫자를 자동으로 채우는 매크로 작성

매크로 실행결과 확인하고 코드에서 매크로 선택클릭



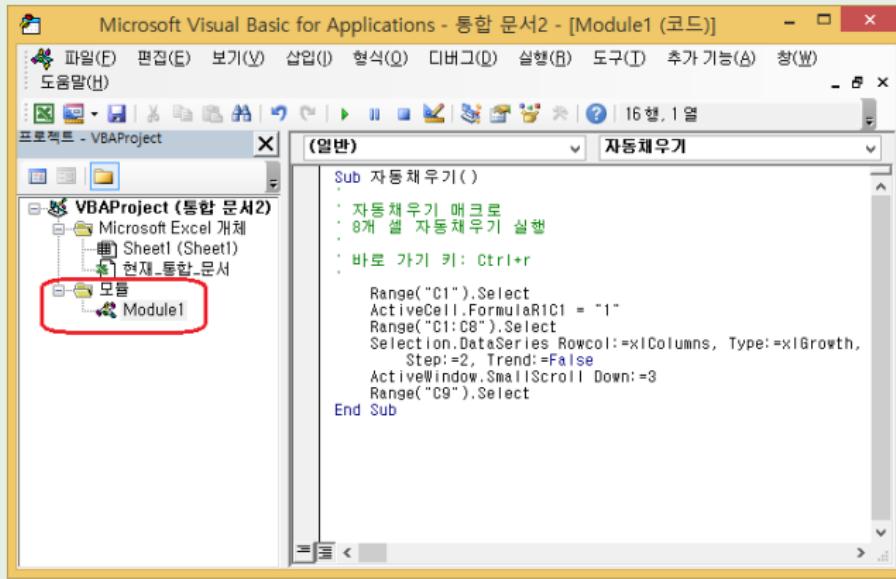
숫자를 자동으로 채우는 매크로 작성

매크로 창에서 추가한 매크로 선택, 편집



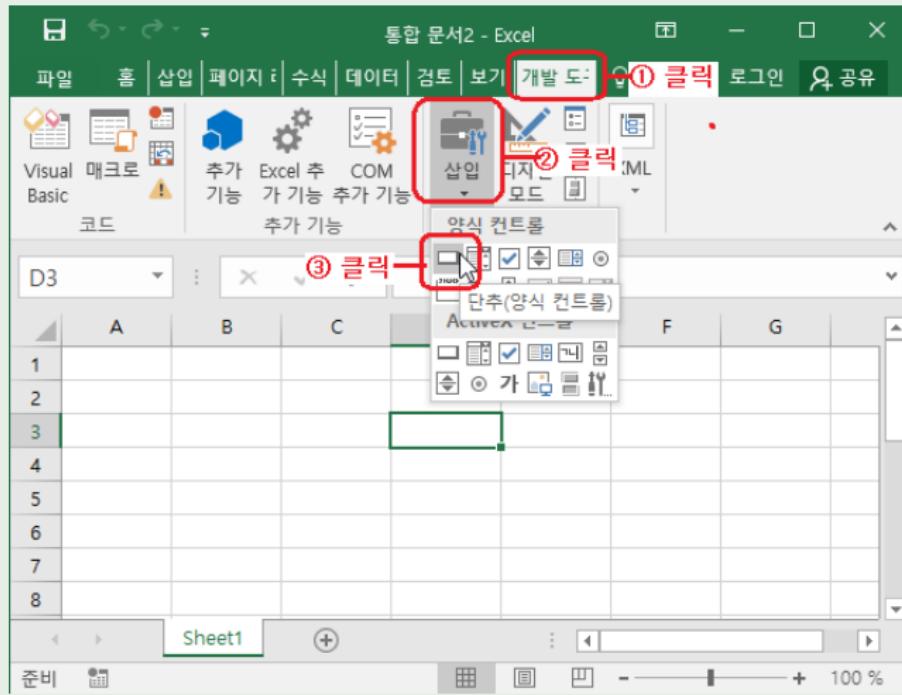
숫자를 자동으로 채우는 매크로 작성

Visual Basic 편집기에서 코드 확인



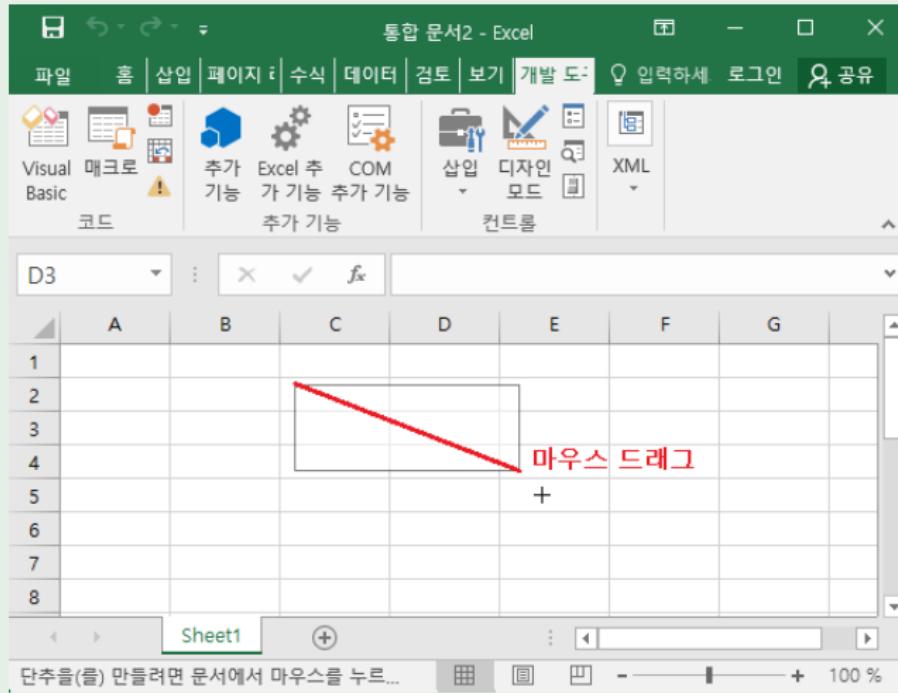
숫자를 자동으로 채우는 매크로 작성

양식 컨트롤에서 명령 단추를 시트에 추가



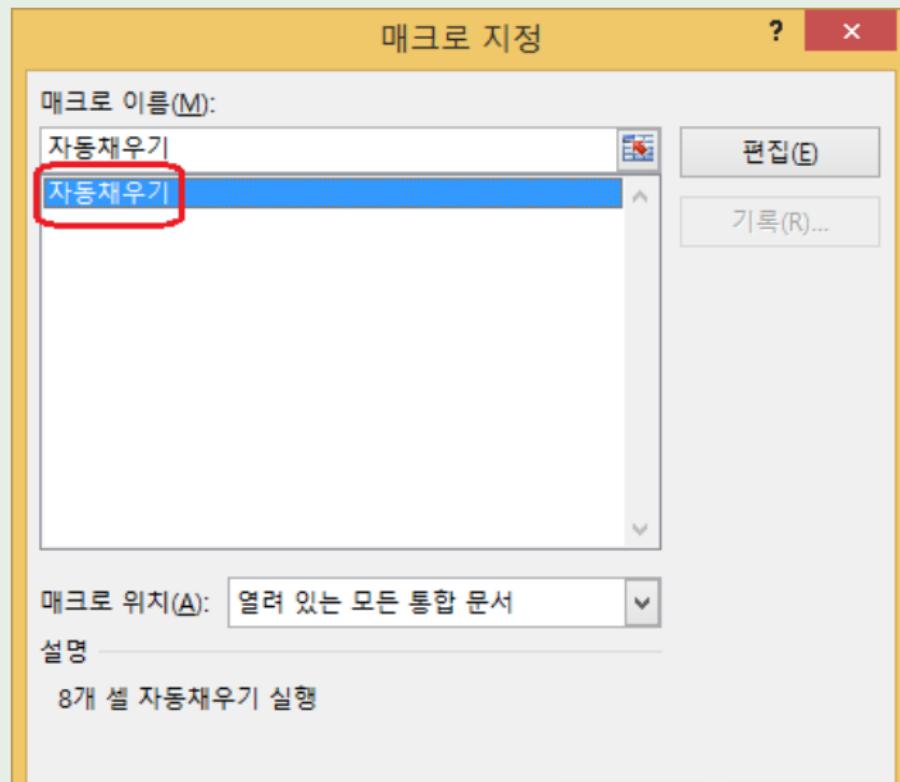
숫자를 자동으로 채우는 매크로 작성

명령 단추가 추가할 영역을 마우스 드래그



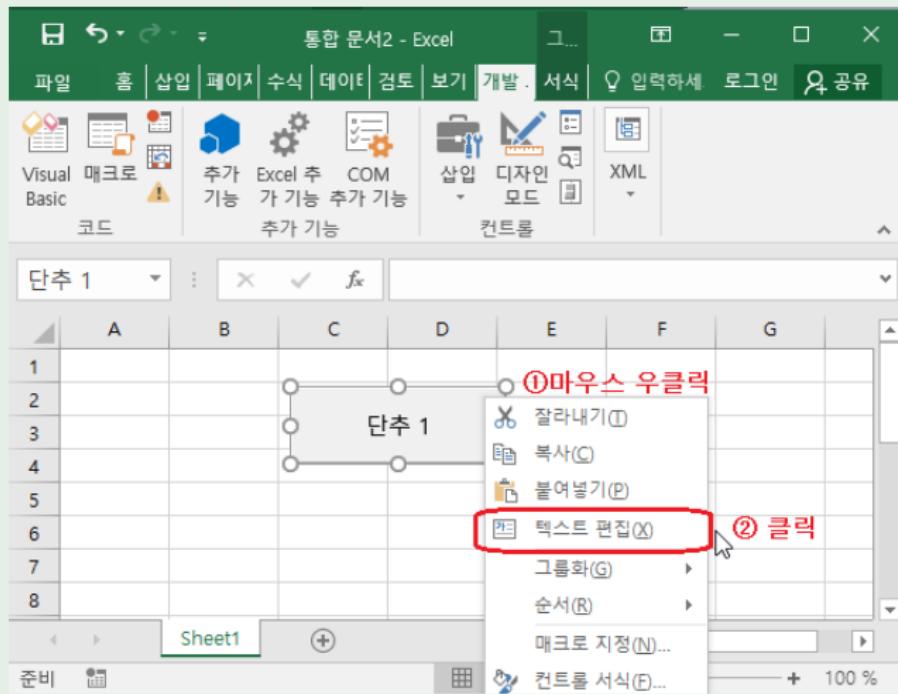
숫자를 자동으로 채우는 매크로 작성

매크로 지정에서 연결하려는 매크로 선택



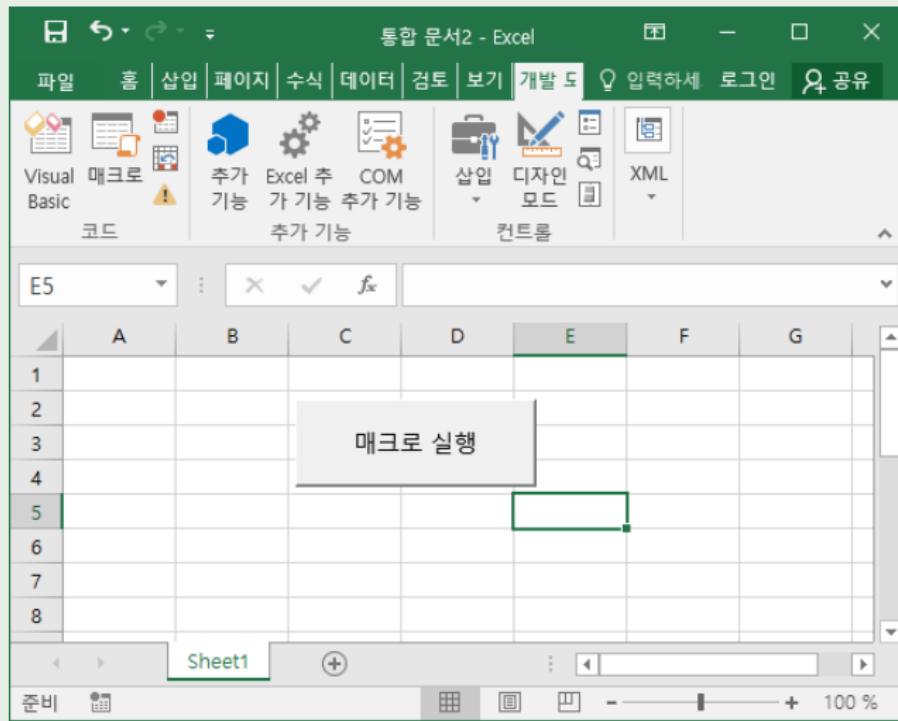
숫자를 자동으로 채우는 매크로 작성

추가한 명령 단추 레이블 편집



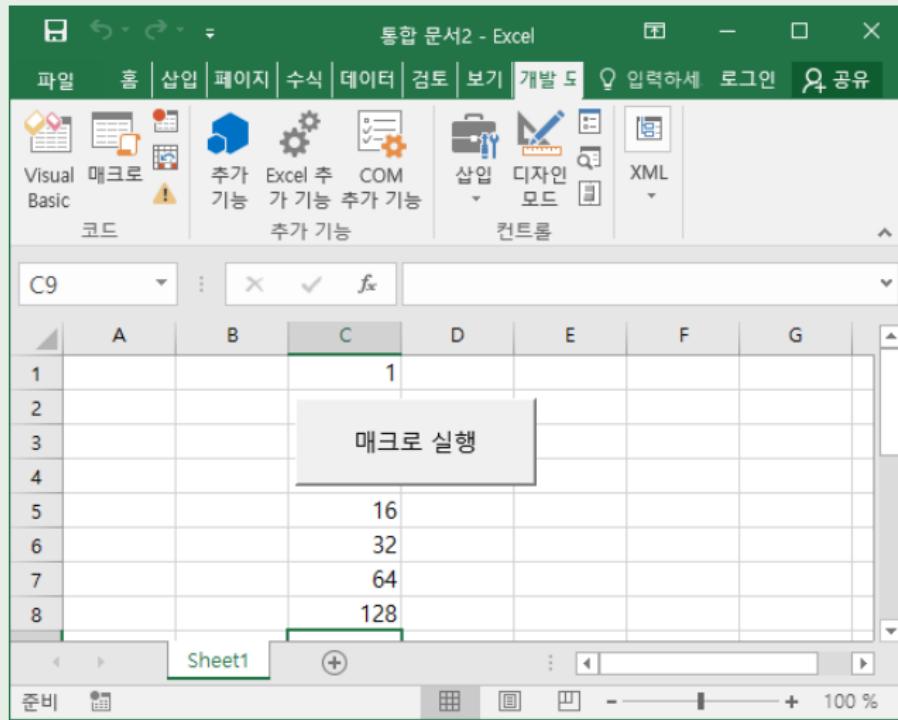
숫자를 자동으로 채우는 매크로 작성

연결한 명령 단추 편집 마침



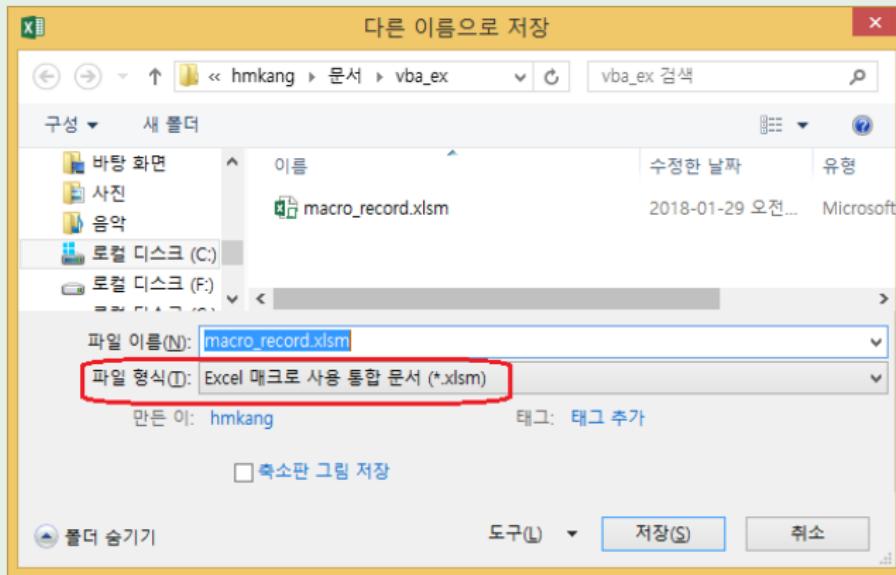
숫자를 자동으로 채우는 매크로 작성

명령 단추에 연결한 매크로로 실행



숫자를 자동으로 채우는 매크로 작성

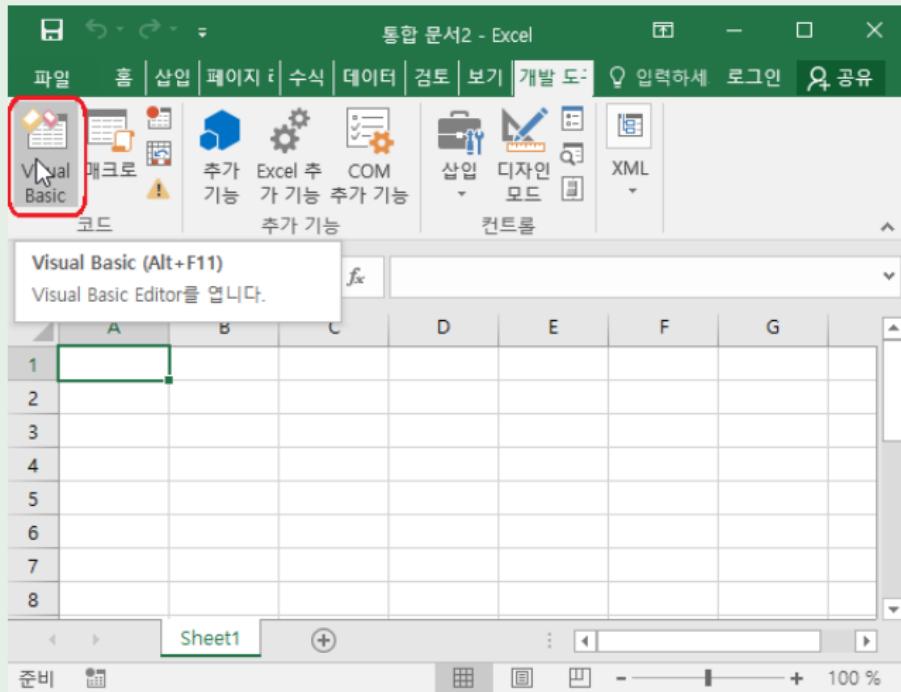
매크로는 Excel 매크로 사용 통합 문서로 저장



Visual Basic Editor

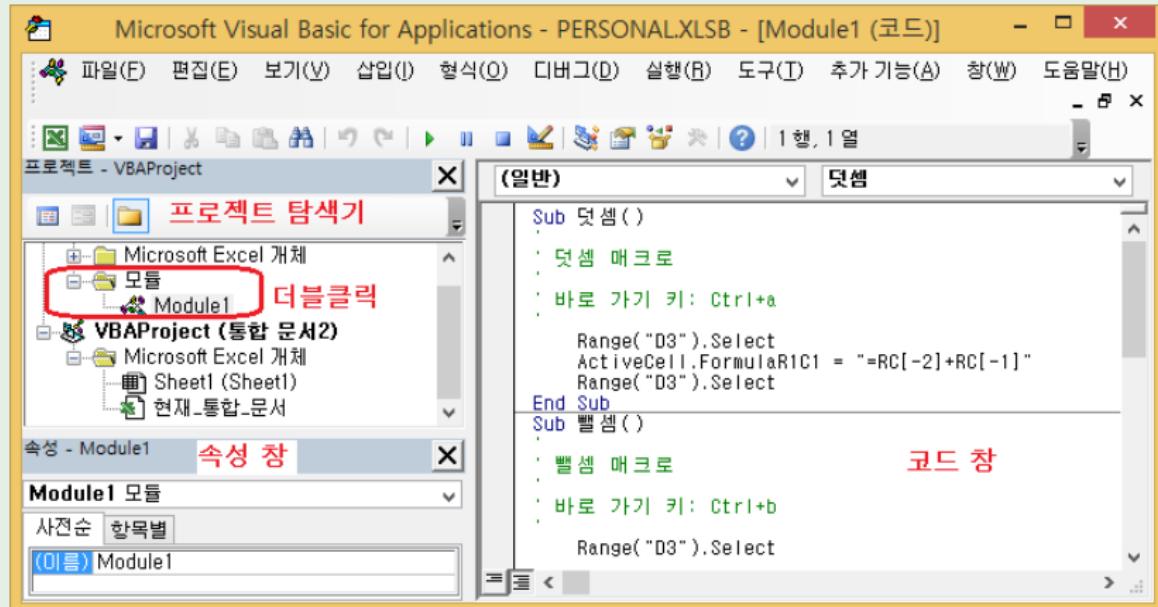
Visual Basic Editor 사용하기

Visual Basic Editor 실행



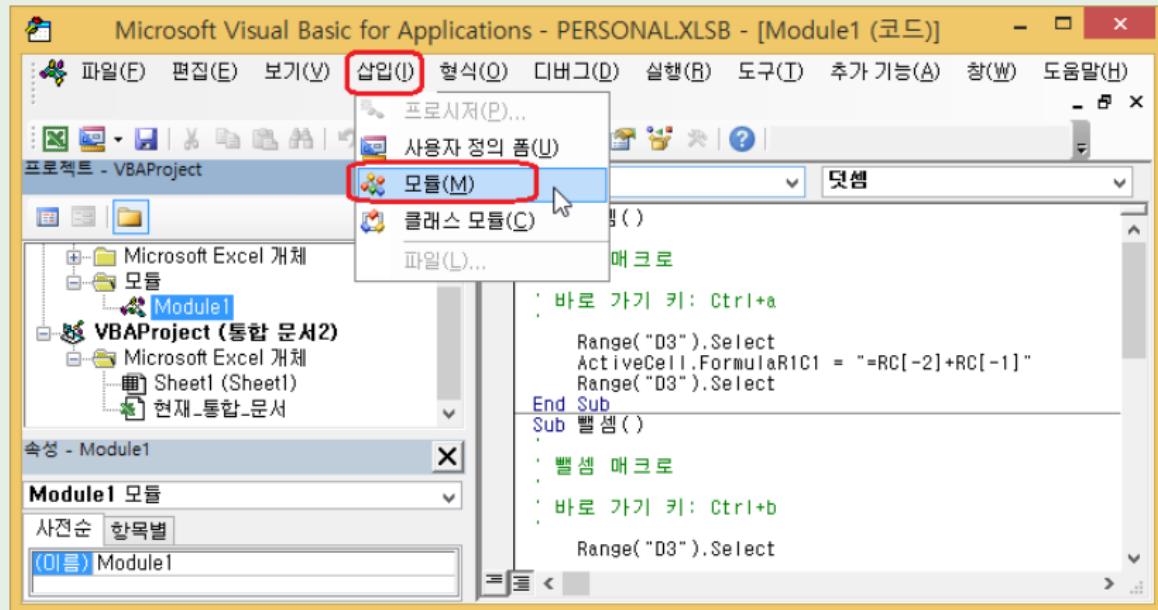
Visual Basic Editor 사용하기

Visual Basic Editor 명칭과 코드 확인



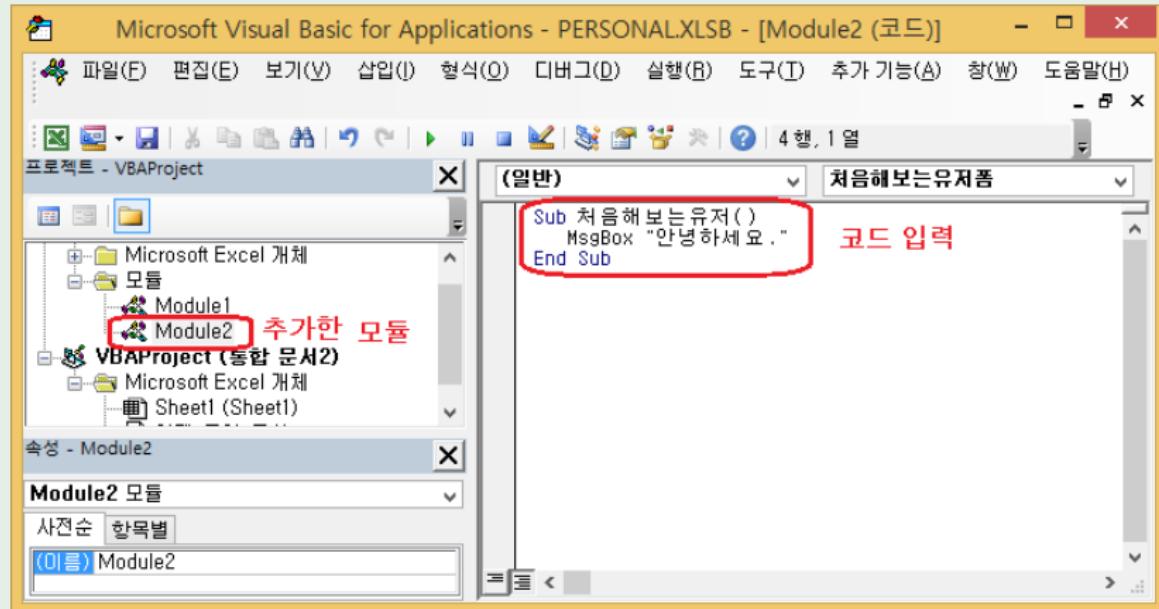
Visual Basic Editor 사용하기

모듈 추가



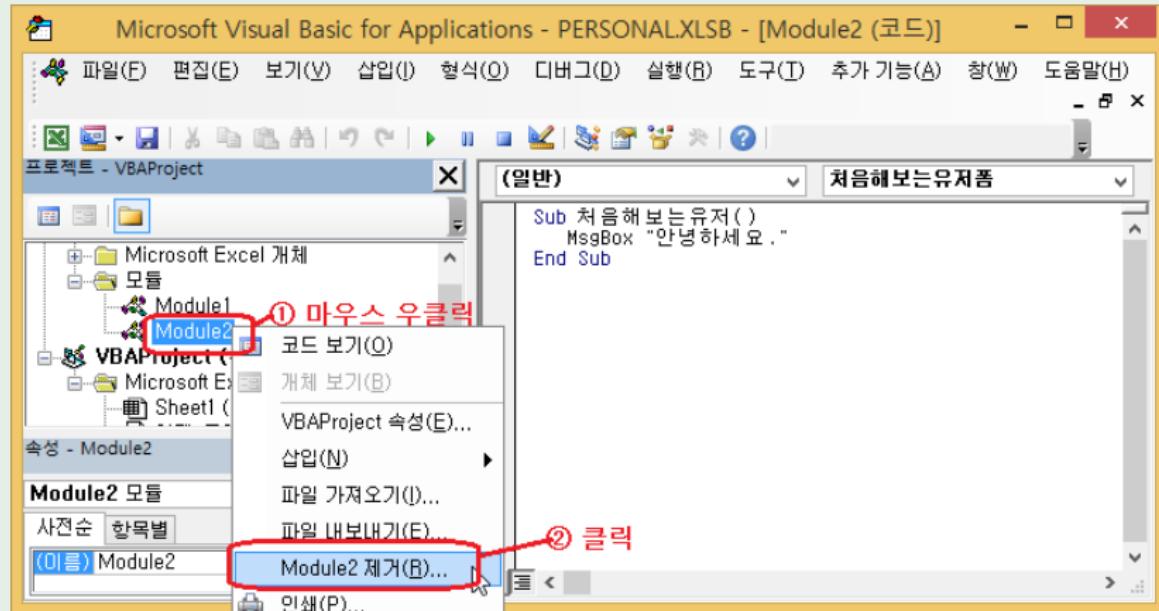
Visual Basic Editor 사용하기

추가한 모듈에 코드 입력



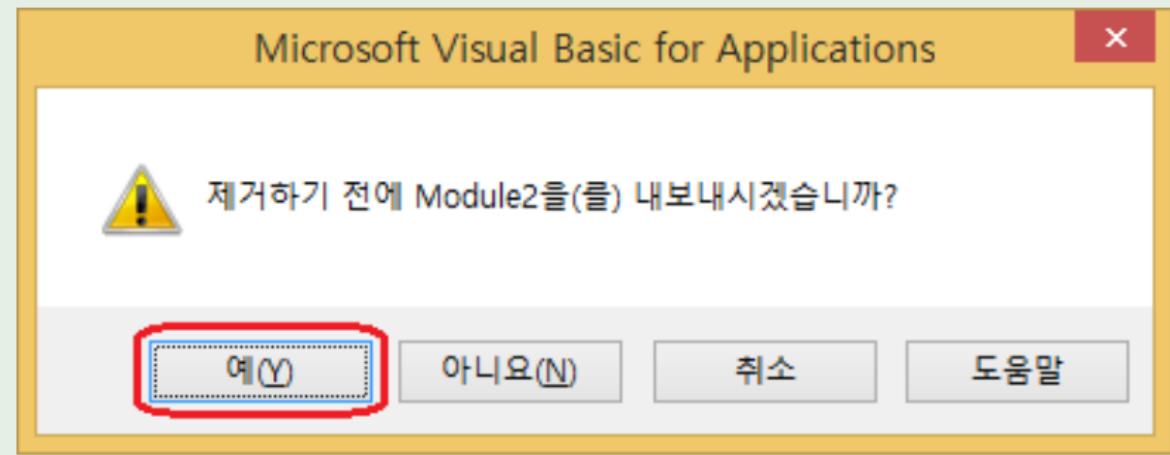
Visual Basic Editor 사용하기

모듈 내보내기와 제거



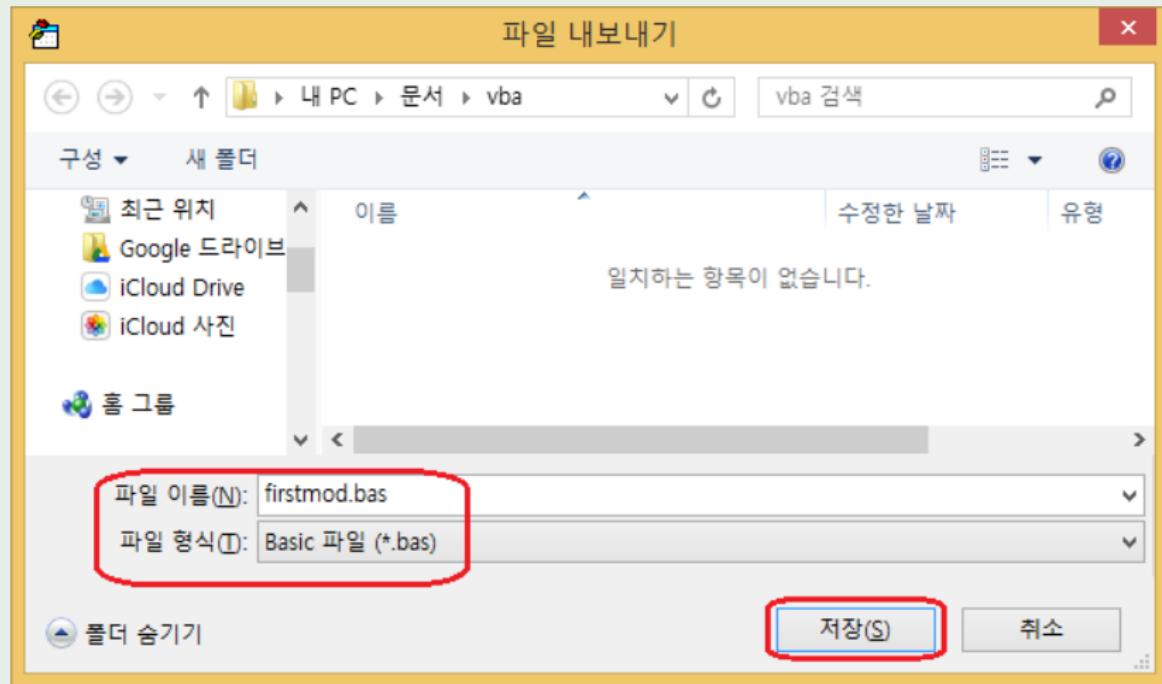
Visual Basic Editor 사용하기

제거하기 전 모듈 내보내기



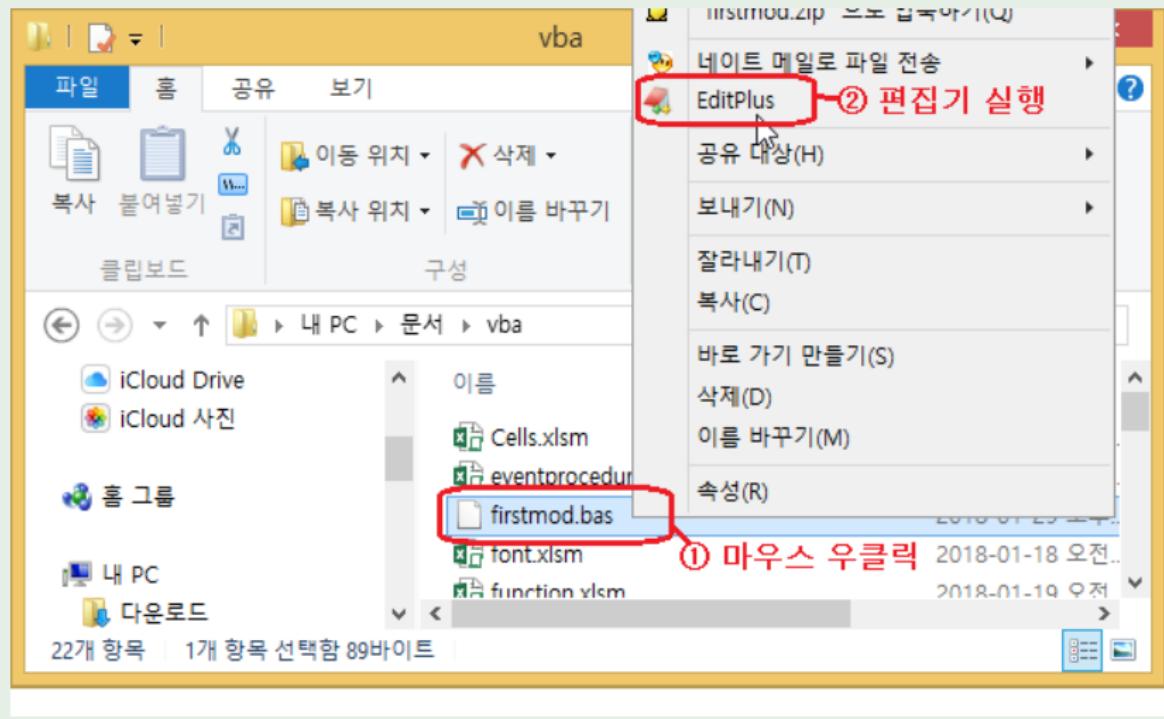
Visual Basic Editor 사용하기

모듈 제거하기 전 파일 저장



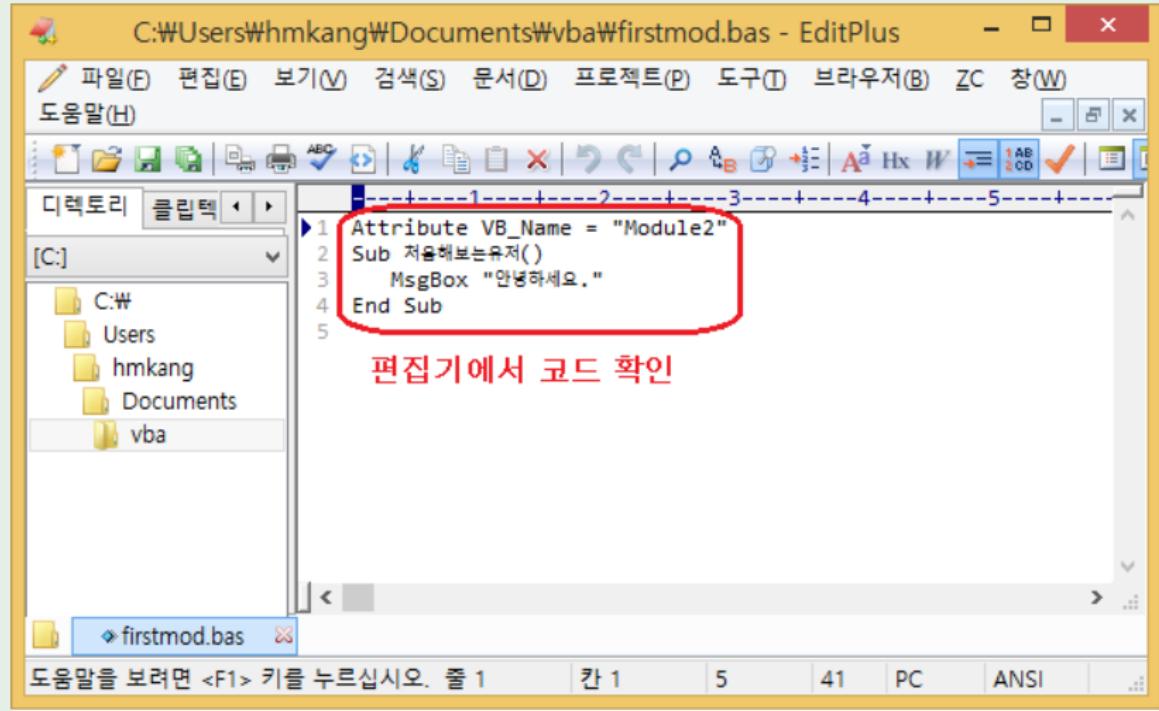
Visual Basic Editor 사용하기

저장한 파일 내용 확인



Visual Basic Editor 사용하기

편집기로 저장한 모듈 내용 확인



The screenshot shows the EditPlus text editor interface. The title bar reads "C:\Users\hmkang\Documents\vba\firstmod.bas - EditPlus". The menu bar includes 파일(F), 편집(E), 보기(V), 검색(S), 문서(D), 프로젝트(P), 도구(I), 브라우저(B), ZC, 창(W), 도움말(H). The toolbar contains various icons for file operations like Open, Save, Print, and Cut/Paste. The left sidebar shows a file tree with the path [C:] \ vba \ firstmod.bas selected. The main code editor window displays the following VBA code:

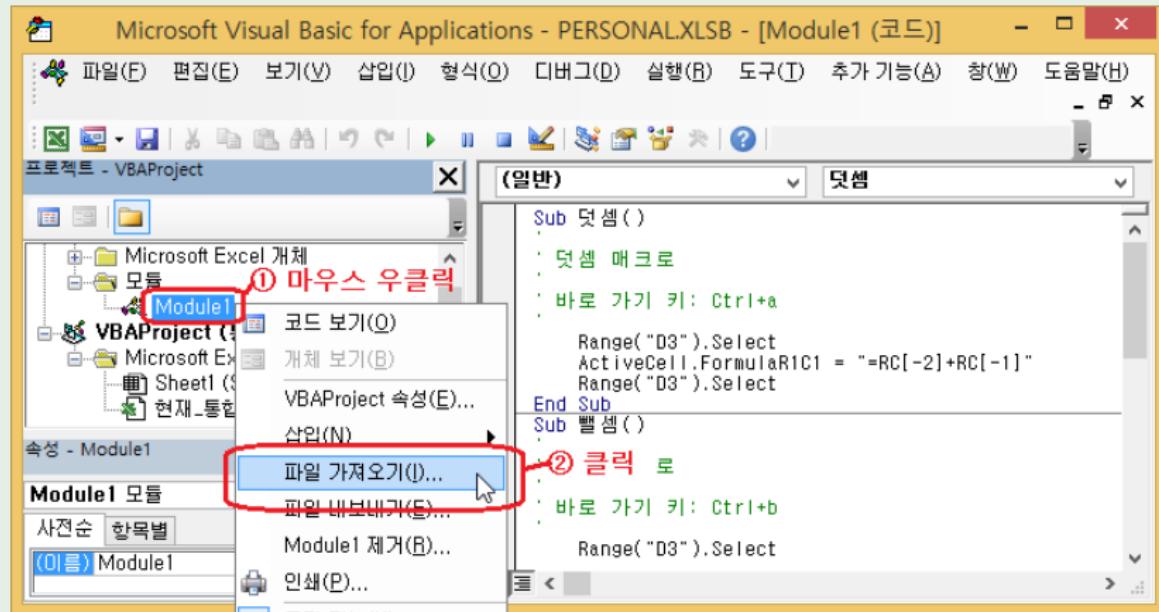
```
Attribute VB_Name = "Module2"
Sub 처음해보는유저()
    MsgBox "안녕하세요."
End Sub
```

A red oval highlights the first two lines of the code. Below the code editor, the status bar says "편집기에서 코드 확인".

도움말을 보려면 <F1> 키를 누르십시오. 줄 1 | 칸 1 | 5 | 41 | PC | ANSI

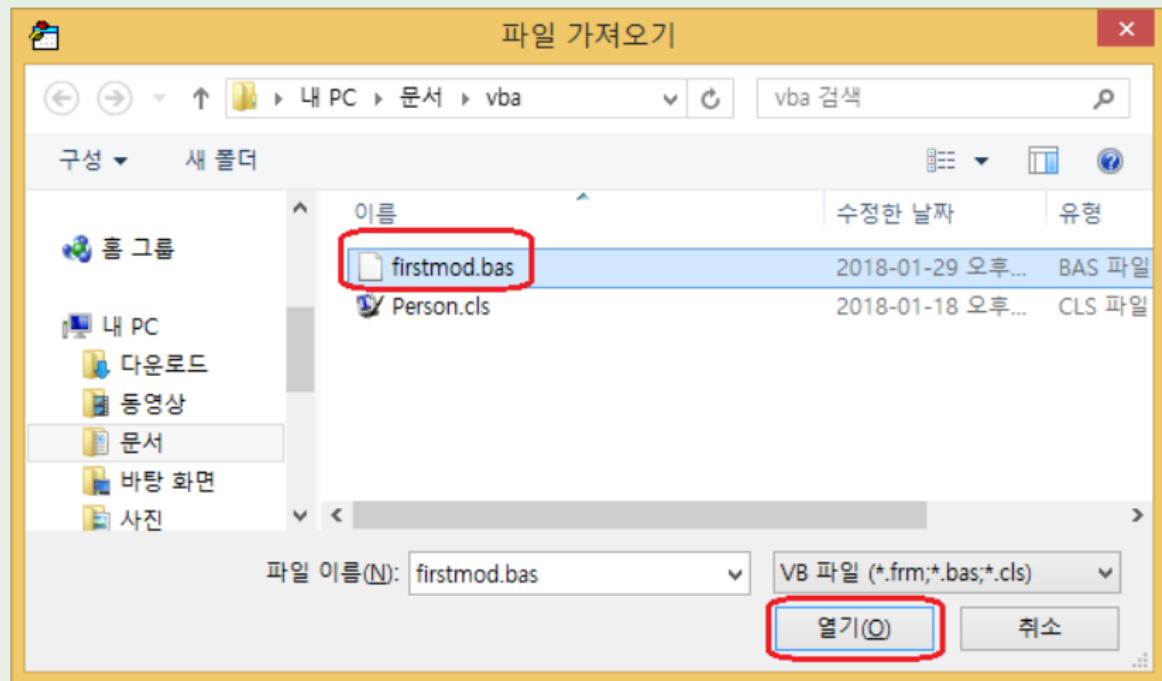
Visual Basic Editor 사용하기

저장한 VB 파일 가져오기



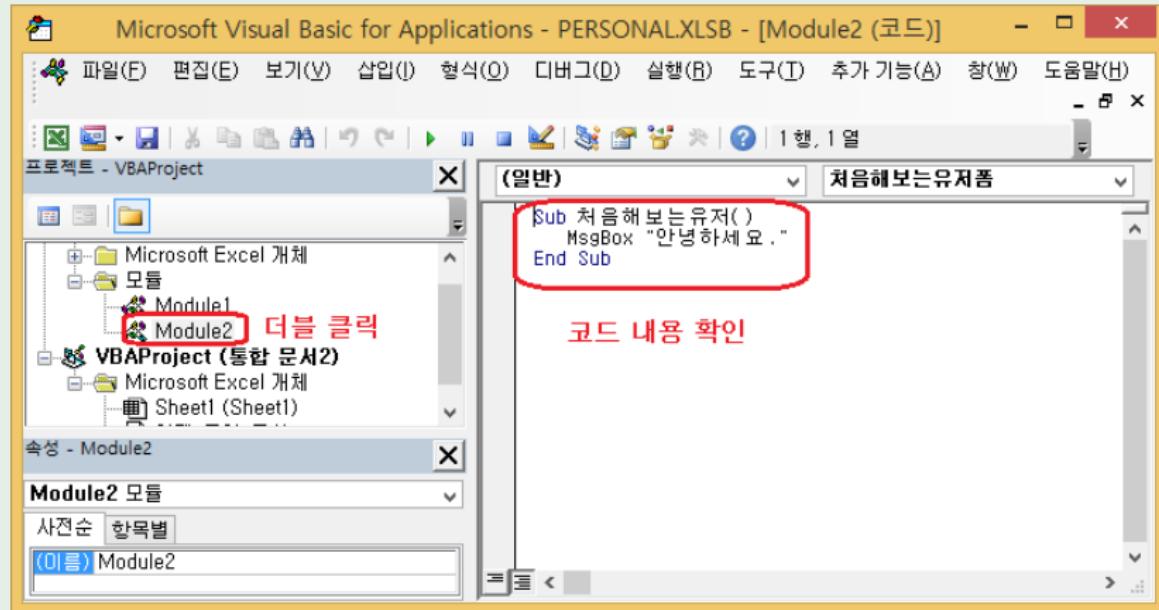
Visual Basic Editor 사용하기

VB 파일 선택



Visual Basic Editor 사용하기

가져온 모듈 확인



Visual Basic 사용방법

VBA에서 자주 사용하는 함수

MsgBox 함수

MsgBox(*prompt*[, *buttons*] [, *title*] [, *helpfile*, *context*])

MsgBox 문 인수 설명

- *prompt* : 반드시 입력하는 필수 요소로 대화 상자에서 메시지로 나타나는 문자열이다.
- *buttons* : 표시할 버튼의 개수와 형식을 지정하는 것으로 기본값은 0이며 선택 요소이다.
- *title* : 제목표시줄에 표현하는 문자열로 선택 요소이다.
- *helpfile* : 도움말이 대화 상자에 제공하며 도움말 파일을 문자열에 입력한다. 이 옵션이 사용되면 *context*도 반드시 사용해야 한다.
- *context* : 도움말이 작성된 것을 도움말 항목에 번호로 나타내는 것이다. 이 옵션이 사용되면 *helpfile*도 반드시 사용해야 한다.
- []는 생략가능

VBA에서 자주 사용하는 함수

사용 예 : MsgBox 함수

```
1 MsgBox "안녕하세요!", 1, "인사"  
2 MsgBox Title:="인사", Prompt:="안녕하세요!"
```

1 줄에서 순서에 맞게 인수 입력은

- 첫 번째 *prompt* 인수 값은 “안녕하세요”이고 반드시 입력
- 두 번째 *buttons* 인수 값은 1이고, 버튼 종류는 확인 버튼
- 세 번째 *title* 인수 값은 제목표시줄에 나타날 문자열 “인사”
- 만일 인수에 값을 입력하지 않으려면 „와 같이 아무값도 넣지 않는다.

2 줄에서

- 이름별 인수 입력은 인수 이름, 콜론, 등호(:=), 인수 값을 차례로 입력한다.

VBA에서 자주 사용하는 함수

InputBox 함수

`InputBox(prompt[, title] [, default] [, xpos] [, ypos] [, helpfile, context])`

InputBox 문 인수 설명

- *prompt* : 반드시 입력하는 요소로 대화 상자에서 보여주는 메시지
- *title* : 제목표시줄 문자열로 선택 요소
- *default* : 입력란에 보여줄 기본 입력값
- *xpos* : 화면 왼쪽에서 대화상자까지 간격으로 생략하면 기본값은 수평 중앙
- *ypos* : 화면 위에서 대화상자까지 간격으로 생략하면 대화상자는 1/3
- *helpfile* : 대화상자에 제공하는 도움말이이다. 파일은 문자열에 입력 이 옵션은 *context*와 함께 사용한다.
- *context* : 작성한 도움말 내용이 도움말 항목에 나타날 번호. 이 옵션은 *helpfile*과 함께 사용한다.

VBA에서 자주 사용하는 함수

사용 예 : InputBox 함수

```
1 Sub Question()
2     myVar = InputBox(Prompt:="숫자를 입력하세요.", _
3         Default:="100", Title:="홀수, 짝수 판별")
4     MsgBox myVar
5 End Sub
```

인수 입력은

- Prompt 입력 상자의 메시지는 숫자를 입력하세요
- Default 입력란 기본 값은 100
- Title 제목표시줄은 홀수, 짝수 판별

Visual Basic 구성

- 프로젝트 : 모듈, 클래스 모듈, 사용자 정의 폼, Microsoft Excel 개체를 포함하는 최상위 항목이다.
- 모듈 : 매크로를 만들고 실행하는 곳이다.
- 클래스 모듈 : 개체 제공을 목적으로 작성한 모듈이다. 모듈에서 이것들을 참조하여 사용한다.
- 사용자 정의 폼 : 창이나 대화상자에 버튼이나 리스트와 같은 개체를 추가하는 컬렉션이다. 버튼 클릭이나 리스트 선택을 이벤트가 발생했다고 한다. 추가한 개체에 이벤트가 발생하면 실행할 코드를 삽입한다.

프로시저 종류

- Sub : 프로그램 작성 단위로 작성한 프로그램을 실행하고 Sub가 종료되면서 되돌려 주는 값이 없다. 즉 리턴(return) 값이 없다.
- Function : 프로그램 작성 단위로 Function이 종료되고 되돌려 주는 값(return)이 있다.
- Property : 폼, 모듈, 클래스 모듈에 속성 값을 변경하거나 저장한 속성 값을 가져온다.

제어문

If-Then-ElseIf-Then-Else 문

```
If condition1 Then  
    [statements1]  
[Elseif condition2 Then  
    [statements2] ]  
[Else  
    [statementsn] ]  
End If
```

If-Then-ElseIf-Then-Else 문 작동과정

- $condition_1$: 이 조건이 True이면 $statements_1$ 이 실행된다.
- $condition_2$: $condition_1$ 을 제외하고 이 조건이 True이면 $statements_2$ 실행된다.
- []는 사용하지 않아도 됨

제어문

사용 예 : If-Then-ElseIf-Then-Else 문

```
1 Sub IfElseIfTest()
2     score = InputBox("점수를 입력하세요", "점수", 89)
3     If score >= 90 Then
4         MsgBox "당신의 성적등급은 매우 잘함입니다."
5     ElseIf score >= 80 Then
6         MsgBox "당신은 성적등급은 잘함입니다."
7     ElseIf score >= 70 Then
8         MsgBox "당신은 성적등급은 보통입니다."
9     ElseIf score >= 60 Then
10        MsgBox "당신은 성적등급은 못함입니다."
11    Else
12        MsgBox "당신은 성적등급은 매우 못함입니다."
13    End If
14 End Sub
```

제어문

Select-Case 문

Select Case *testexpression*

Case *expressionlist*

[*statements*]

:

Case Else

[*statements*]

End Select

Select-Case 문 작동과정

- *testexpression* : 반드시 입력해야 하는 값으로 수식이나 문자식
- *testexpressionlist* : Case에 반드시 입력하는 값으로 특정값이나 범위를 지정할 수 있음
- []는 사용하지 않아도 됨

제어문

사용 예 : Select-Case 문

```
1 Sub SelectCaseTest()
2     score = InputBox("점수를 입력하세요", "점수", 89)
3     Select Case score
4         Case score >= 90
5             MsgBox "당신의 성적등급은 매우 잘함입니다."
6         Case score >= 80
7             MsgBox "당신은 성적등급은 잘함입니다."
8         Case score >= 70
9             MsgBox "당신은 성적등급은 보통입니다."
10        Case score >= 60
11            MsgBox "당신은 성적등급은 못함입니다."
12        Case Else
13            MsgBox "당신은 성적등급은 매우 못함입니다."
14    End Select
15 End Sub
```

반복문

Do-Loop 문

Do [{While | Until} *condition*]

[*statements*]

[Exit Do]

Loop

Do-Loop 문

Do

[*statements*]

[Exit Do]

Loop [{While | Until} *condition*]

Do-Loop 문 작동과정

- While은 *condition*이 True이면 Until은 False이면 *statements*을 실행
- 처음 Do-Loop 문은 *condition*에 따라 실행하지 않을 수도 있고 나중 Do-Loop 문은 반드시 한 번 실행함

반복문

사용 예 : Do-Loop 문

```
1 Sub ChkFirstWhile()
2     counter = 0
3     myNum = 20
4     Do While myNum > 10
5         myNum = myNum - 1
6         counter = counter + 1
7     Loop
8     MsgBox "루프를 "& counter &"번 반복하였다."
9 End Sub
10 Sub ChkLastWhile()
11     counter = 0
12     myNum = 9
13     Do
14         myNum = myNum - 1
15         counter = counter + 1
16     Loop While myNum > 10
17     MsgBox "루프를 "& counter &"번 반복하였다."
18 End Sub
```

반복문

For-Each-Next 문

For Each *element* In *group*

[*statements*]

[Exit For]

Next *element*

For-Each-Next 문 작동과정

- *element* : 컬렉션이나 배열의 요소들을 반복실행에 사용하는 변수
- *group* : 개체 컬렉션이나 배열이름
- *group* 개체로 선언된 *element*는 반복하여 모두 실행

반복문

사용 예 : For-Each-Next 문

```
1 Sub ForEachTest()
2     Dim rng As Range
3     Dim cel As Range
4     Set rng = Application.Range("Sheet1!B2:C3")
5     For Each cel In rng.Cells
6         MsgBox cel.value
7     Next cel
8 End Sub
```

반복문

For-Next 문

For *counter*=*start* To *end* [Step *step*]
[*statements*]

[Exit For]

Next *counter*

For-Next 문 작동과정

- *counter* : 반복문에 루프 카운터로 사용할 숫자 변수명
- *start* : 반복문에서 시작값
- *end* : 반복문에서 종료값
- *step* : 반복문에서 루프에서 변화하는 값으로 생략하면 1
- 설정한 횟수만큼 문 블럭을 반복

반복문

사용 예 : For-Next 문

```
1 Sub Speak()
2     Dim intro(3) As String
3     intro(1) = "안녕하세요"
4     intro(2) = InputBox("이름입력")
5     intro(3) = "님 반갑습니다."
6     For x = 1 To 3
7         Application.Speech.Speak intro(x)
8     Next x
9 End Sub
```

반복문

With 문

With *object*

[*statements*]

End With

With 문 작동과정

- *object*는 개체의 이름이나 사용자 정의 형식
- 연속되는 문에서 개체나 사용자 정의 형식을 한 번만 지정하여 사용

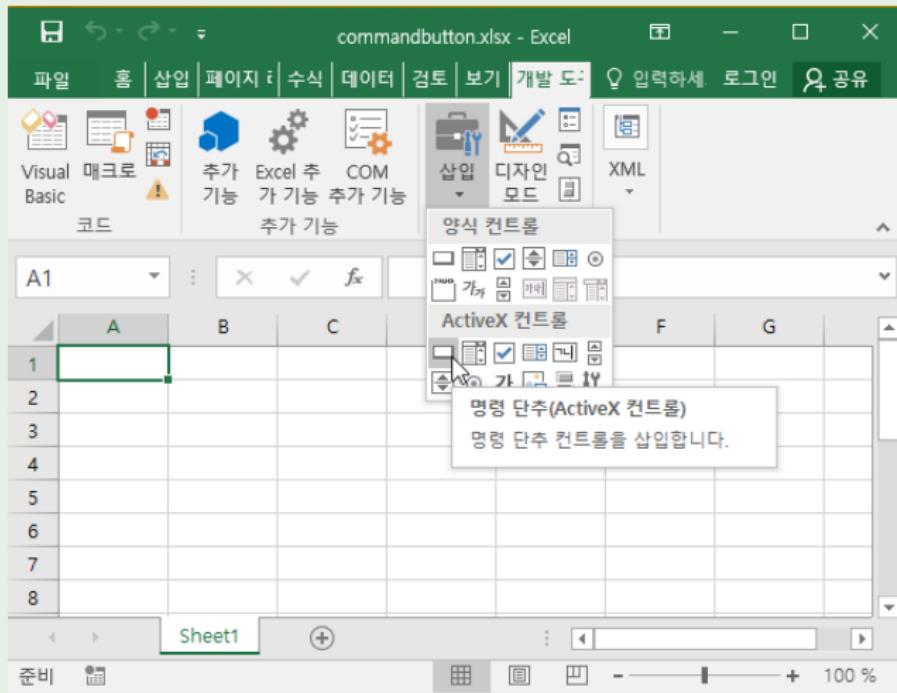
사용 예 : With 문

```
1 Sub FormatRange()
2     With Worksheets("Sheet1").Range("A1:C10")
3         .Value = 30
4         .Font.Bold = True
5         .Interior.Color = RGB(255, 255, 0)
6     End With
7 End Sub
```

ActiveX 컨트롤 엑셀 연결

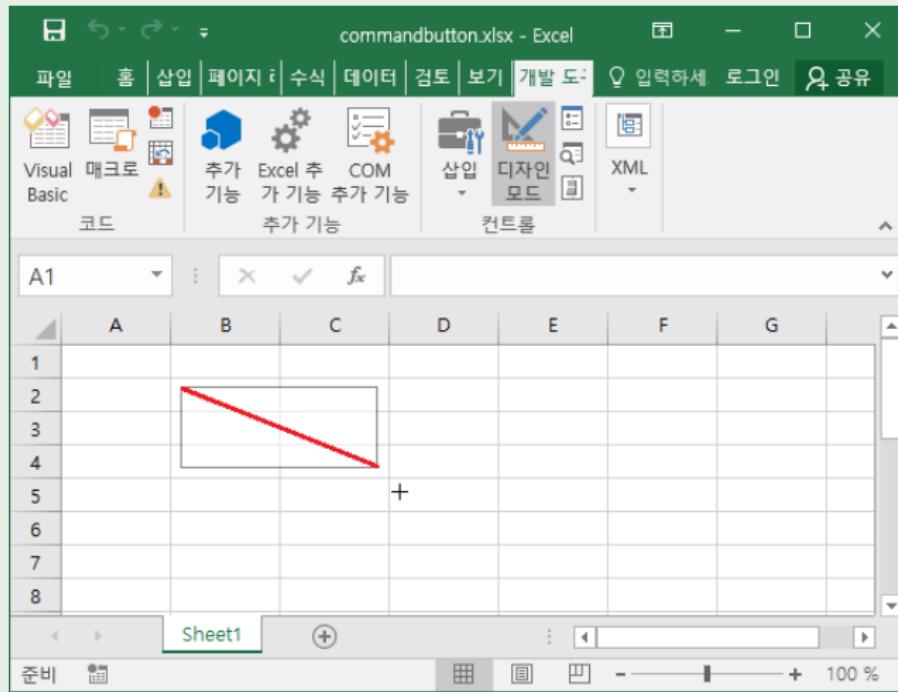
ActiveX 컨트롤 엑셀 연결 – 명령 단추

ActiveX 컨트롤에서 명령 단추 클릭



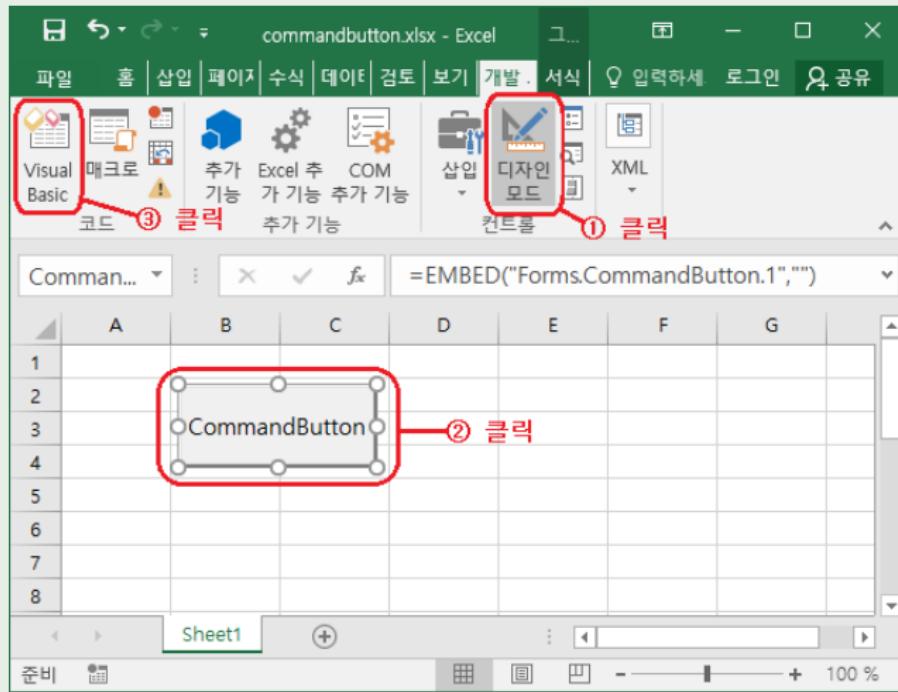
ActiveX 컨트롤 엑셀 연결 – 명령 단추

명령 단추를 시트에 마우스 드래그로 추가



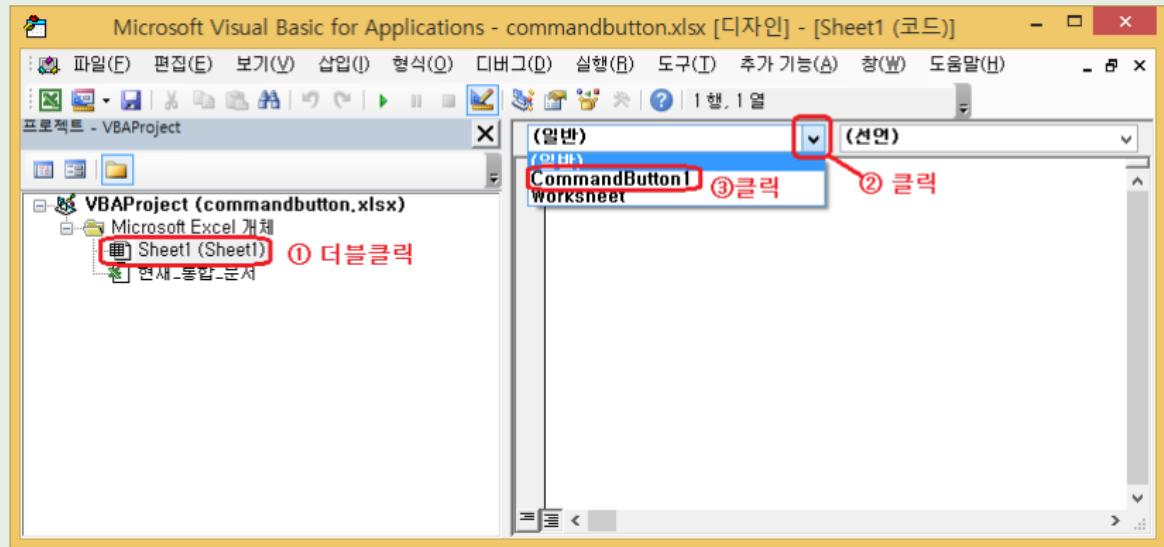
ActiveX 컨트롤 엑셀 연결 – 명령 단추

추가한 명령 단추는 자동으로 디자인 모드가 됨



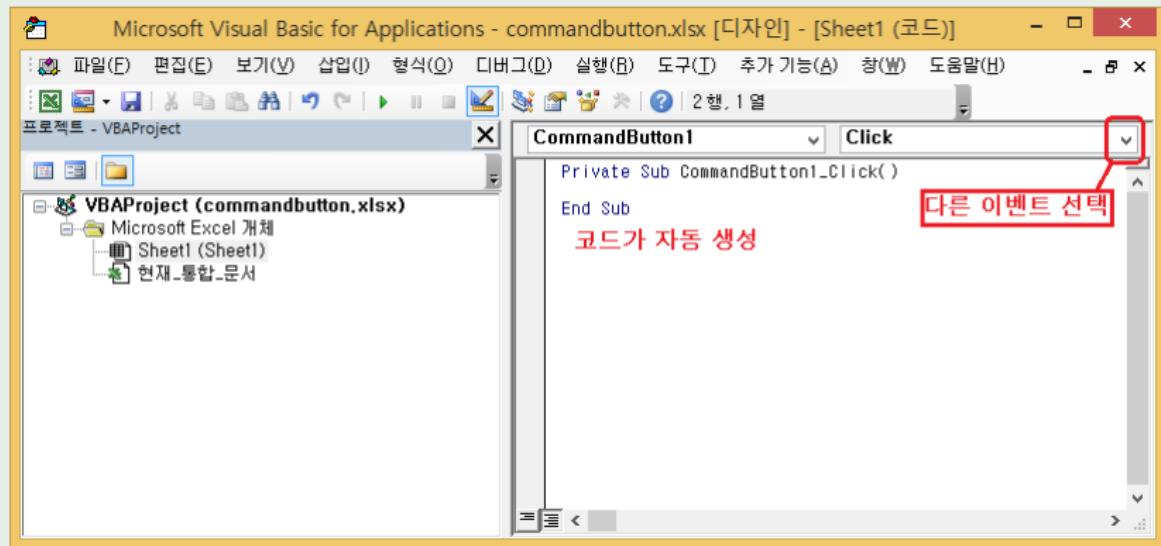
ActiveX 컨트롤 엑셀 연결 – 명령 단추

VBE에서 시트 더블클릭, 개체 선택에서 CommandButton1 선택



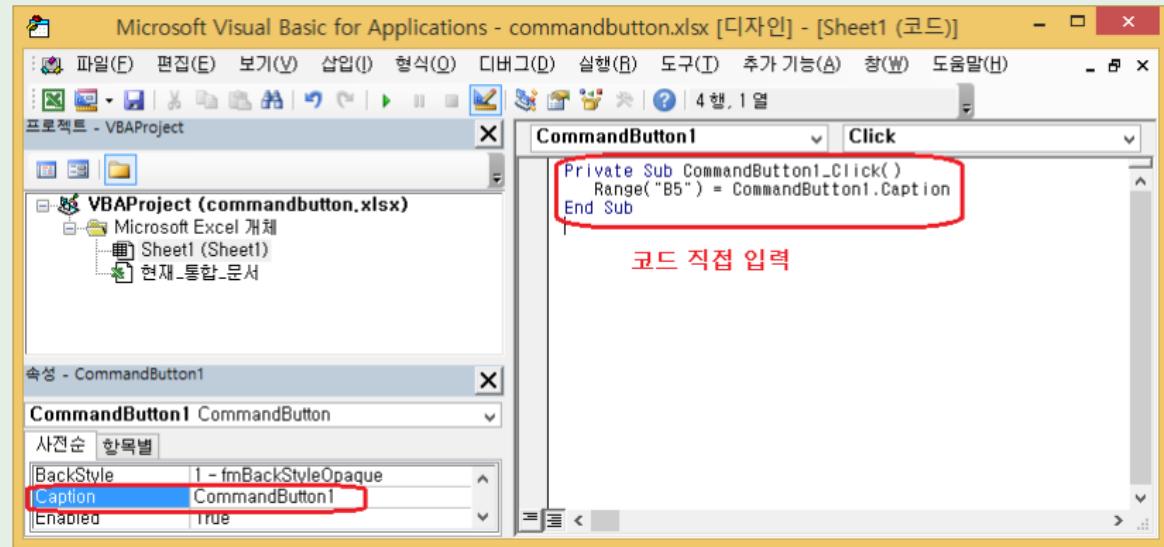
ActiveX 컨트롤 엑셀 연결 – 명령 단추

개체 이벤트가 자동 추가, 다른 이벤트 변경 가능



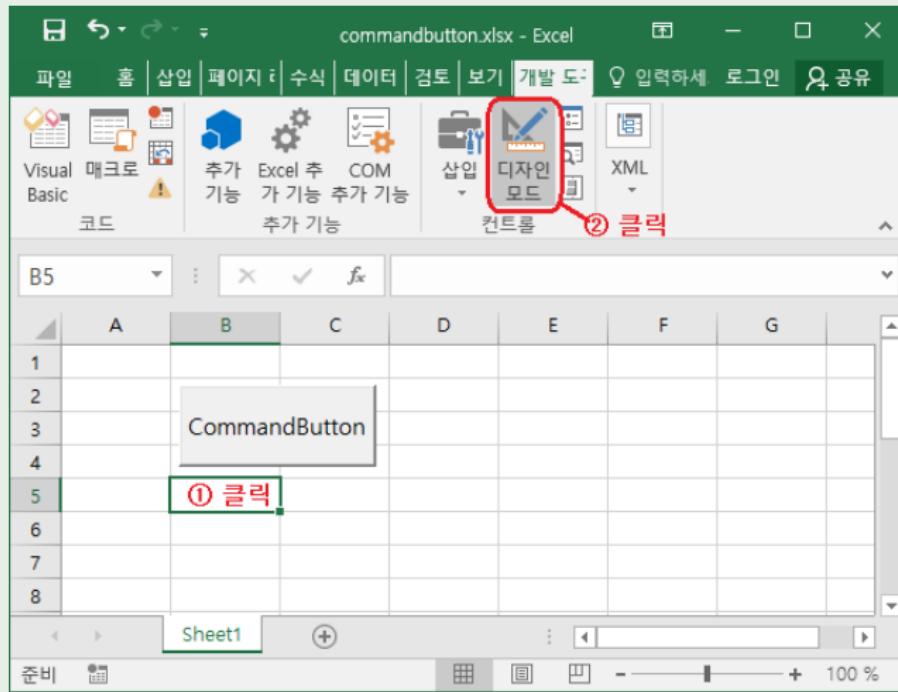
ActiveX 컨트롤 엑셀 연결 – 명령 단추

선택한 이벤트 프로시저에 직접 코드 입력



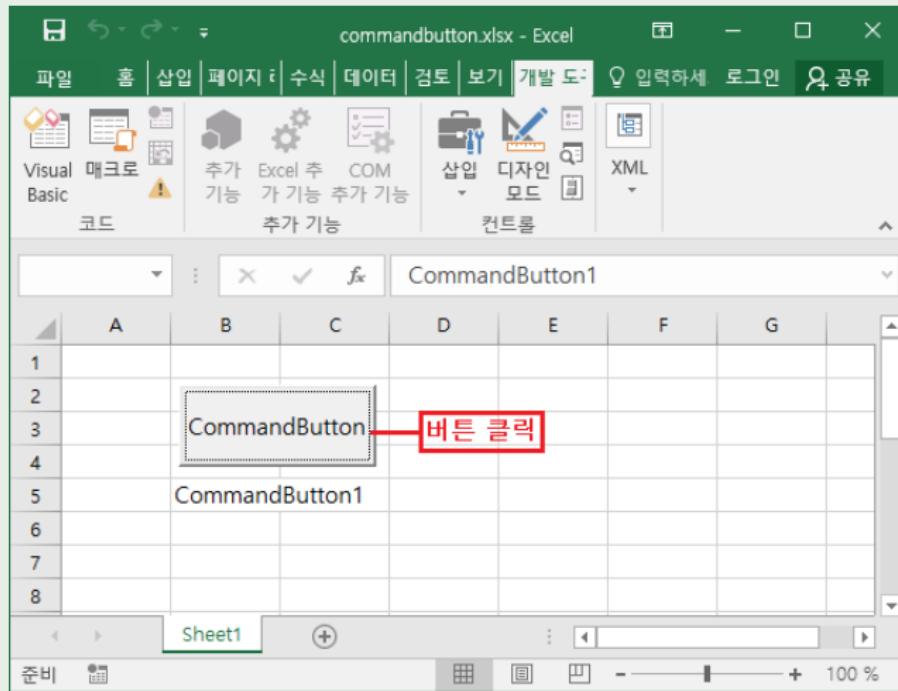
ActiveX 컨트롤 엑셀 연결 – 명령 단추

셀 클릭하고 디자인 모드 해제



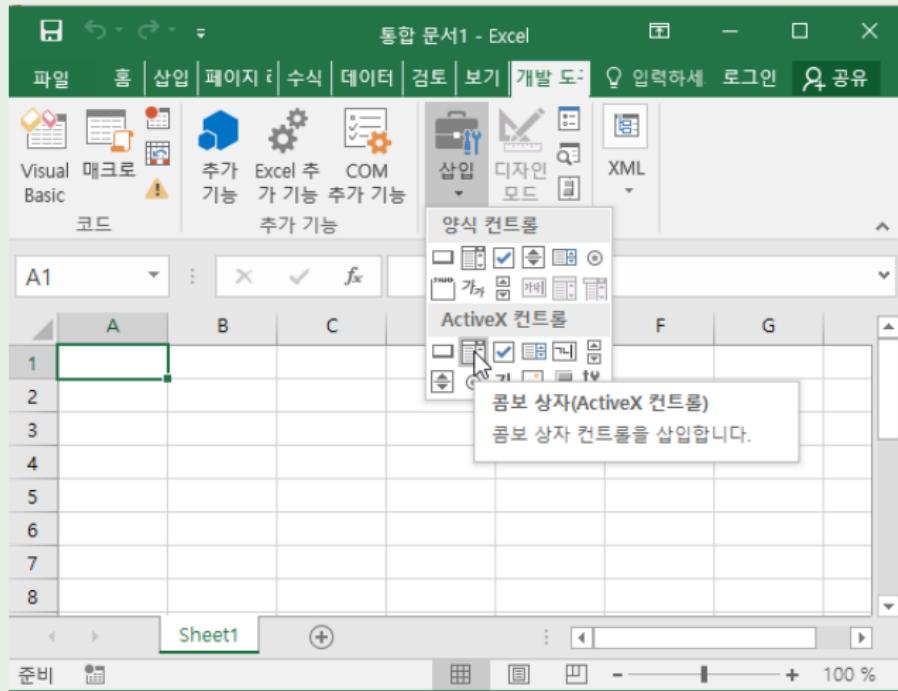
ActiveX 컨트롤 엑셀 연결 – 명령 단추

추가한 명령 단추 클릭



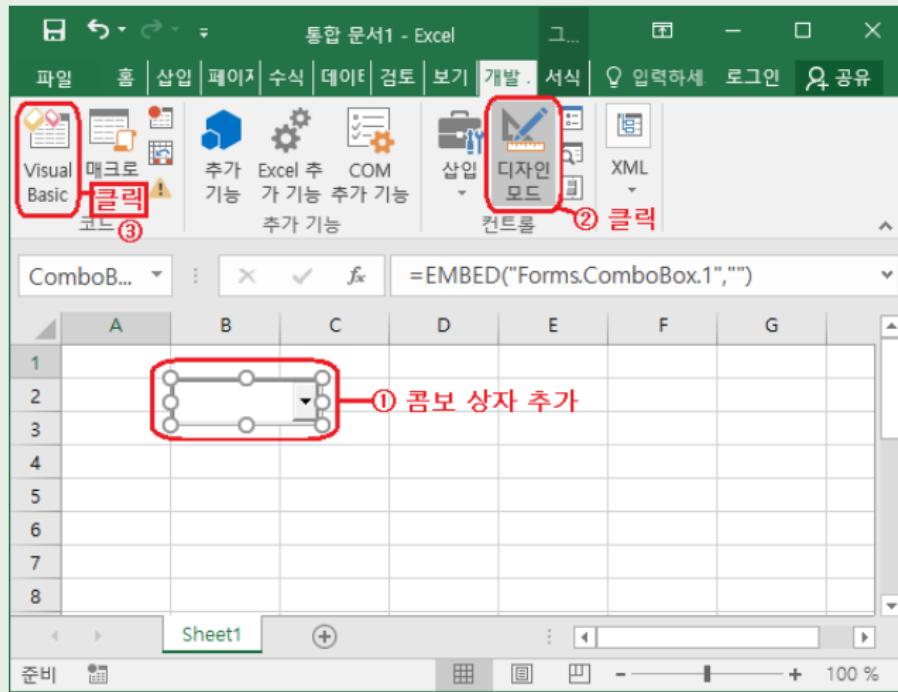
ActiveX 컨트롤 엑셀 연결 – 콤보 상자

추가할 콤보 상자 클릭



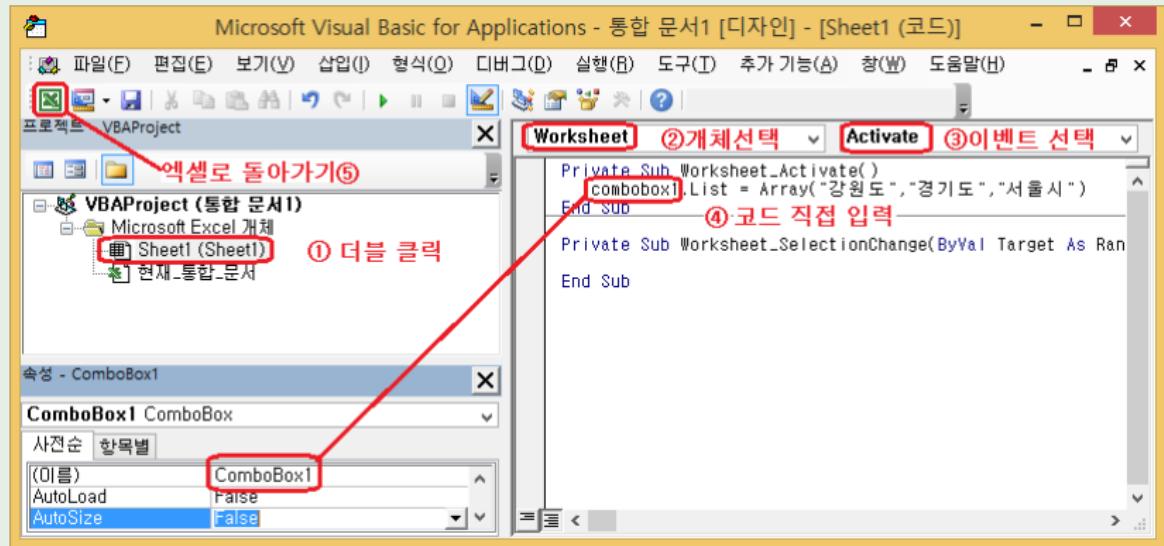
ActiveX 컨트롤 엑셀 연결 – 콤보 상자

콤보 상자 추가하고 Visual Basic 클릭



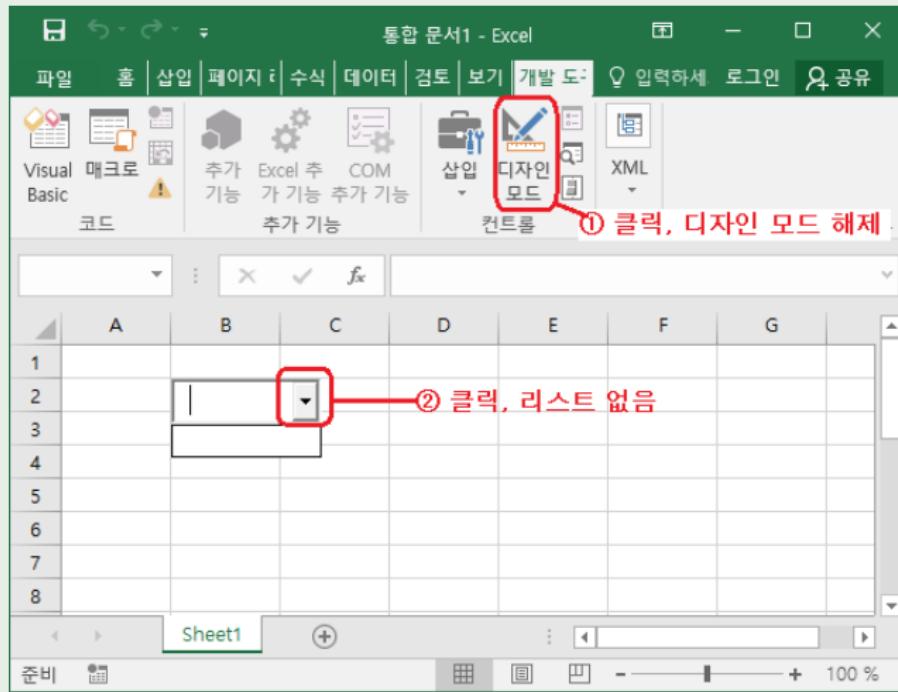
ActiveX 컨트롤 엑셀 연결 – 콤보 상자

Worksheet 개체 Activate 프로시저 선택하여 콤보 상자 목록 생성



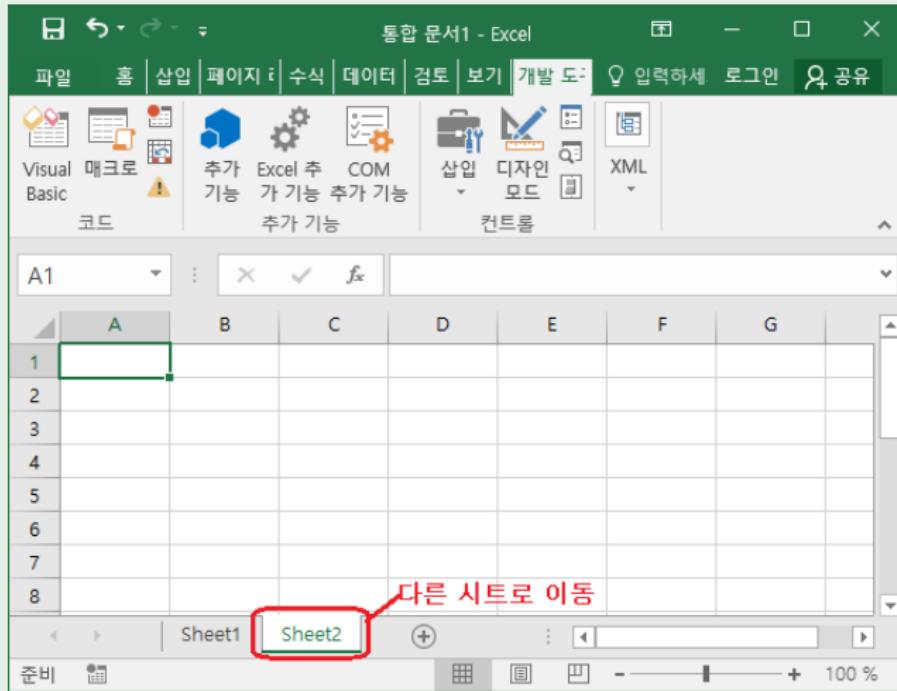
ActiveX 컨트롤 엑셀 연결 – 콤보 상자

디자인 모드 해제, 추가된 목록 없음



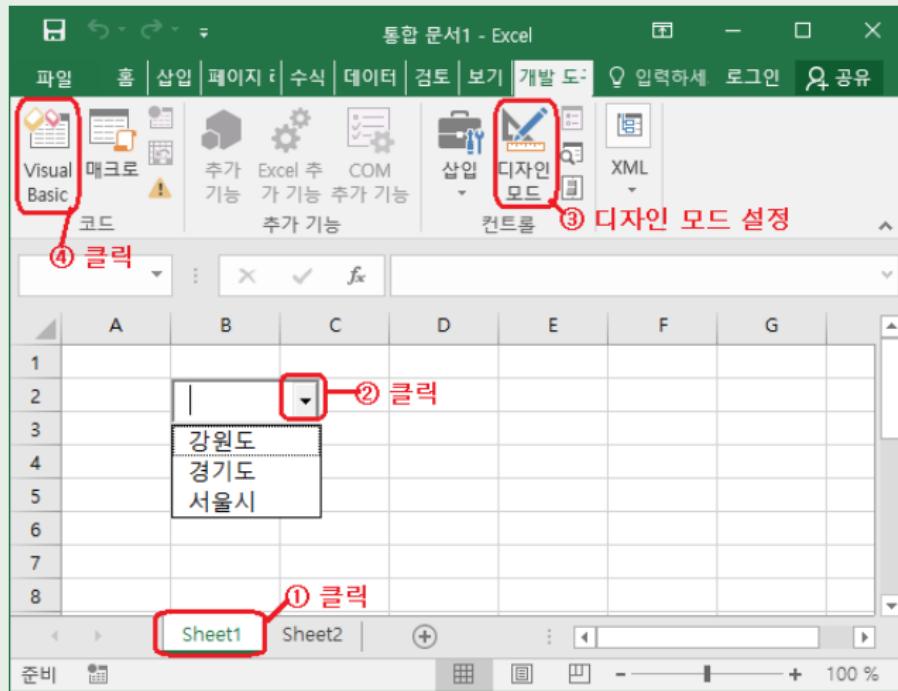
ActiveX 컨트롤 엑셀 연결 – 콤보 상자

시트 추가하고 다른 시트로 이동



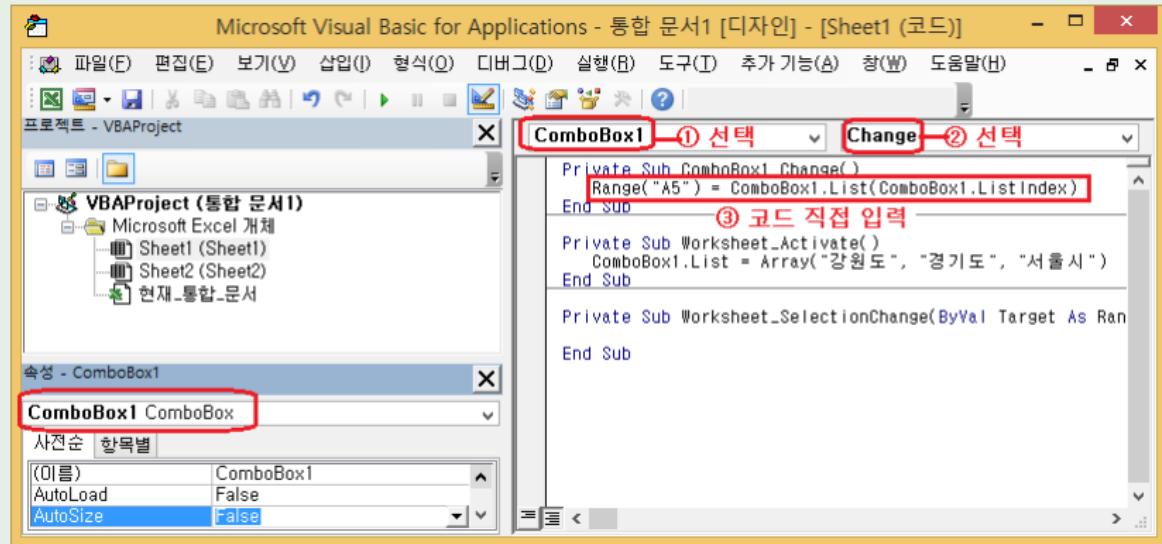
ActiveX 컨트롤 엑셀 연결 – 콤보 상자

콤보 상자가 있는 시트로 이동하면 목록이 나타남



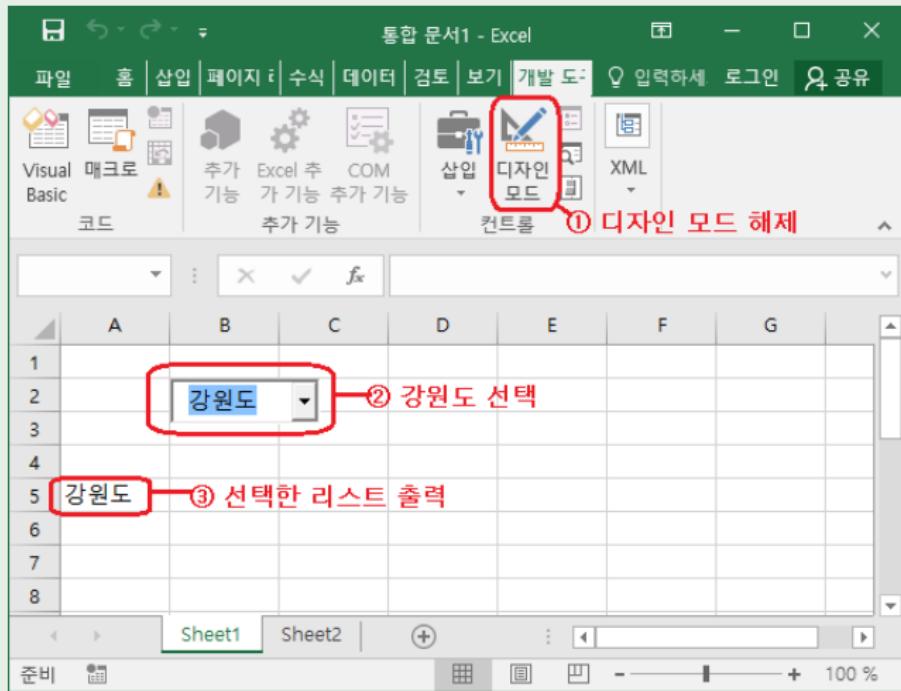
ActiveX 컨트롤 엑셀 연결 – 콤보 상자

콤보 상자 개체, Change 프로시저 선택하고 코드 직접 입력



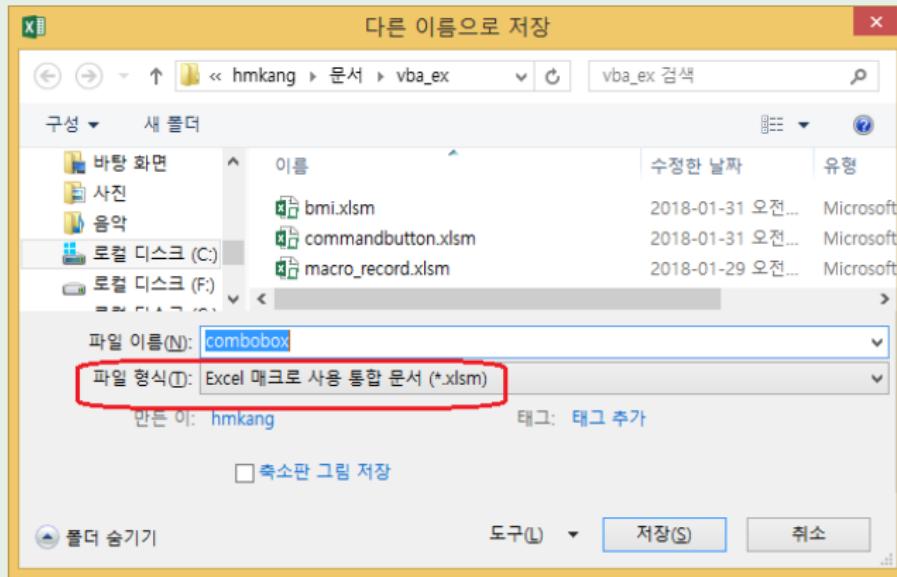
ActiveX 컨트롤 엑셀 연결 – 콤보 상자

콤보 상자 목록을 클릭하면 셀에 목록이 출력



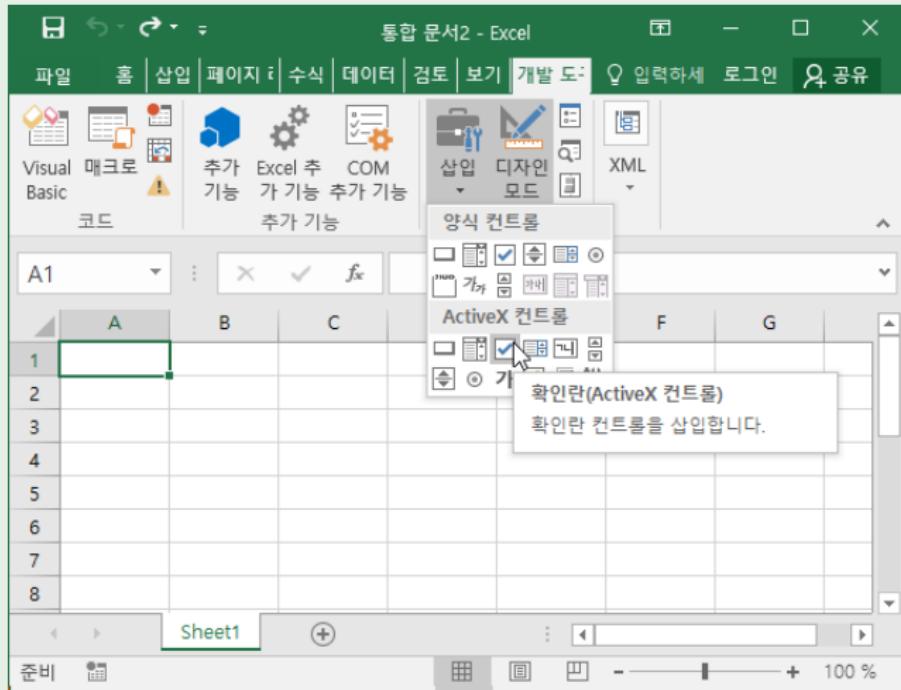
ActiveX 컨트롤 엑셀 연결 – 콤보 상자

Excel 매크로 사용 통합 문서 (*.xlsm)으로 저장



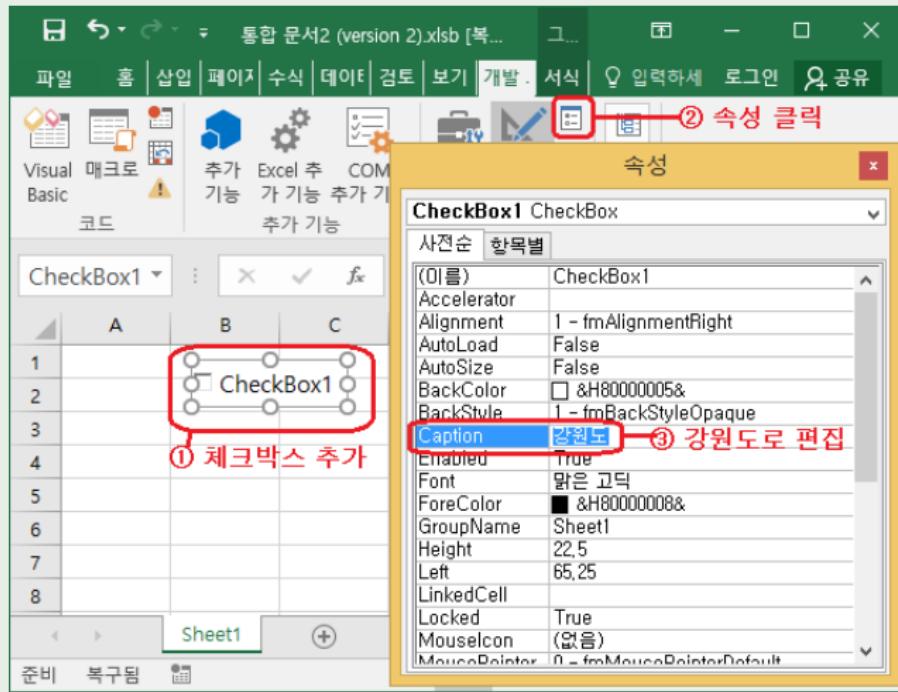
ActiveX 컨트롤 엑셀 연결 – 확인란

추가할 확인란 클릭



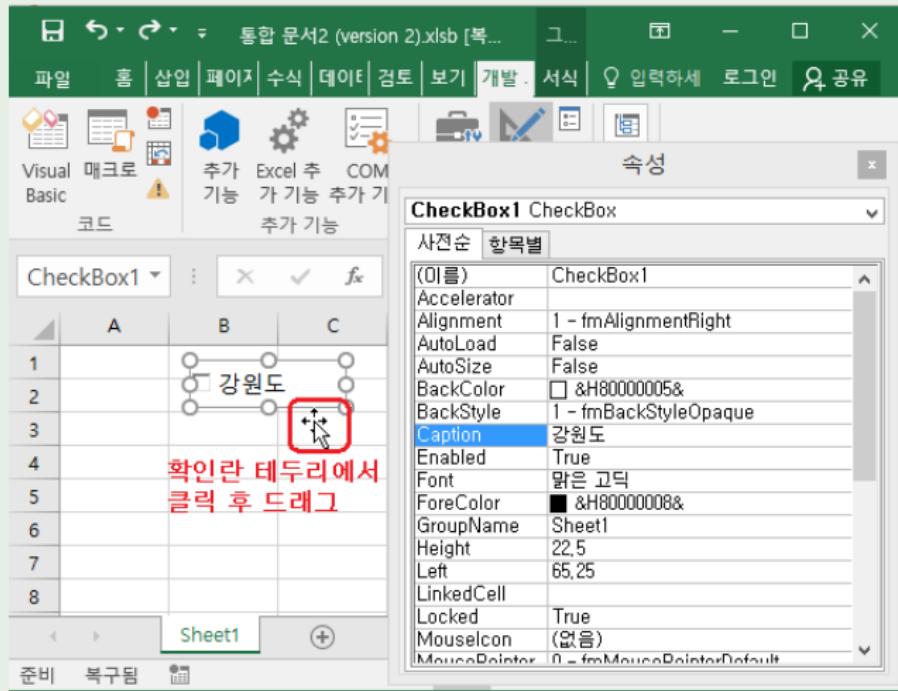
ActiveX 컨트롤 엑셀 연결 – 확인란

확인란 추가, Caption 속성 값 변경



ActiveX 컨트롤 엑셀 연결 – 확인란

확인란 복사는 Ctrl + 확인란 테두리에서 마우스 클릭후 드래그



ActiveX 컨트롤 엑셀 연결 – 확인란

확인란 2개 복사

The screenshot shows a Microsoft Excel window with the ribbon menu at the top. The ribbon tabs include 파일 (File), 홈 (Home), 삽입 (Insert), 페이지 (Page Layout), 수식 (Formulas), 데이터 (Data), 검토 (Review), 보기 (View), 개발 (Developer), 서식 (Format), 입력하세요 (Enter), 로그인 (Log In), and 공유 (Share). The Developer tab is selected.

In the center of the screen, there is a worksheet titled "Sheet1". The worksheet contains three rows of data. Row 1 has columns A, B, and C. Row 2 has a checkbox in column B labeled "□ 강원도". Row 3 has a checkbox in column B labeled "□ 강원도". Row 4 has a checkbox in column B labeled "□ 강원도". Row 5 has a checkbox in column B labeled "□ 강원도" and contains three circular ActiveX control handles (two vertical, one horizontal) indicating it is selected.

A context menu is open over the third checkbox in row 3. The menu items are: Visual Basic, 매크로 (Macro), 추가 기능 (Add-in), Excel 추가 기능 (Excel Add-in), COM 추가 기능 (COM Add-in), and 코드 (Code). The "추가 기능" item is highlighted with a yellow warning icon.

An "속성" (Properties) dialog box is open on the right side of the screen, showing the properties for "CheckBox3". The "항목별" (List by Item) tab is selected. The properties listed are:

(이름)	CheckBox3
Accelerator	1 - fmAlignmentRight
Alignment	1 - fmAlignmentRight
AutoLoad	False
AutoSize	False
BackColor	□ &H80000000&
BackStyle	1 - fmBackStyleOpaque
Caption	강원도
Enabled	True
Font	맑은 고딕
ForeColor	■ &H80000000&
GroupName	Sheet1
Height	22.5
Left	64.5
LinkedCell	
Locked	True
MouseIcon	(없음)
MousePointer	0 - fmMousePointerDefault

ActiveX 컨트롤 엑셀 연결 – 확인란

추가한 2개 확인란 Caption 속성 값 변경

통합 문서2 (version 2).xlsv [복...]

파일 | 홈 | 삽입 | 페이지 | 수식 | 데이터 | 검토 | 보기 | 개발 | 서식 | 입력하세요 | 로그인 | 공유

Visual Basic | 매크로 | 추가 기능 | Excel 추가 기능 | COM 추가 기능 | 추가 기능

코드

속성

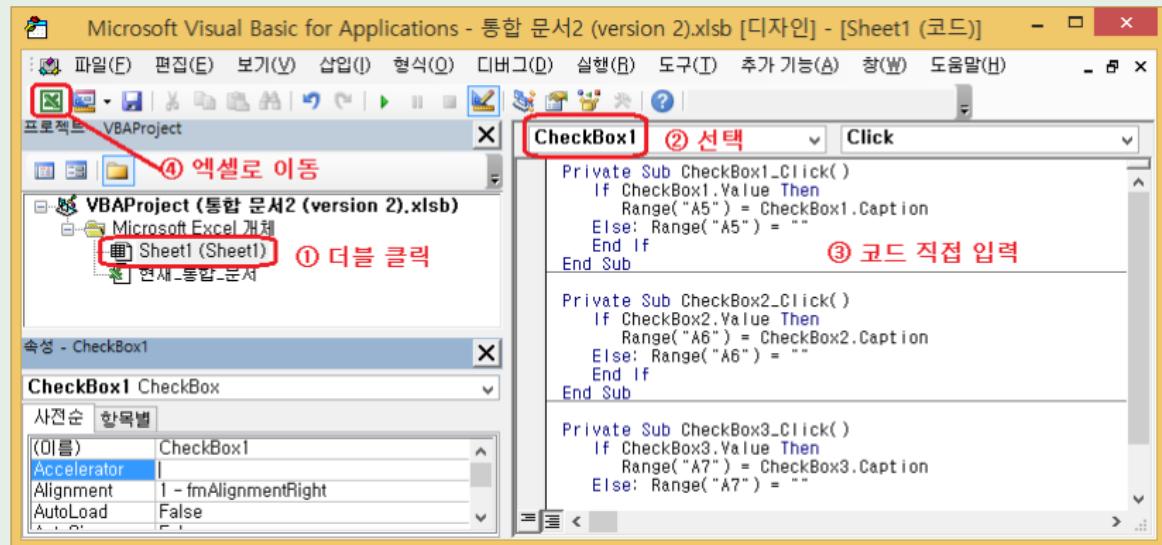
CheckBox3 CheckBox

사전순 항목별

(이름)	CheckBox3
Accelerator	1 - fmAlignmentRight
Alignment	1 - fmAlignmentRight
AutoLoad	False
AutoSize	False
BackColor	□ &H80000005&
BorderStyle	1 - fmBorderStyleOpaque
Caption	서울시 ①값 변경
Enabled	True
Font	맑은 고딕
ForeColor	■ &H80000008&
GroupName	Sheet1
Height	22.5
Left	64.5
LinkedCell	
Locked	True
MouseIcon	(없음)
MousePointer	0 - fmMousePointerDefault

ActiveX 컨트롤 엑셀 연결 – 확인란

확인란을 클릭하면 실행할 코드 직접 입력



ActiveX 컨트롤 엑셀 연결 – 확인란

디자인 모드 해제하고 추가한 확인란 선택하여 이벤트 발생

통합 문서2 (version 2).xlslb [복구됨] - Excel

파일 홈 삽입 페이지 수식 데이터 검토 보기 개발 도구 입력하세요 로그인 공유

Visual Basic 매크로 추가 기능 Excel 추가 COM 추가 기능 추가 기능

코드 캡처 디자인 모드 XML

① 디자인 모드 해제

② 확인란 이벤트 작동 확인

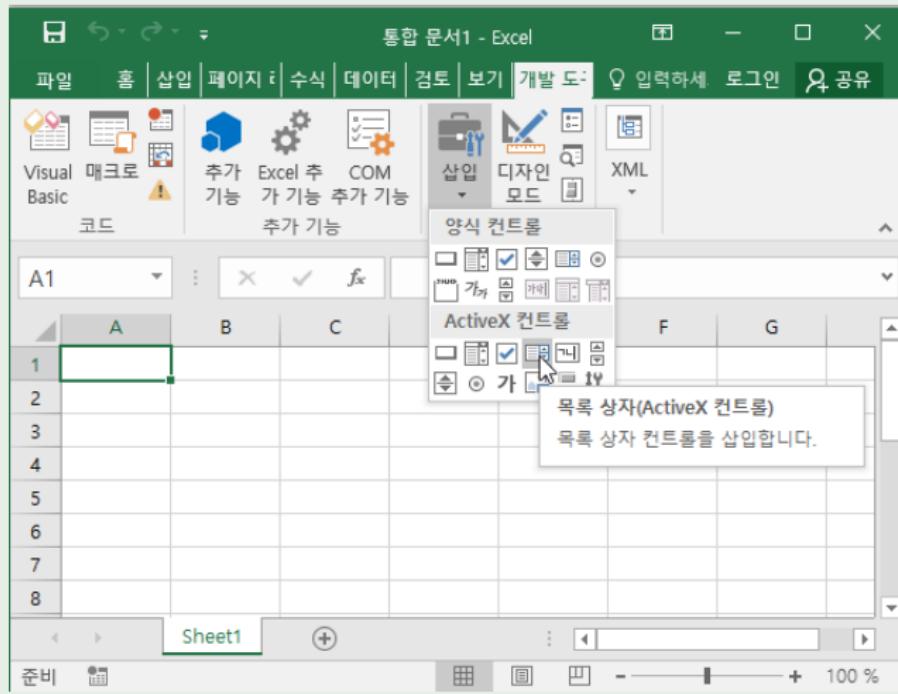
A	B
1	<input type="checkbox"/> 강원도
2	<input type="checkbox"/> 경기도
3	<input type="checkbox"/> 서울시
4	
5 강원도	
6 경기도	
7 서울시	
8	

Sheet1

준비 복구됨

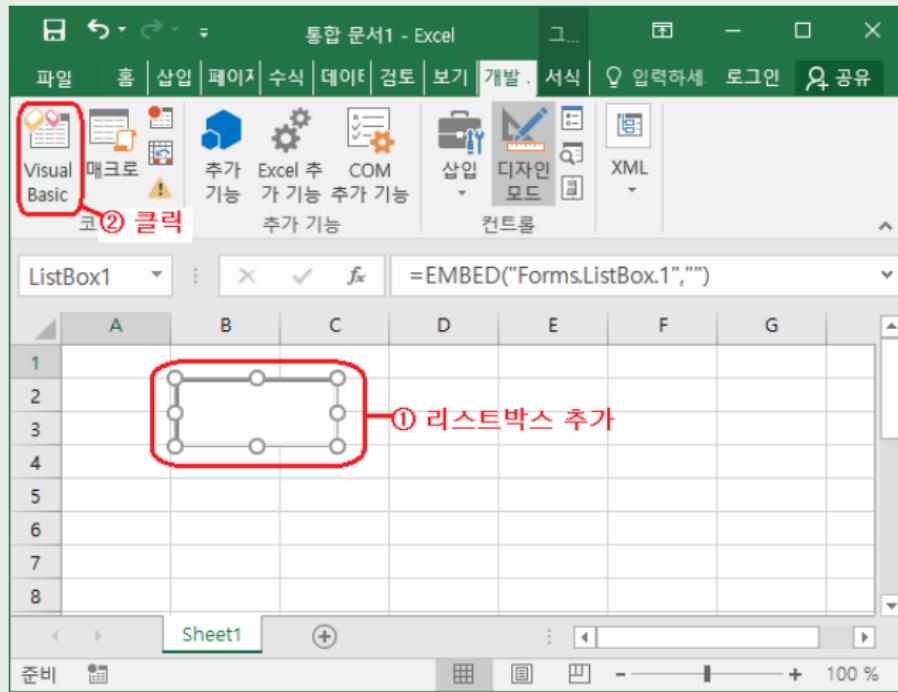
ActiveX 컨트롤 엑셀 연결 – 목록 상자

추가할 목록 상자 클릭



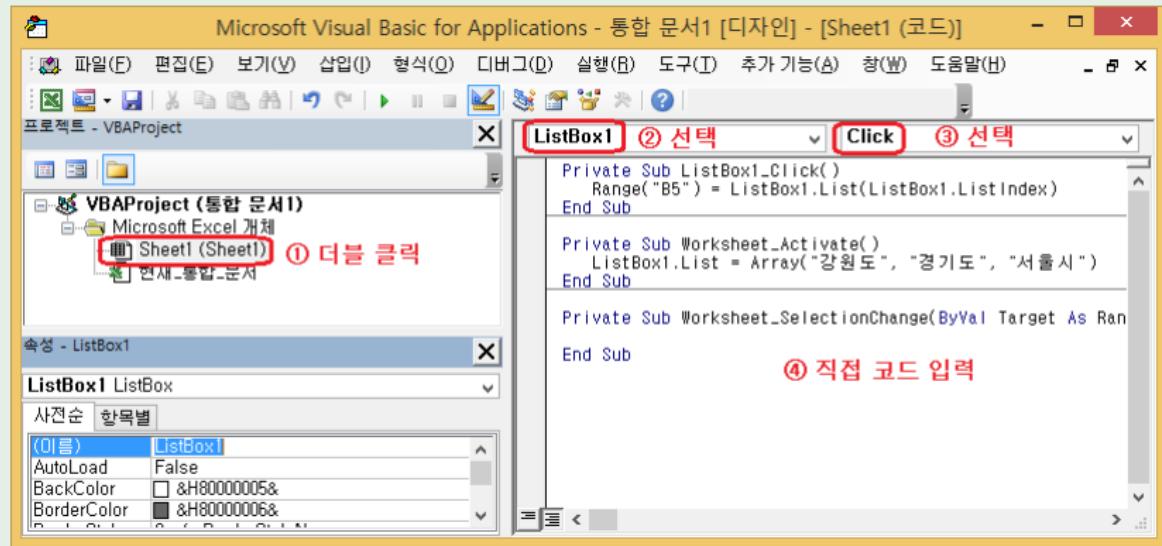
ActiveX 컨트롤 엑셀 연결 – 목록 상자

목록 상자를 시트에 추가



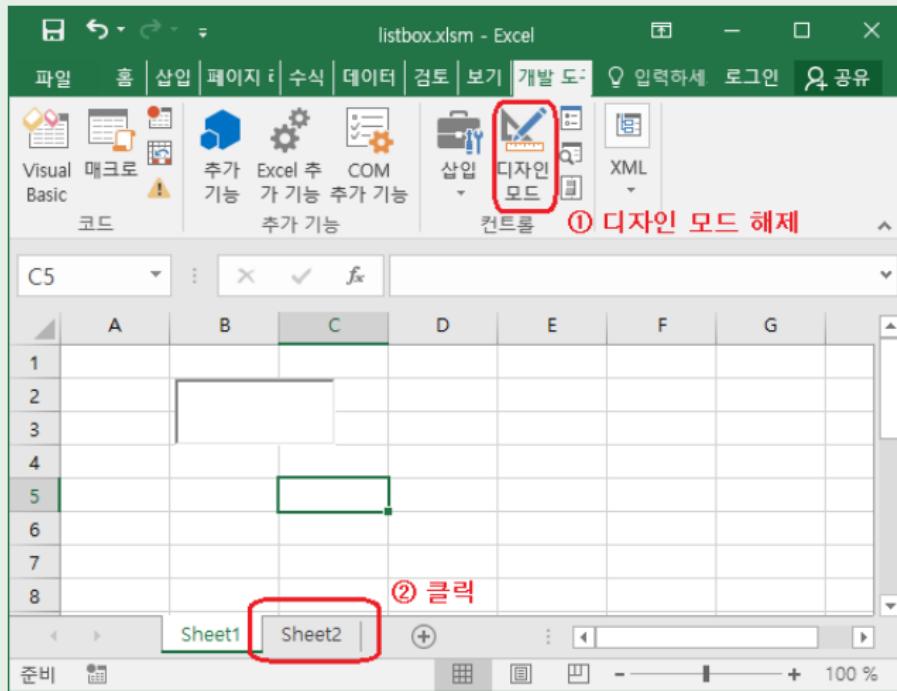
ActiveX 컨트롤 엑셀 연결 – 목록 상자

목록 상자 개체 선택, 프로시저 자동추가, 코드 입력



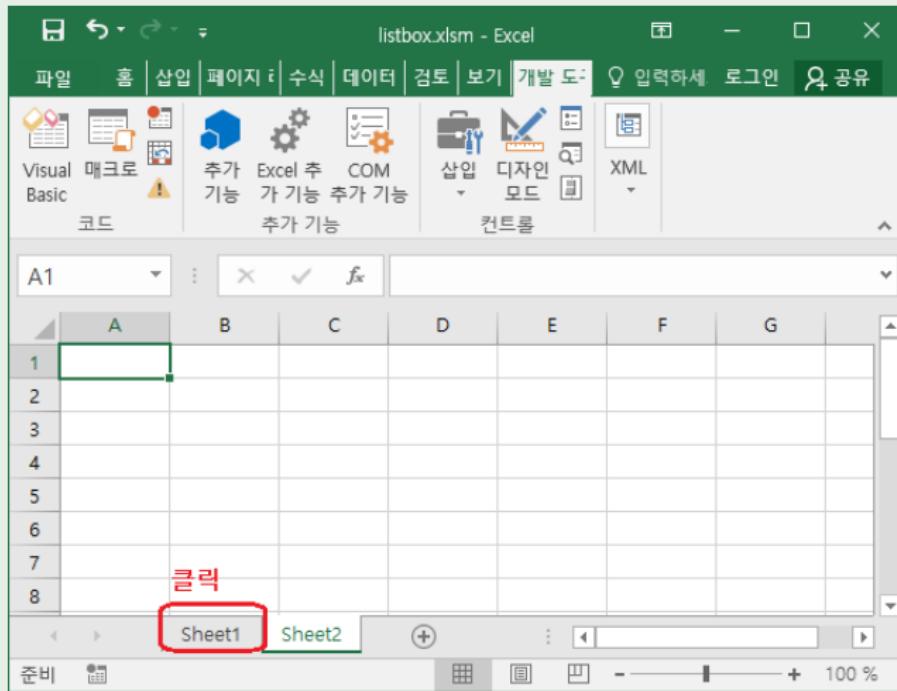
ActiveX 컨트롤 엑셀 연결 – 목록 상자

디자인 모드 해제, 다른 시트로 이동



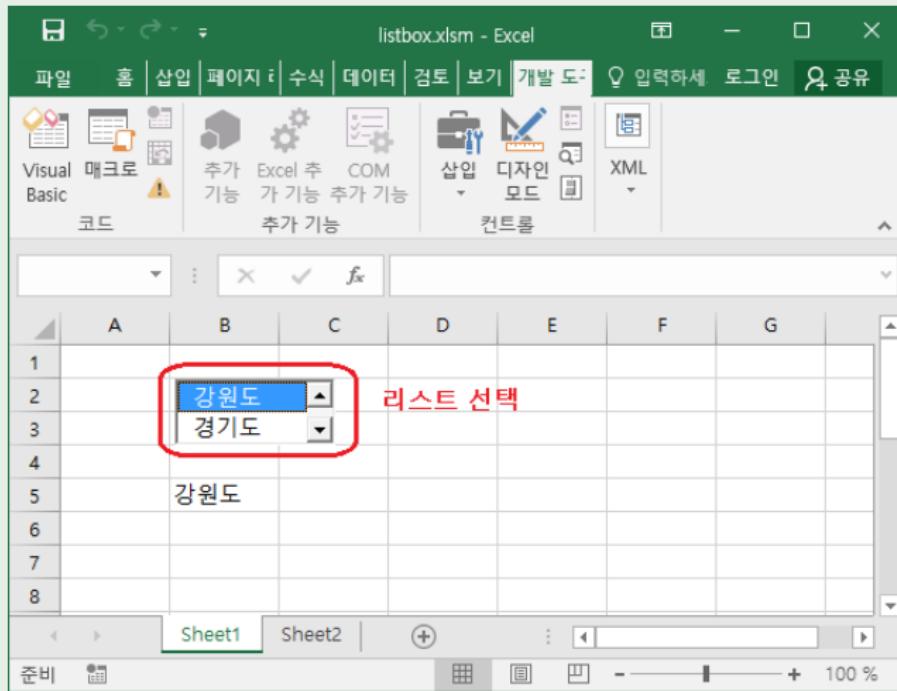
ActiveX 컨트롤 엑셀 연결 – 목록 상자

목록 상자가 있는 시트로 이동



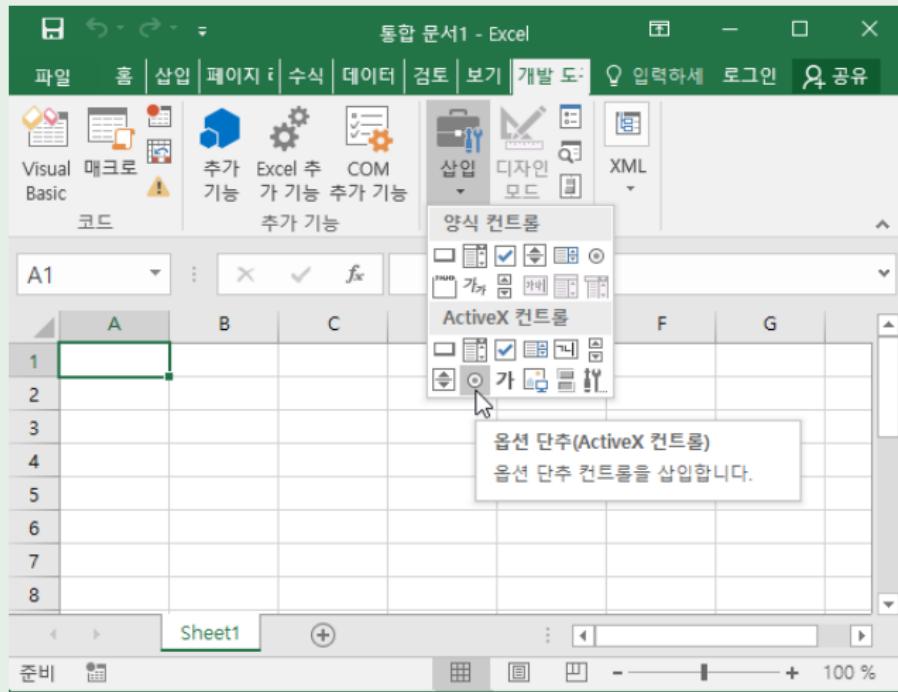
ActiveX 컨트롤 엑셀 연결 – 목록 상자

목록 상자에 설정한 목록, 이벤트 작동 확인



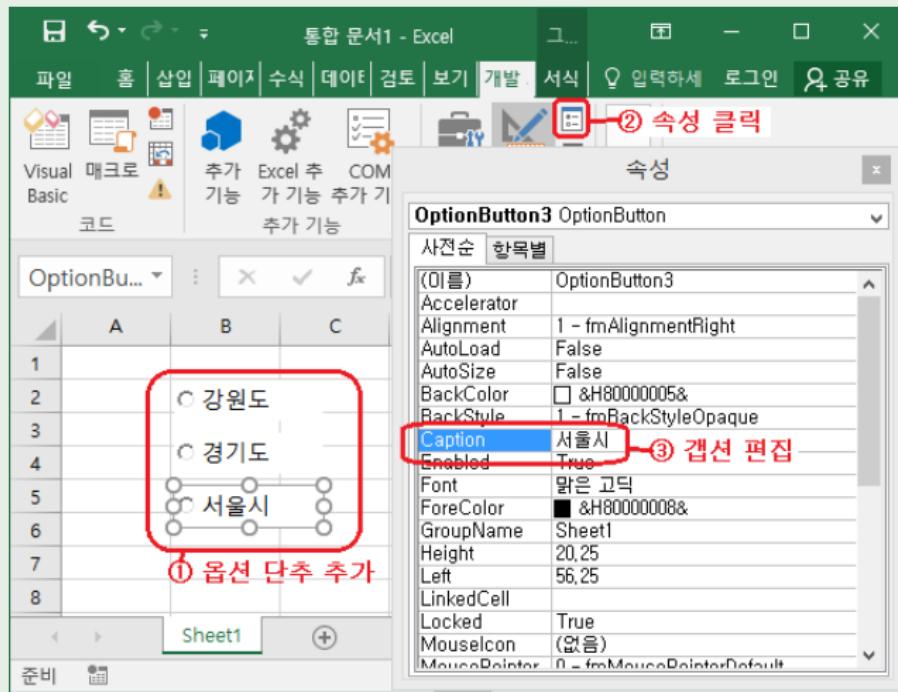
ActiveX 컨트롤 엑셀 연결 – 옵션 단추

추가할 옵션 단추 클릭



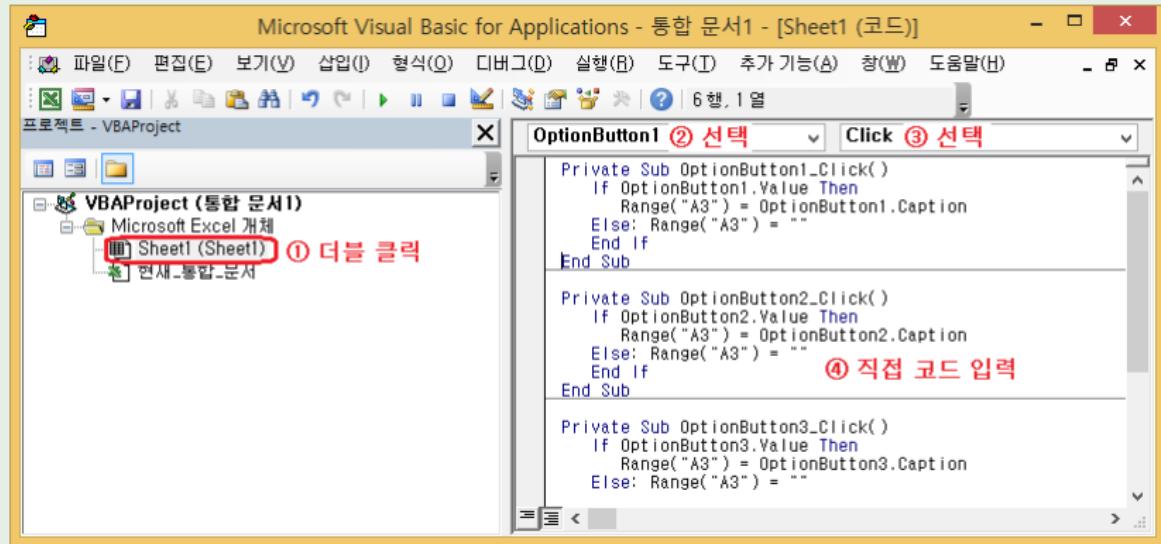
ActiveX 컨트롤 엑셀 연결 – 옵션 단추

옵션 단추 추가, Caption 속성 값 변경



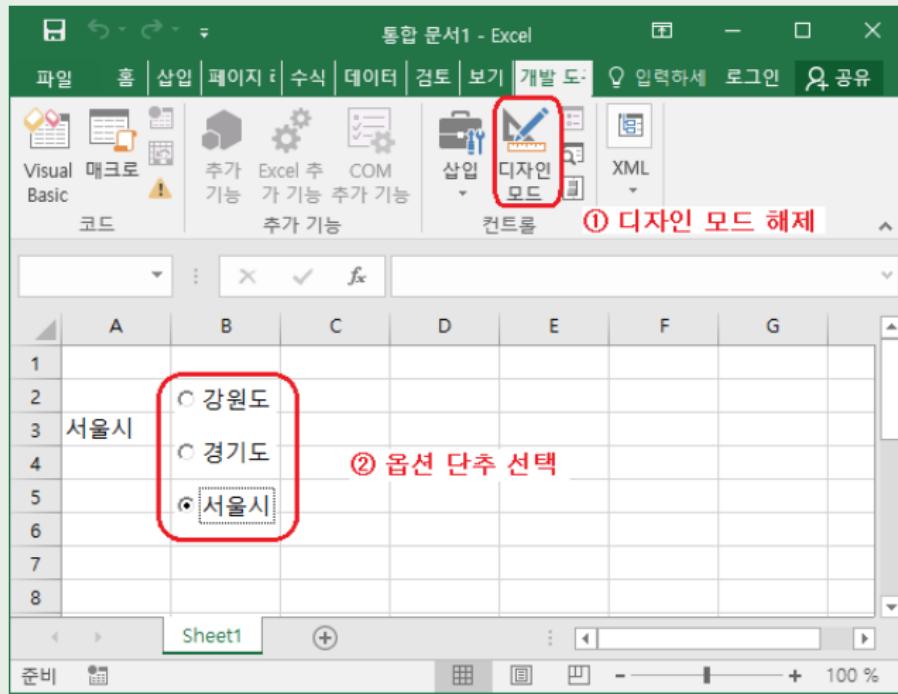
ActiveX 컨트롤 엑셀 연결 – 옵션 단추

옵션 단추 개체 선택, 자동으로 프로시저 생성, 코드 입력



ActiveX 컨트롤 엑셀 연결 – 옵션 단추

디자인 모드 해제, 이벤트 작동 확인



ActiveX 컨트롤 엑셀 연결 – 옵션 단추

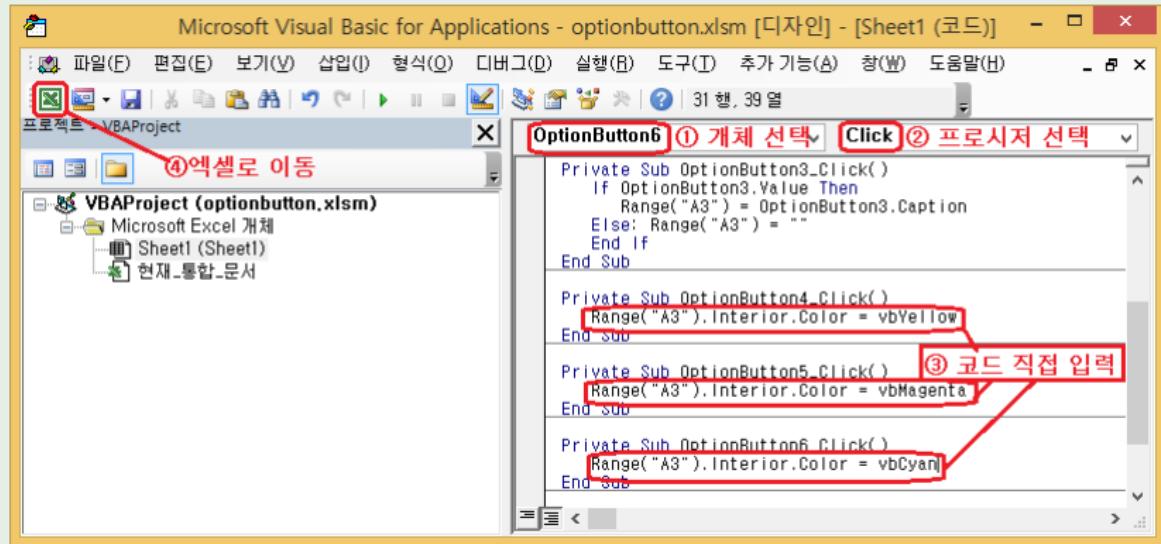
그룹이 다른 옵션 단추 추가, GroupName 속성 값으로 그룹 구분

The screenshot shows the Microsoft Excel ribbon with the '개발' (Development) tab selected. A red box highlights the 'Visual Basic' icon in the ribbon's left pane. In the main area, there is a worksheet with data in columns A, B, and C. Row 3 contains the text '강원도' in column A, and in column B, there are two radio buttons: '강원도' (selected) and '노란색'. Row 4 contains the text '경기도' in column A, and in column B, there are two radio buttons: '경기도' and '자홍색'. Row 5 contains the text '서울시' in column A, and in column B, there are two radio buttons: '서울시' and '하늘색'. A red box encloses the radio buttons in row 5, with the text '① 옵션 단추 추가' (Add option button) pointing to it. To the right of the worksheet, the 'Properties' window is open for 'OptionButton6' (an Option Button control). The 'GroupName' property is highlighted with a red box and the text '② 속성 값 변경' (Change property value) pointing to it. The properties listed in the Properties window are:

(이름)	OptionButton6
Accelerator	
Alignment	1 - fmAlignmentRight
AutoLoad	False
AutoSize	False
BackColor	□ &H80000005&
BackStyle	1 - fmBackStyleOpaque
Caption	하늘색
Enabled	True
Font	맑은 고딕
ForeColor	■ &H80000008&
GroupName	Sheet1
Height	20.25
Left	110,25

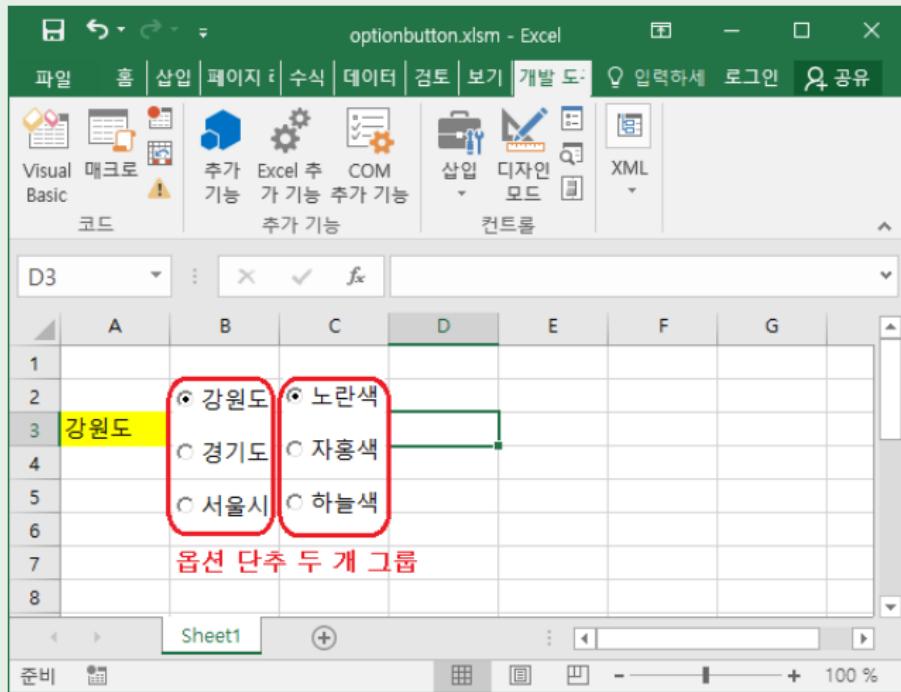
ActiveX 컨트롤 엑셀 연결 – 옵션 단추

옵션 단추 개체 선택, 자동으로 프로시저 생성, 코드 입력



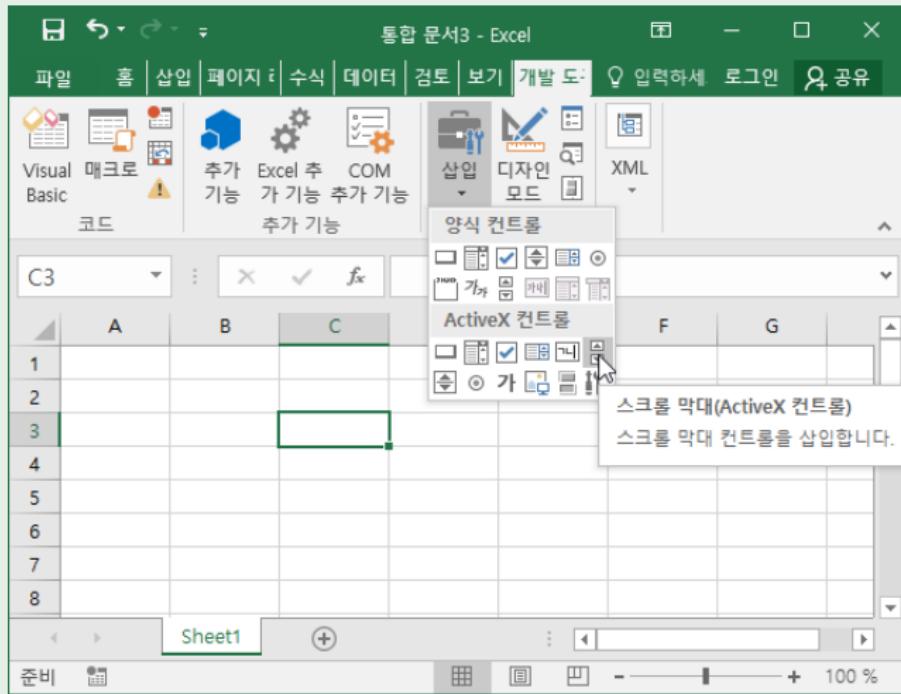
ActiveX 컨트롤 엑셀 연결 – 옵션 단추

디자인 모드 해제, 각 옵션 단추가 독립으로 작동 확인



ActiveX 컨트롤 엑셀 연결 – 스크롤 막대

추가할 스크롤 막대 클릭



ActiveX 컨트롤 엑셀 연결 – 스크롤 막대

스크롤 단추 추가와 속성 값 변경

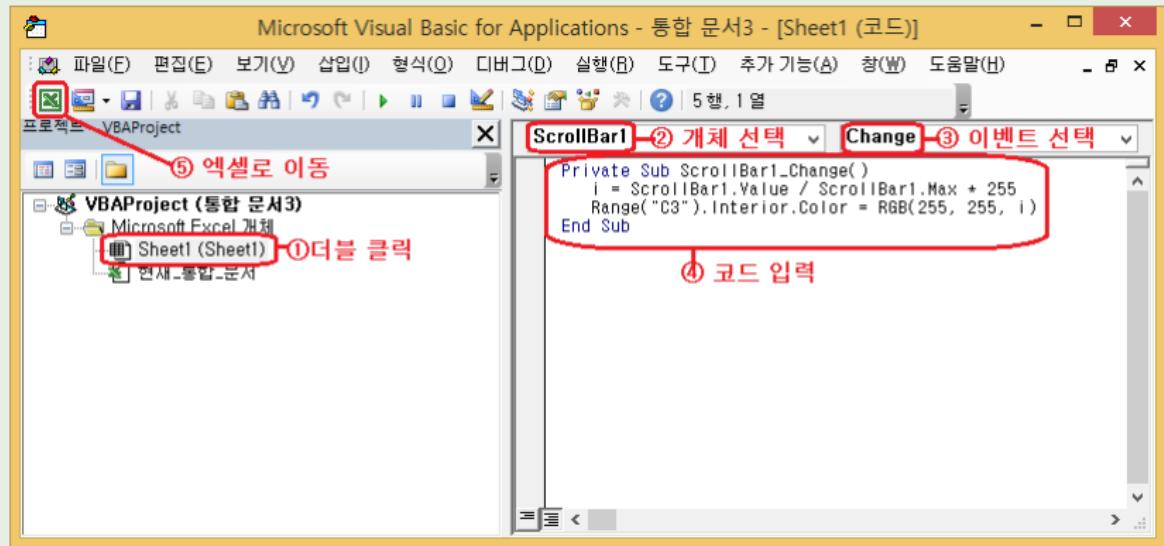
The screenshot shows a Microsoft Excel window with the ribbon menu open. The '개발' (Development) tab is selected. On the left, the 'Visual Basic' icon is highlighted with a red box and labeled ① 클릭 (Click). In the center, a scroll bar control is placed on the worksheet, with its handle and track highlighted by a red box and labeled ① 스크롤바 추가 (Add scroll bar). On the right, the '디자인 모드' (Design Mode) button is highlighted with a red box and labeled ③ 디자인 모드 (③ Design Mode). A callout arrow points from the '디자인 모드' button to the text ③ 디자인 모드. A tooltip '② 속성 값 변경' (② Change properties) points to the 'LinkedCell' entry in the Properties window.

Properties window content:

속성	값
사전순	LargeChange
	1
	Left
	69
	LinkedCell
	C3 ② 속성 값 변경
	Locked
	True
	Max
	32767
	Min
	0
	MouseIcon
	(없음)
	MousePointer
	0 - fmMousePointerDefault
	Orientation
	-1 - fmOrientationAuto
	Placement
	2
	PrintObject
	True
	ProportionalThumb
	True
	Shadow
	False
	SmallChange
	1

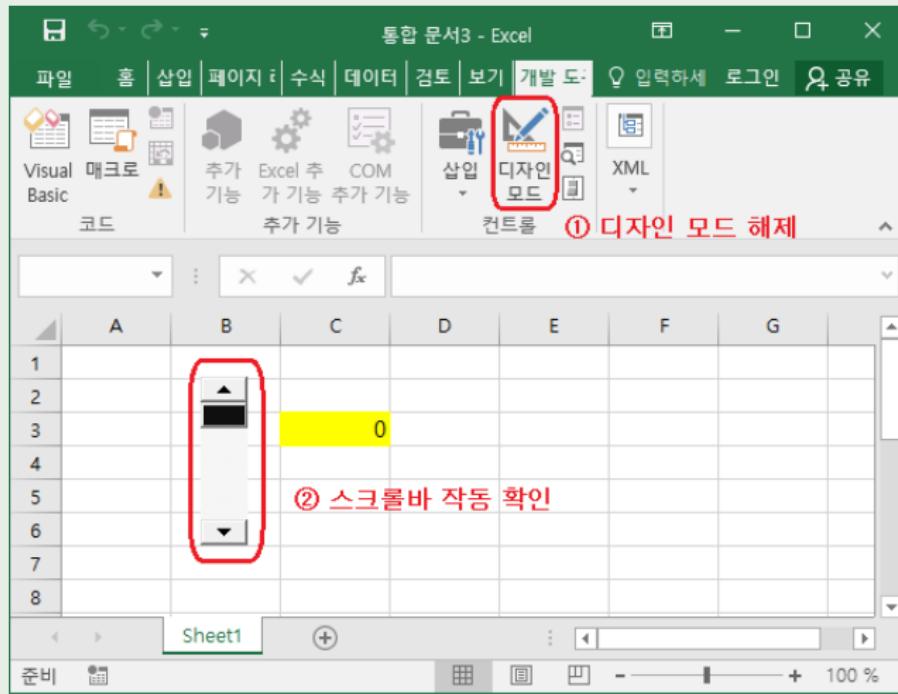
ActiveX 컨트롤 엑셀 연결 – 스크롤 막대

스크롤 단추 개체 선택, 자동으로 프로시저 생성, 코드 입력



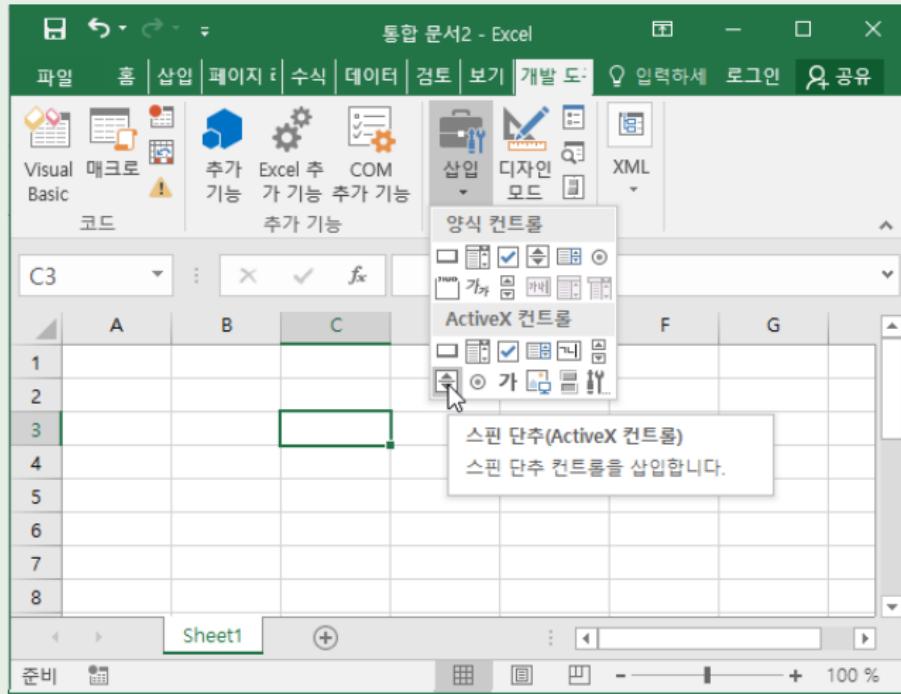
ActiveX 컨트롤 엑셀 연결 – 스크롤 막대

디자인 모드 해제, 이벤트 작동 확인



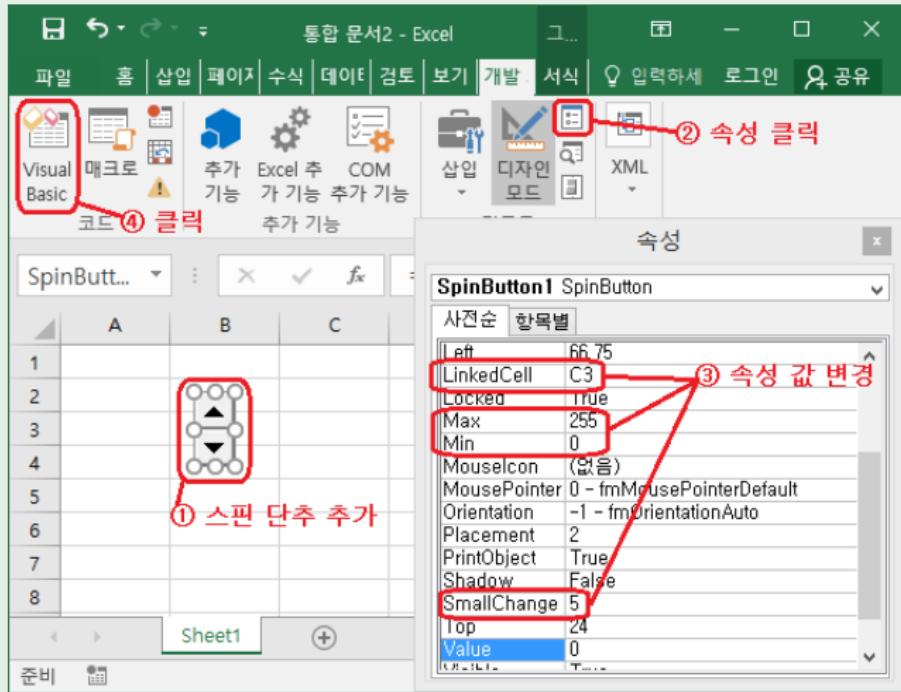
ActiveX 컨트롤 엑셀 연결 – 스픈 단추

추가할 스픈 단추 클릭



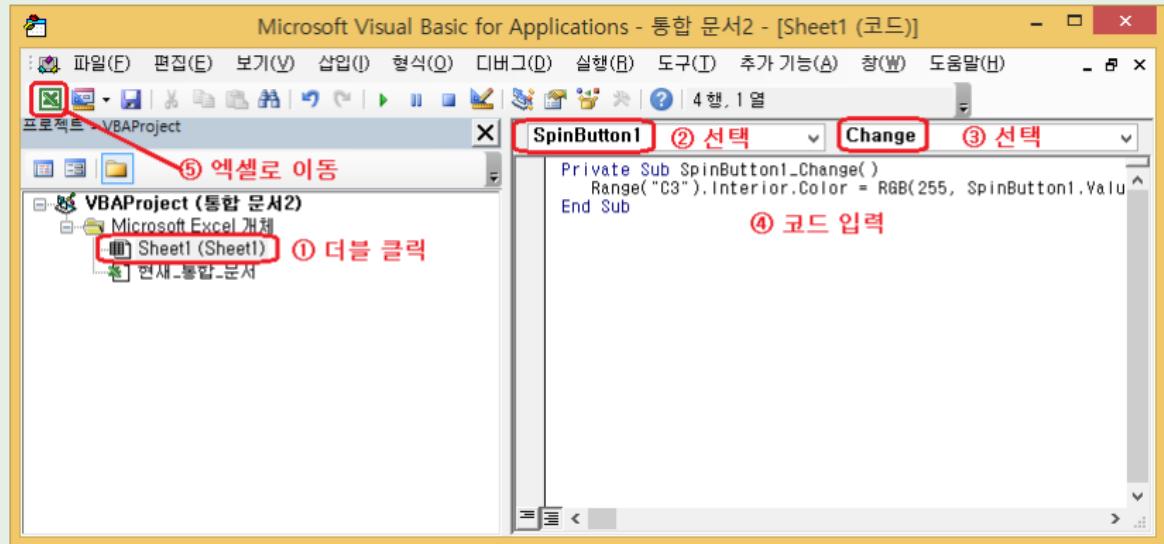
ActiveX 컨트롤 엑셀 연결 – 스픈 단추

스핀 단추 추가와 속성 값 변경



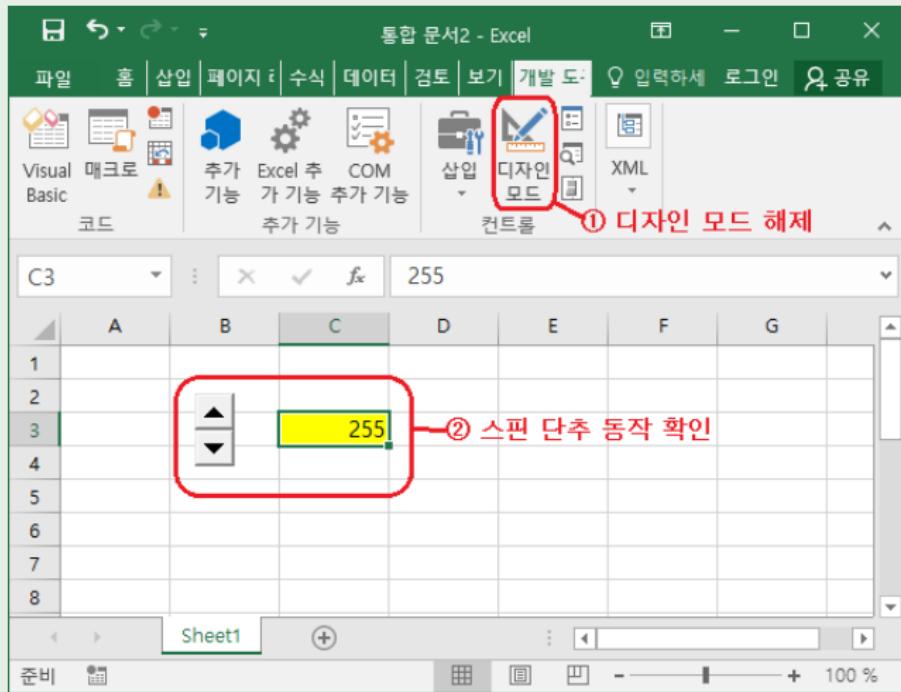
ActiveX 컨트롤 엑셀 연결 – 스피ن 단추

스피ن 단추 개체 선택, 자동으로 프로시저 생성, 코드 입력



ActiveX 컨트롤 엑셀 연결 – 스픈 단추

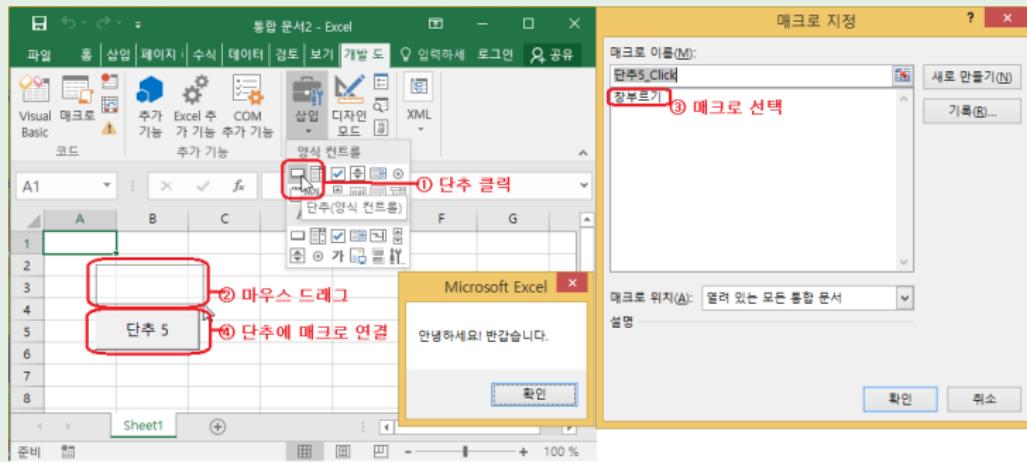
디자인 모드 해제, 이벤트 작동 확인



양식 컨트롤 엑셀 연결

양식 컨트롤 엑셀 연결 – 명령 단추

명령 단추 추가, 매크로로 지정

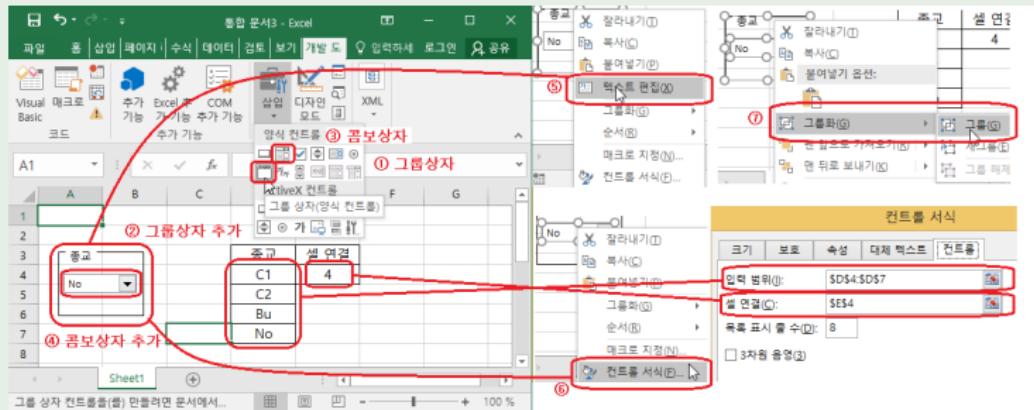


매크로 내용

```
1 Sub 창부르기()
2     MsgBox "안녕하세요! 반갑습니다."
3 End Sub
```

양식 컨트롤 엑셀 연결 – 콤보 상자

콤보 상자 추가, 매크로 지정



양식 컨트롤 엑셀 연결 – 확인란

확인란 추가, 매크로 지정

The screenshot shows an Excel spreadsheet with a form control checkbox in cell A1. The checkbox is labeled "확인란 추가" (④ Add Checkbox). The cell contains the formula `=IF(F1=TRUE, "TRUE", "")`. A red box highlights the formula bar with the value `=E4`.

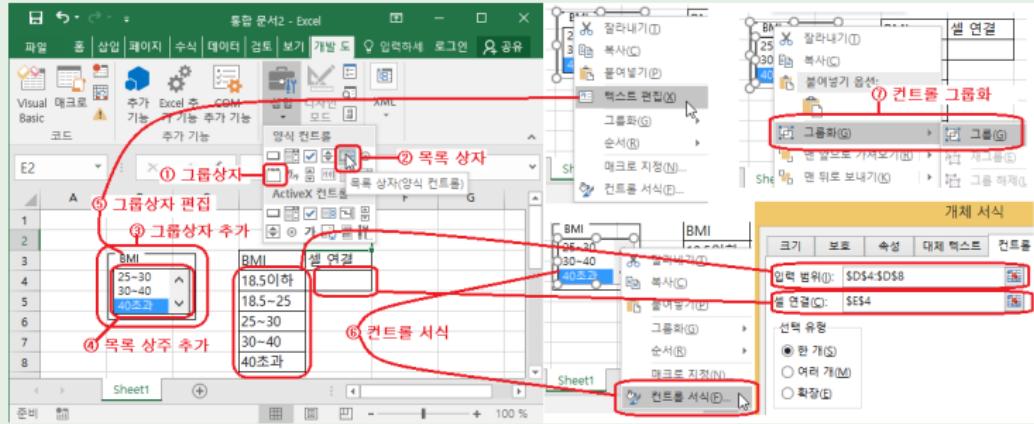
To the right, the "Control Properties" dialog is open (컨트롤 서식). It shows the formula `=E4` in the "설 연결(L)" (Link Address) field. The "Type" section has the radio button "선택하지 않은 상태(U)" (Not Selected State) selected.

Annotations in red boxes:

- ① 그룹상자 추가 (Add Group Box): Points to the group box icon in the ribbon.
- ② 그룹상자 추가 (Add Group Box): Points to the group box icon in the ribbon.
- ③ 확인란 (Checkbox): Points to the checkbox control in the ribbon.
- ④ 확인란 추가 (Add Checkbox): Points to the checkbox in cell A1.
- ⑤ 컨트롤 값 연결 (Link Control Value): Points to the formula `=IF(F1=TRUE, "TRUE", "")` in cell A1.

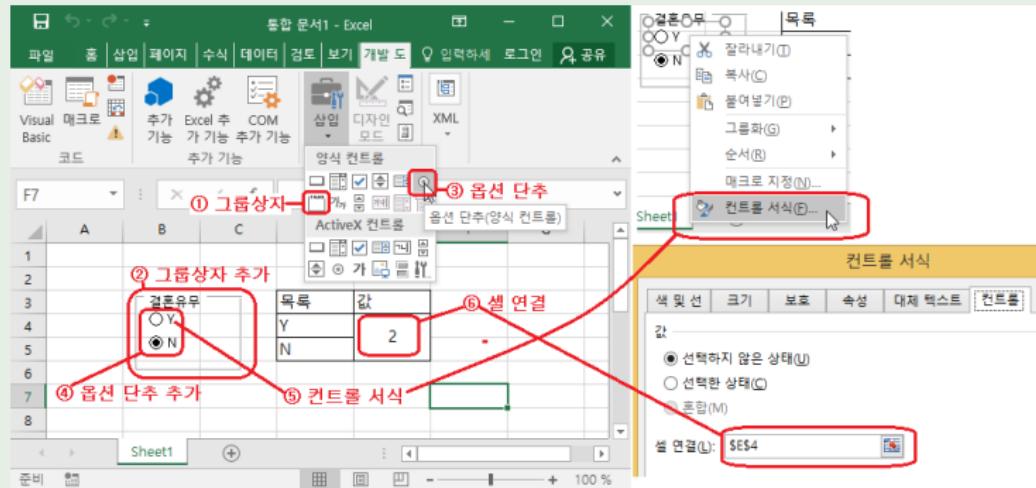
양식 컨트롤 엑셀 연결 – 목록 상자

목록 상자 추가, 매크로 지정



양식 컨트롤 엑셀 연결 – 옵션 단추

옵션 단추 추가, 매크로로 지정



고급 필터 만들기

양식 컨트롤 – 고급 필터

고급 필터 완성

bmi4.xlsm - Excel

파일 흡입 페이지 수식 데이터 검토 보기 개발 도구 입력하세요. 로그인 공유

붙여넣기 클립보드 글꼴 스타일 셀 편집

B6 A B C D E F G

① 콤보 상자 ② 확인란 ③ 옵션 단추 ④ 목록 상자 ⑤ 명령 단추

	키	몸무게	출생년도	종교	성별	결혼여부	BMI
7	153	56	1976	C1	F	N	23.92242
8							

준비 강희모

결과

엑셀 VBA

106 / 123

양식 컨트롤 – 고급 필터

고급 필터 조건, 양식컨트롤 참조, 셀 연결

The screenshot shows an Excel spreadsheet titled "bmi4.xlsxm - Excel". The ribbon tabs include 파일 (File), 홈 (Home), 삽입 (Insert), 페이지 레이아웃 (Page Layout), 수식 (Formulas), 데이터 (Data), 검토 (Review), 보기 (View), 개발 도구 (Developer Tools), and 도록인 (Developer Tools). The formula bar shows cell I11.

Table 1 (Top Left):

종교	BMI
C1	18.5이하
C2	18.5~25
Bu	25~30
No	30~40
	40조과

Table 2 (Top Right):

④ 양식컨트롤 이벤트 값들 고급필터 조건에 사용				
[고급필터 조건]				
종교	성별	결혼여부	BMI	BMI
C2	F	N	>=18.5	<25
C2	M	N	>=18.5	<25

Table 3 (Bottom Left):

[양식 컨트롤 셀 연결]				
종교	성별	결혼여부	BMI	BMI
2	TRUE	2	2	
	TRUE			

Table 4 (Bottom Right):

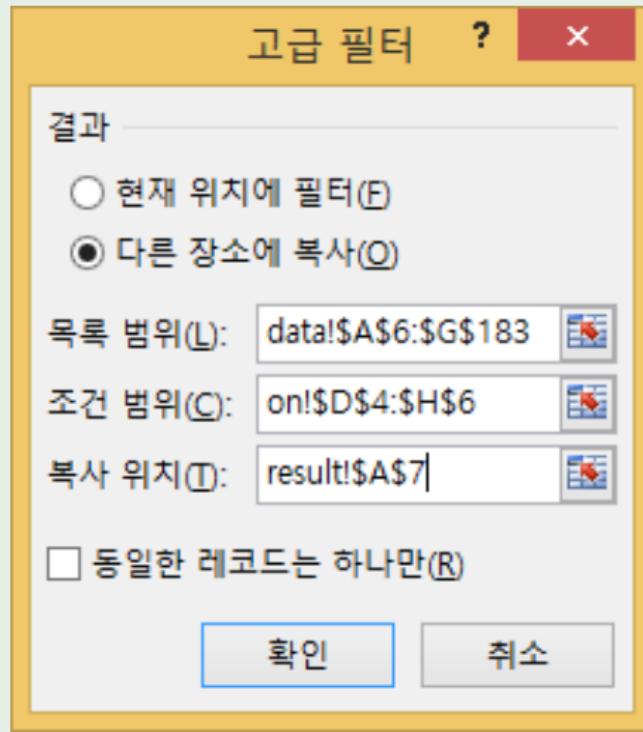
[BMI 범위]		
index	BMI	BMI
1	0	18.5
2	18.5	25
3	25	30
4	30	40
5	40	100

Annotations:

- ① 콤보 상자 목록 리스트 상자 목록 ②
- ③ 양식컨트롤 이벤트 값 출력

양식 컨트롤 – 고급 필터

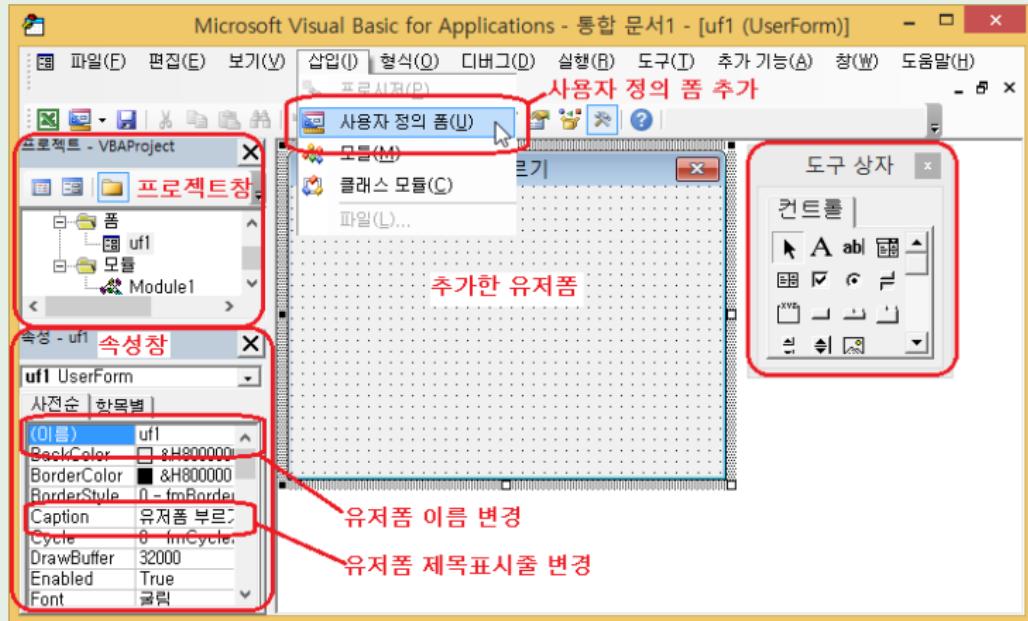
고급 필터 설정



사용자 정의 폼

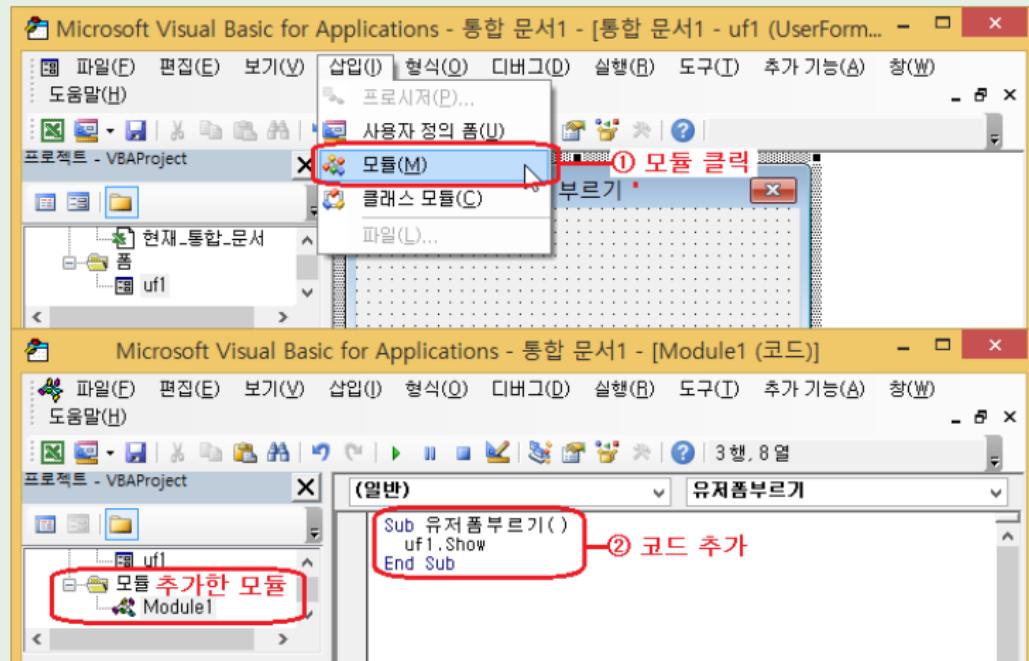
사용자 정의 폼

사용자 정의 폼 추가



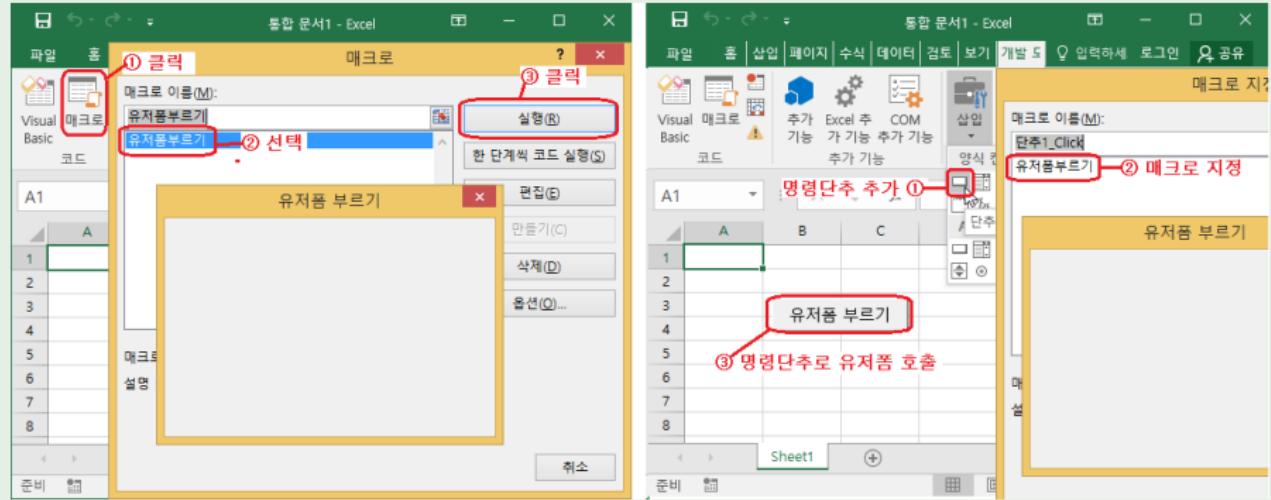
사용자 정의 폼

사용자 정의 폼에 모듈 추가와 코드 입력



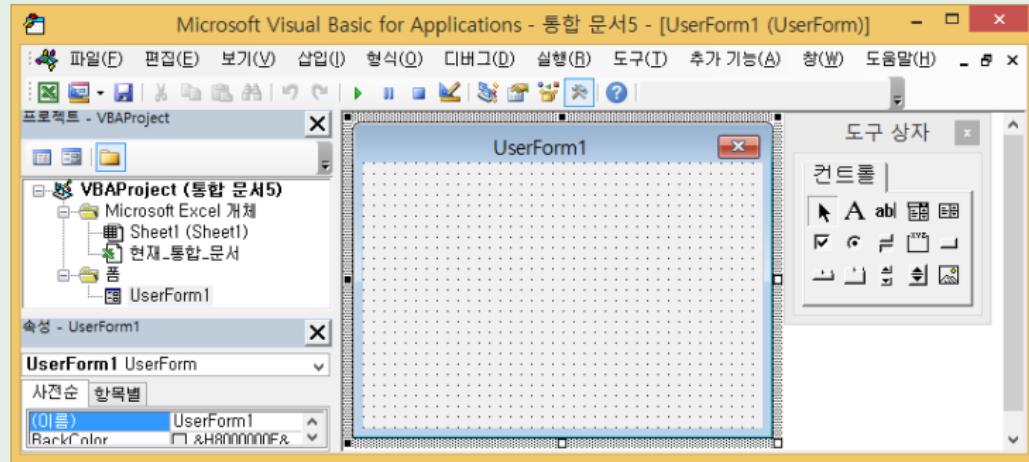
사용자 정의 폼

사용자 정의 폼 호출하는 두 가지 방법



사용자 정의 폼

사용자 정의 폼 추가

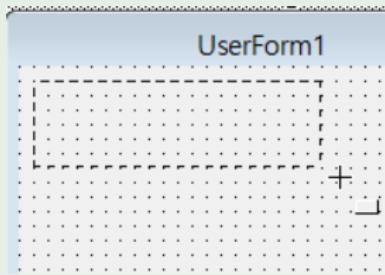


사용자 정의 폼

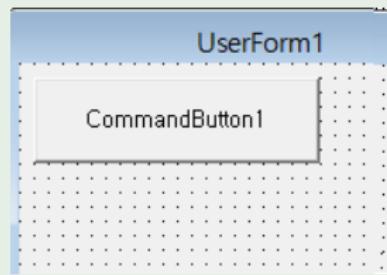
사용자 정의 폼에 명령 단추 추가



(a) 명령 단추



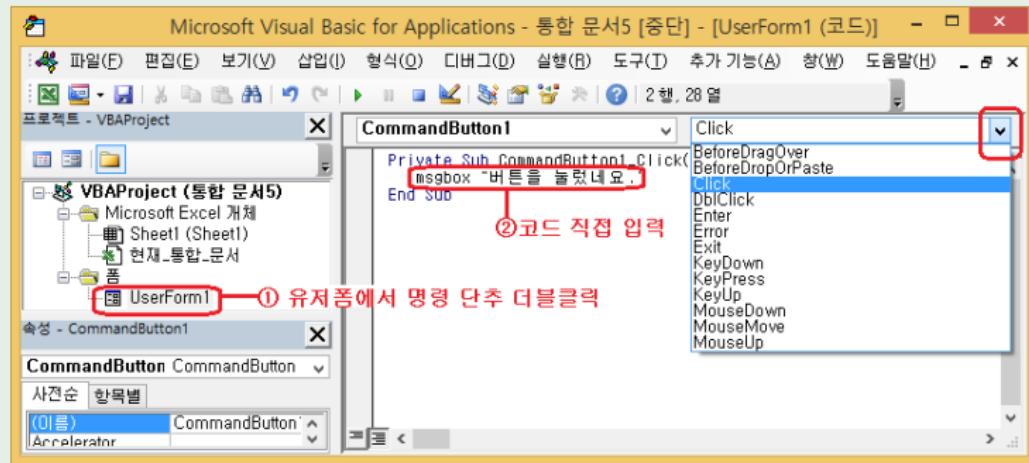
(b) 명령 단추 영역



(c) 명령 단추 추가

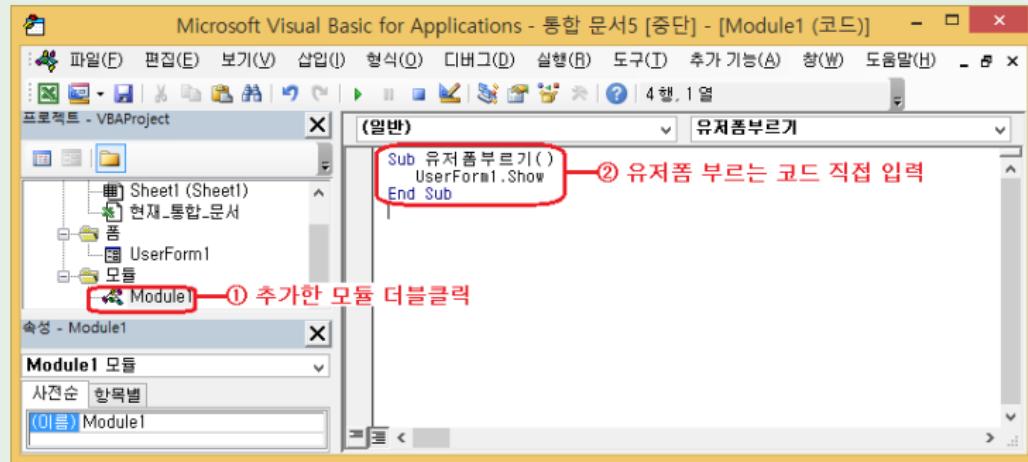
사용자 정의 폼

명령 상자 이벤트 프로시저



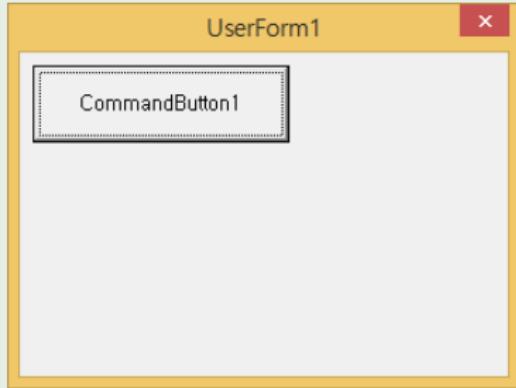
사용자 정의 폼

사용자 정의 폼 추가하고 호출

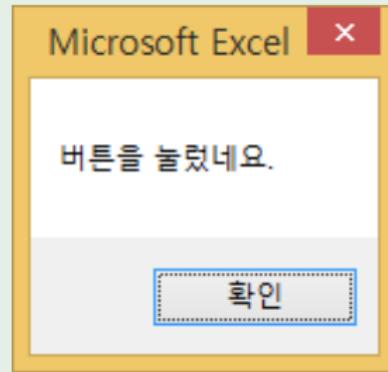


사용자 정의 폼

사용자 정의 폼에 명령 단추 추가하고 실행



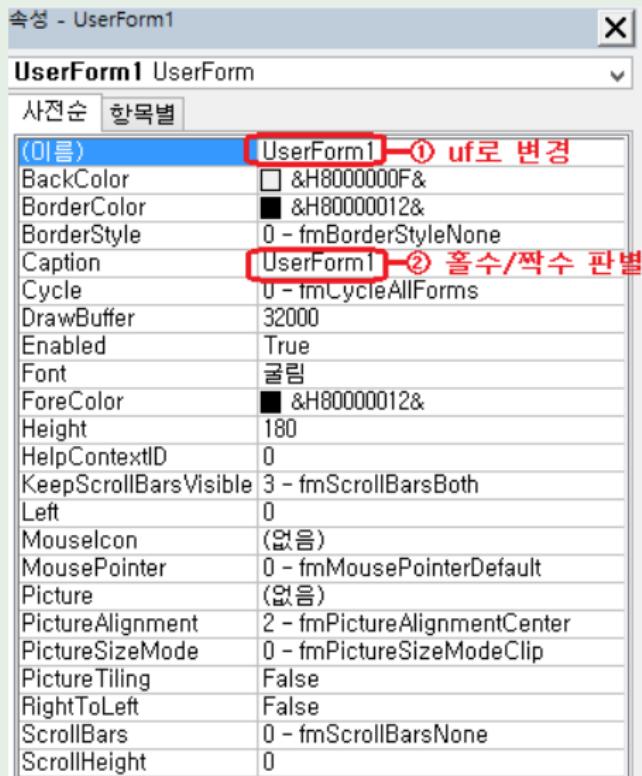
(d) 유저폼 실행



(e) 명령 단추 클릭

사용자 정의 폼

사용자 정의 폼 – 홀·짝수 판별; 유저폼 속성창



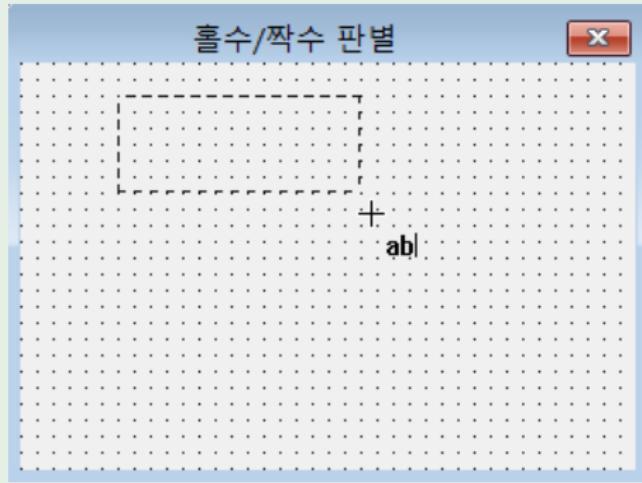
사용자 정의 폼

사용자 정의 폼 – 홀·짝수 판별; 제목표시줄 변경



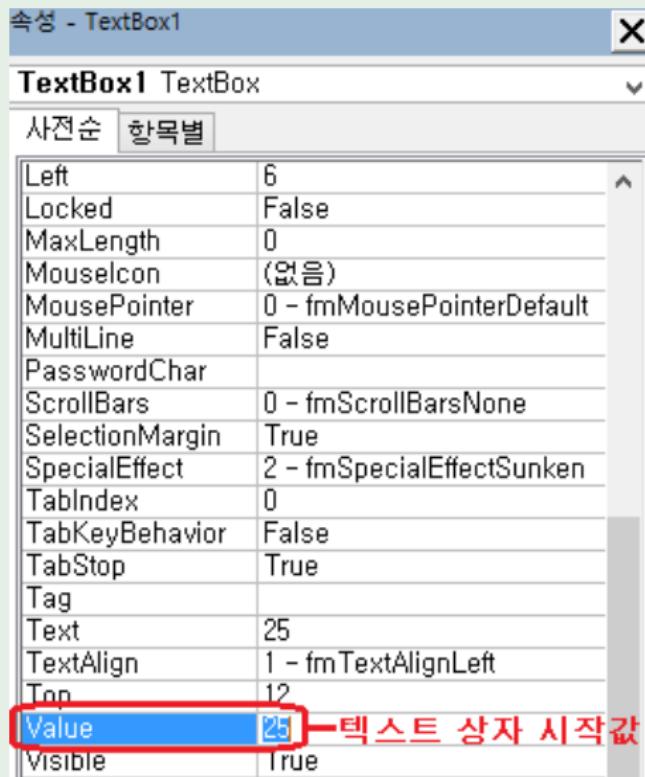
사용자 정의 폼

사용자 정의 폼 – 홀·짝수 판별; 텍스트 상자 추가



사용자 정의 폼

사용자 정의 폼 – 홀·짝수 판별; 텍스트 상자 속성창



사용자 정의 폼

사용자 정의 폼 – 홀·짝수 판별; 유저폼 완성



사용자 정의 폼

사용자 정의 폼 – 홀·짝수 판별; 명령 단추 이벤트 프로시저 코드

```
1 Private Sub cb1_Click()
2     If (tb.Value Mod 2 = 0) Then
3         Range("A1") = tb.Value & "은(는) 짝수"
4     Else
5         Range("A1") = tb.Value & "은(는) 홀수"
6     End If
7 End Sub
```

사용자 정의 폼 – 홀·짝수 판별; 유저폼 호출 코드

```
1 Sub 유저폼호출()
2     uf.Show
3 End Sub
```