

# **Import/Export Data Using VBA**

# Objectives

---

- Referencing Excel Cells in VBA
- Importing Data from Excel to VBA
- Using VBA to Modify Contents of Cells



✓ 새 서브 프로시저 작성하기

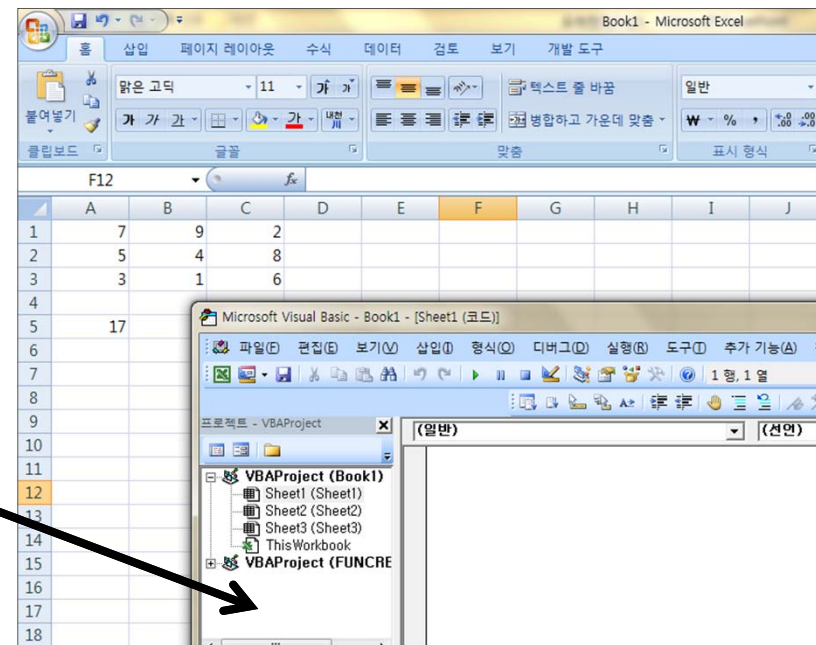
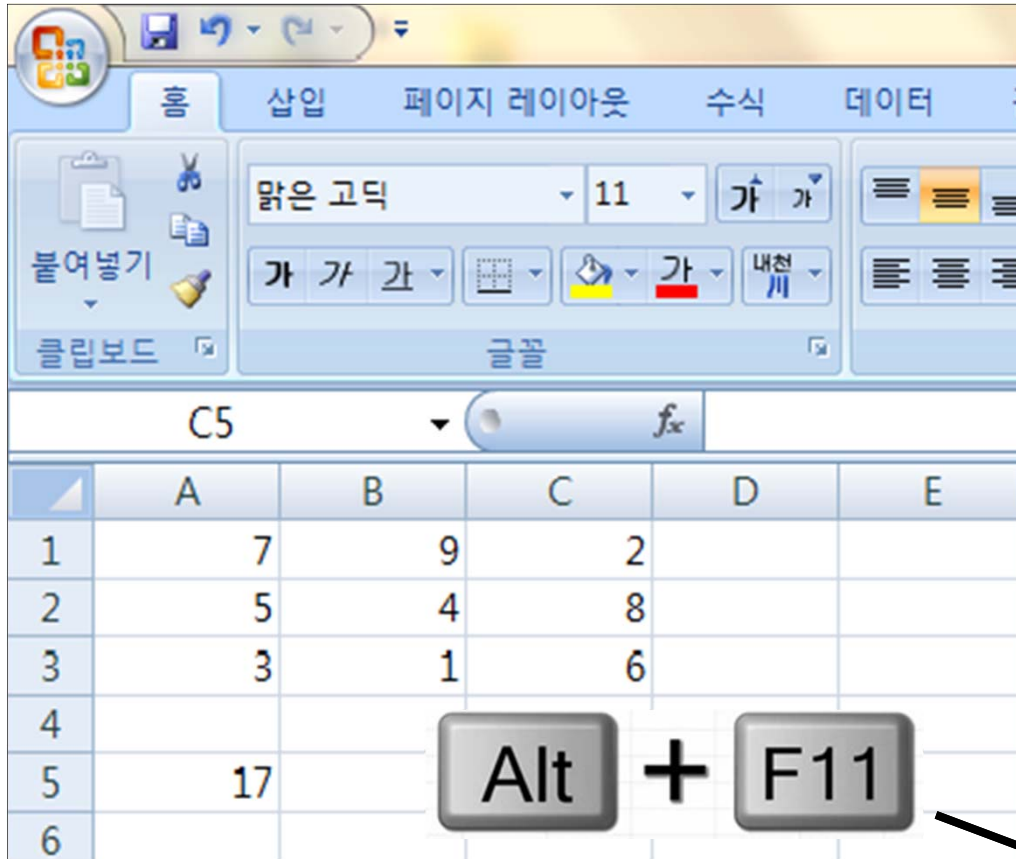


✓ 프로시저 실행하고 결과 확인하기

✓ VBA 코드 이해하기

# Referencing Excel Cells in VBA

# Referencing Excel Cells in VBA



✓ 엑셀창에서 **Alt+F11** 단축기  
Visual Basic 편집기 창호출.

# Referencing Excel Cells in VBA

**1** 삽입(I) 형식(O) 디버그(D) 실행(R) 도구(T) 추가 기능(A) 창(W) 도움말(H)

**2** 프로시저(P)... 사용자 정의 품(U) **모듈(M)** 클래스 모듈(C) 파일(L)...

**3** ✓모듈 개체 추가

➤모듈개체 추가하는 이유

- 전체 개체에서 사용 가능한 여러 가지 선언과 정의 작업이 가능하기 때문에

# Referencing Excel Cells in VBA

---

## ✓새 서브 프로시저 작성하기

```
Option Explicit  
Dim a As Double, b As Double, c As Double
```

```
Sub getvalue()  
a = Range("A5").Value  
MsgBox ("value in A5") & a  
End Sub
```

```
Sub putvalue()  
b = 25  
Range("A6").Value = b  
End Sub
```

```
Sub putvalueInputbox()  
c = InputBox("Enter a number to be placed in cell A7:")  
End Sub
```

# Referencing Excel Cells in VBA

✓ 프로시저 실행하고 결과 확인하기

1 F5 누르기

2 실행(R)

취소

한 단계씩 코드 실행(S)

편집(E)

만들기(C)

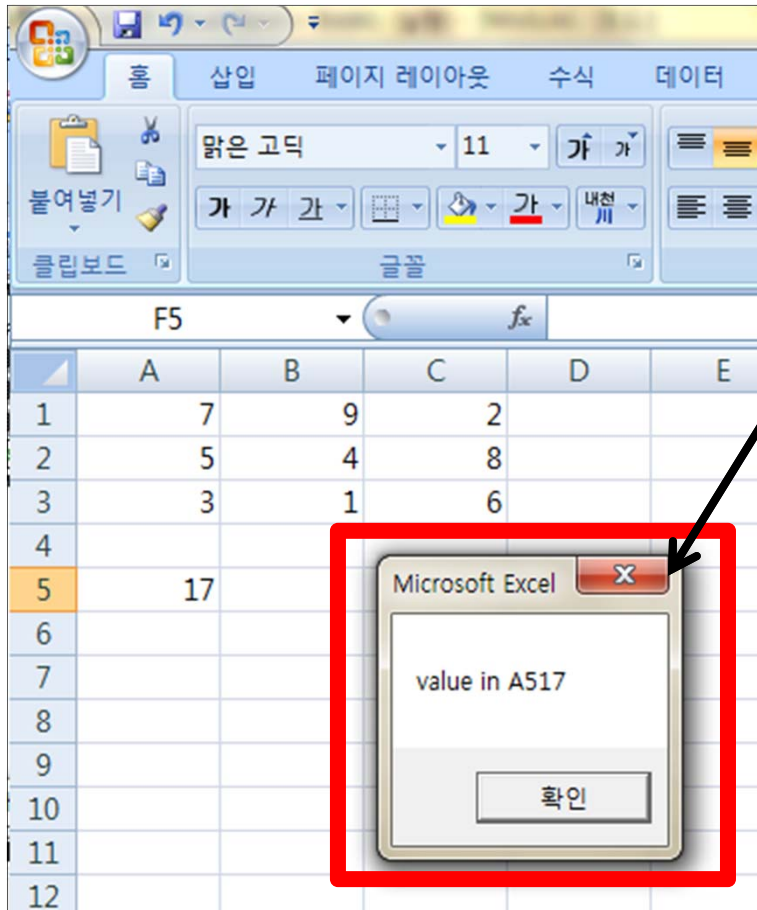
삭제(D)

마이크로 이름(M):  
getvalue  
putvalue  
putvalueinputbox

마이크로 위치(A): VBAProject (Book1)

# Referencing Excel Cells in VBA

✓프로시저 실행하고 결과 확인하기



```
Option Explicit  
Dim a As Double, b As Double, c As Double
```

```
Sub getvalue()  
a = Range("A5").Value  
MsgBox ("value in A5") & a  
End Sub
```



# Referencing Excel Cells in VBA

## ✓VBA 코드 이해하기

```
Option Explicit
```

```
Dim a As Double, b As Double, c As Double
```

### ✓ Option Explicit 명령

변수 선언 요청 명령. '변수 선언 요구'에 체크 표시한 경우 나타남

### ✓ 변수와 상수의 선언

**선언문 변수(상수)명 As 데이터형식**

•선언문 ⇒ 변수인 경우 **Dim문**, 상수인 경우 **Const문** 사용

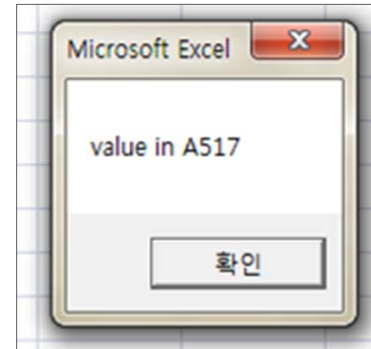
•데이터 형식

자료형	이름	범위	사이즈 (바이트)
바이트형	Byte	0~255	1
정수형	Integer	-32768~32767	2
긴 정수형	Long	-2147483648~2147483647	4
단정도부동소수 점수형	Single	$\pm 3.4 \times 10^{38} \sim \pm 1.4 \times 10^{-45}$	4
배정도부동소수 점수형	Double	$\pm 1.8 \times 10^{308} \sim \pm 4.9 \times 10^{-324}$	8

# Referencing Excel Cells in VBA

## ✓VBA 코드 이해하기

```
Sub getvalue( )  
a = Range("A5").Value  
MsgBox ("value in A5") & a  
End Sub
```



## ✓서브 프로시저(Sub Procedure) 구성

```
Sub 프로시저명( )  
실행명령  
End Sub
```

### ➤ a = Range("A5"). Value

✓Range 개체는 셀 범위를 나타내는 개체.

✓A5 cell 의미

✓속성명

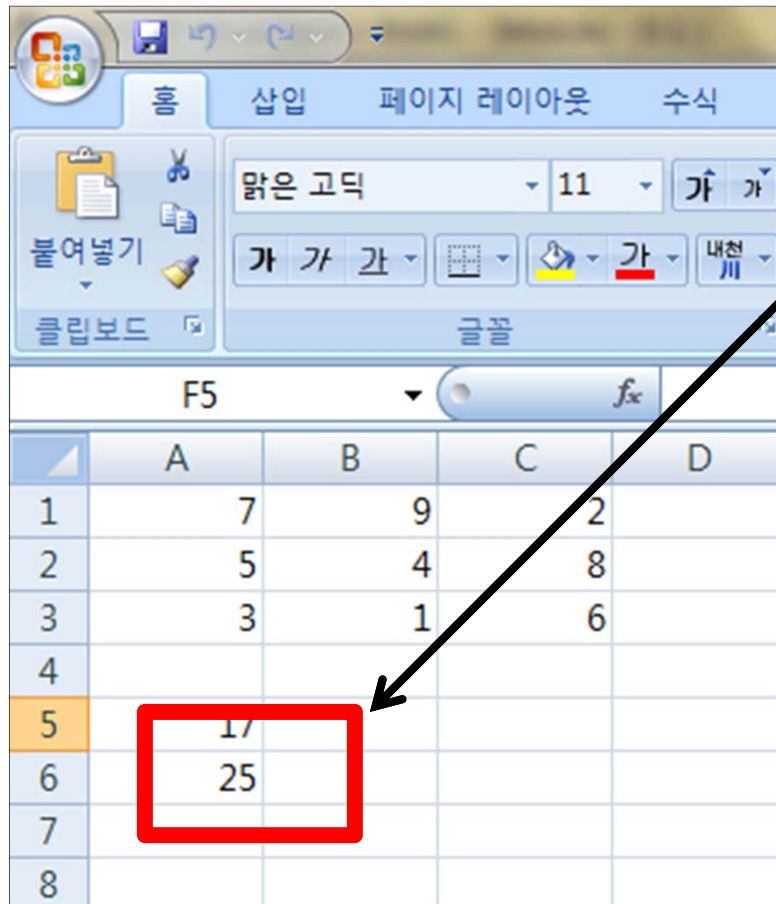
### ➤ MsgBox ("Value in A5") & a

✓Msgbox 함수는 사용자에게 메시지 창을 표시할 때 사용  
✓Msbbox(메시지 내용)

✓지정한 a의 값

# Referencing Excel Cells in VBA

✓프로시저 실행하고 결과 확인하기



	A	B	C	D
1	7	9	2	
2	5	4	8	
3	3	1	6	
4				
5	17			
6	25			
7				
8				

```
Sub putvalue( )  
b = 25  
Range("A6").Value = b  
End Sub
```

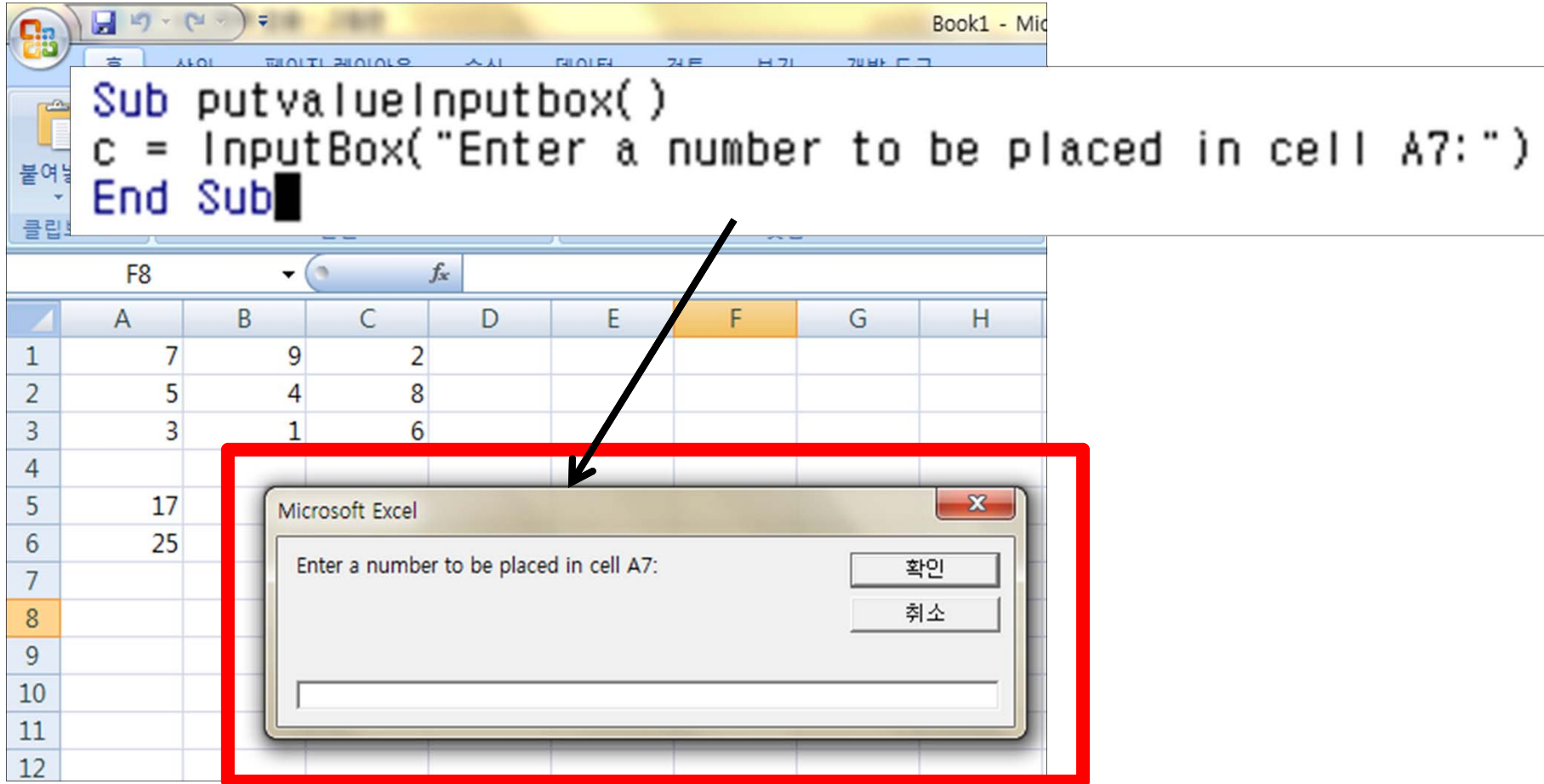
✓VBA 코드 이해하기

**b=25**  
**Range("A6"). Value=b**

✓A6 cell에 25(b)넣기

# Referencing Excel Cells in VBA

✓프로시저 실행하고 결과 확인하기



The image shows a screenshot of the Microsoft Excel VBA editor. The VBA code in the editor is as follows:

```
Sub putvalueInputbox( )  
c = InputBox("Enter a number to be placed in cell A7:")  
End Sub
```

Below the code, a portion of the Excel spreadsheet is visible. The data in the spreadsheet is as follows:

	A	B	C	D	E	F	G	H
1	7	9	2					
2	5	4	8					
3	3	1	6					
4								
5	17							
6	25							
7								
8								
9								
10								
11								
12								

An input dialog box titled "Microsoft Excel" is displayed over the spreadsheet. The dialog box contains the text "Enter a number to be placed in cell A7:" and two buttons: "확인" (OK) and "취소" (Cancel). A red box highlights the dialog box, and an arrow points from the VBA code to it.

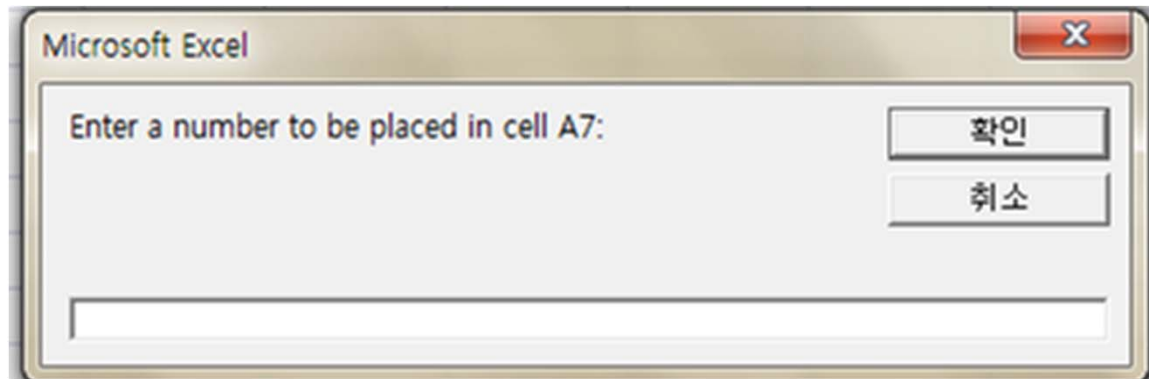
# Referencing Excel Cells in VBA

## ✓VBA 코드 이해하기

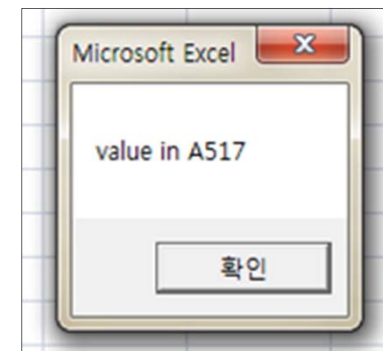
```
Sub putvalueInputbox( )  
c = InputBox("Enter a number to be placed in cell A7:")  
End Sub
```

c=**InputBox** ("Enter a number to be placed in cell a7:")

- ✓ **InputBox** 함수: 사용자의 **입력 값을 받아 처리**하고자 할 때 사용.
- ✓ **MsgBox** 함수: 사용자에게 **전달한 내용을 메시지창에 표시**.  
특정상황에 **사용자의 선택이 필요**한 경우 사용.



InputBox 함수



MsgBox 함수

# Referencing Excel Cells in VBA

**1** 보기(V) | 삽입(I) | 형식(O) | 디버그(D) | 실행(R) | 도구(T) | 추가 기능(A) | 창(W) | 도움말(H)

코드(C) F7  
개제(B) Shift+F7  
정의(D) Shift+F2  
이전 위치(N) Ctrl+Shift+F2  
개제 찾아보기(O) F2  
직접 실행 창(I) Ctrl+G  
**지역 창(S)** **2**  
조사식 창(H)  
호출 스택(K)... Ctrl+L  
프로젝트 탐색기(P) Ctrl+R  
속성 창(W) F4  
도구 상자(X)  
탭 순서(A)  
도구 모음(I)  
Microsoft Excel Alt+F11

```
Option Explicit
Dim a As Double, b As Double, c As Double

Sub getvalue()
a = Range("A5").Value
MsgBox ("value in A5") & a
End Sub

Sub putvalue()
a = 25
Range("A6").Value = b
End Sub

Sub putvalueinputbox()
c = InputBox("Enter a number to be placed in cell A7:")
End Sub
```

**4** F8 키 누르기  
'호출스택표식'아이콘  
코드가 중단된 위치 표시

지역	값	형식
VBAPProject.Module1.putvalue		
Module1		Module1/Module1
a	17	Double
b	25	Double
c	0	Double

# **Importing Data from Excel to VBA**

# Importing Data from Excel to VBA

The image shows a screenshot of the Microsoft Visual Basic Editor interface. The main window displays the '삽입(I)' (Insert) menu, which is open. The '모듈(M)' (Module) option is highlighted, and a red circle with the number '2' is placed over it. The '프로젝트 - VBAProject' (Project - VBAProject) window on the left shows the project structure, including 'VBAProject (Book1)' and 'VBAProject (FUNCRE)'. A red circle with the number '1' is placed over the '삽입(I)' menu item. A red circle with the number '3' is placed over the 'Module2' item in the project tree, which is highlighted with a red box. A dashed box on the right contains the text '✓모듈 개체 추가' (Check: Add module object).

1

2

3

✓모듈 개체 추가



# Importing Data from Excel to VBA

---

✓새 서브 프로시저 작성하기

(일반)

```
Option Explicit
Dim a As Variant
Sub GetArray2( )
a = Range( "A1:C5" )
Range( "D5:F7" ).Value = a
End Sub
|
```

# Importing Data from Excel to VBA

✓프로시저 실행하고 결과 확인하기

The image shows an Excel spreadsheet with the following data:

	A	B	C	D	E	F	G
1		7	9	2			
2		5	4	8			
3		3	1	6			
4							
5	17			7	9	2	
6	25			5	4	8	
7				3	1	6	
8							
9							

A blue box highlights the source data in cells A1:C3. A red box highlights the destination data in cells D5:F7. A black box labeled "생성" (Generate) with an arrow points from the source data to the destination data.

# Importing Data from Excel to VBA

✓VBA 코드 이해하기

	A	B	C	D	E	F	G
1	7	9	2				
2	5	4	8				
3	3	1	6				
4							
5	17			7	9	2	
6				5	4	8	
7				3	1	6	
8							
9							

```
Sub GetArray2()  
a = Range("A1:C5")  
Range("D5:F7").Value = a  
End Sub
```

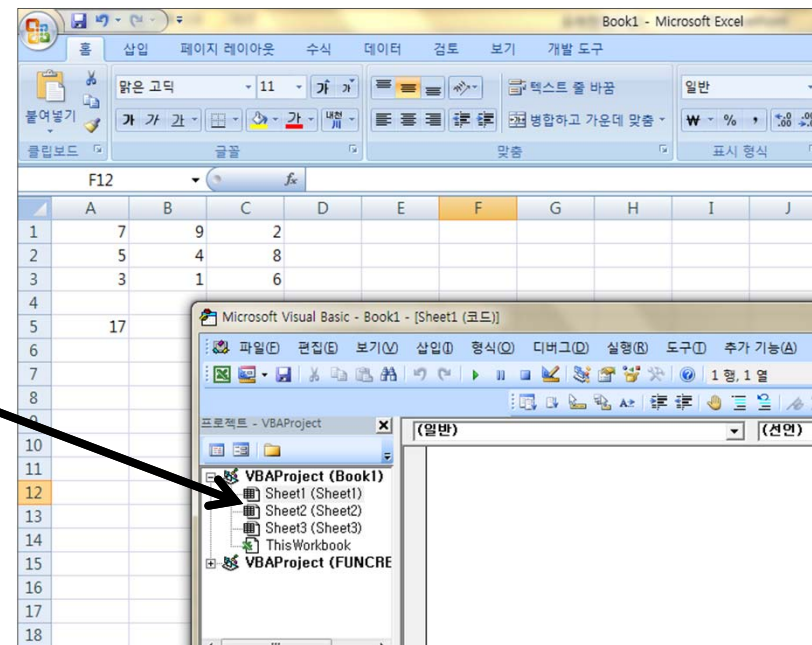
**Using VBA  
to Modify Contents of Cells**

# Referencing Excel Cells in VBA

The screenshot shows an Excel spreadsheet with the following data:

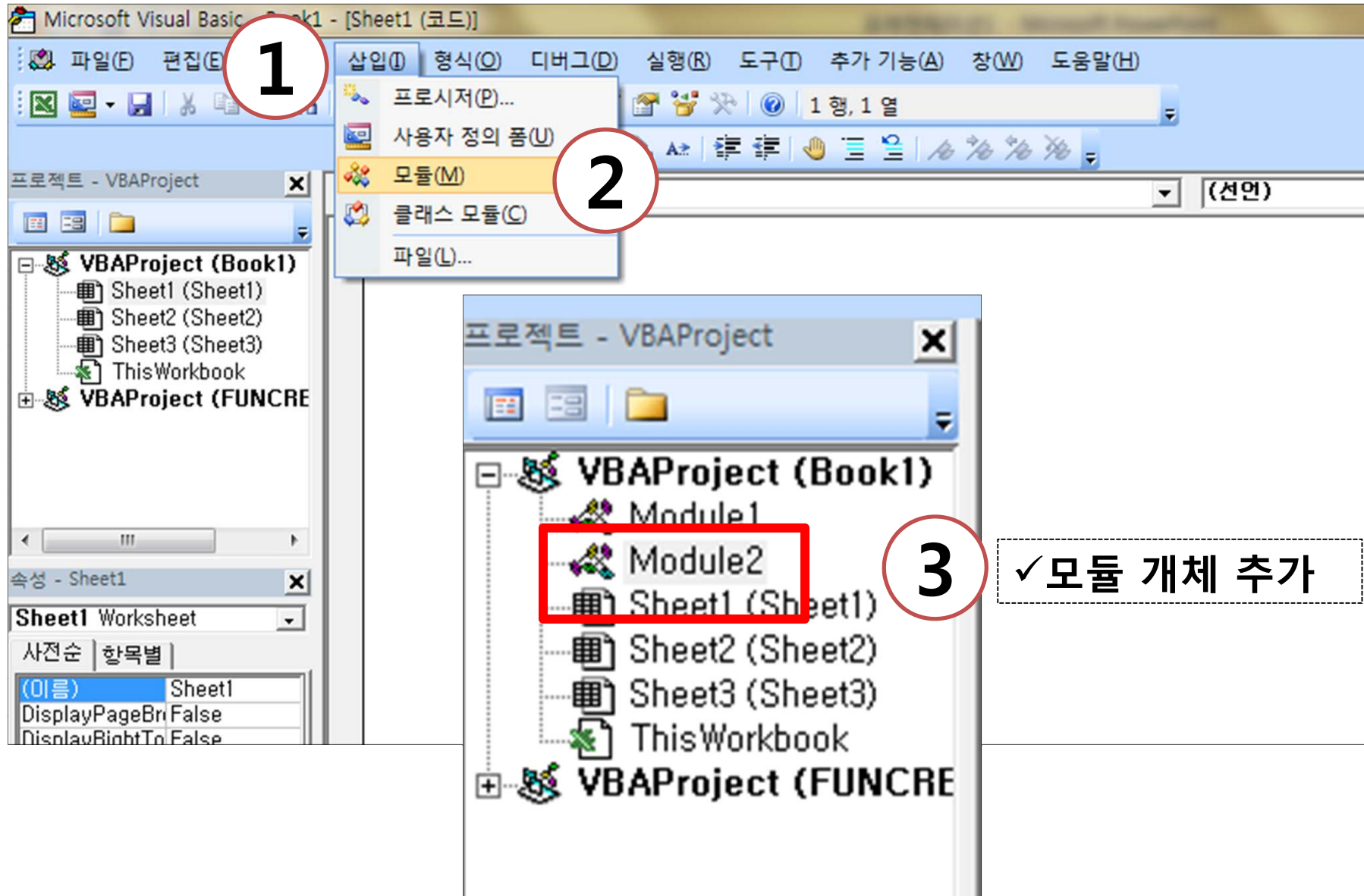
	A	B	C	D
1	7	8	12	
2	10	13	11	
3	6	9	14	
4				
5				
6				
7				

Below the spreadsheet, a callout shows the keyboard shortcut: **Alt + F11**.



✓ 엑셀창에서 **Alt+F11** 단축기  
Visual Basic 편집기 창호출.

# Using VBA to Modify Contents of Cells



# Using VBA to Modify Contents of Cells

---

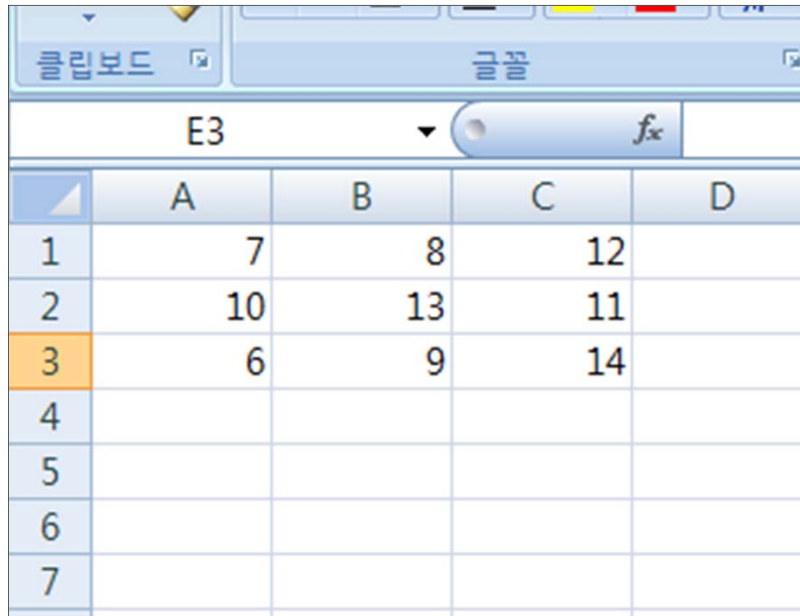
## ✓새 서브 프로시저 작성하기

```
Option Explicit
Dim c As Variant
Sub foreach( )
For Each c In Worksheets( "sheet1" ).Range( "A1:C3" )
c.Value = c.Value + 5
Next c
End Sub
```

```
Sub foreachif( )
For Each c In Selection
If c.Value = 15 Then
    c.Value = 9999
End If
Next c
End Sub
```

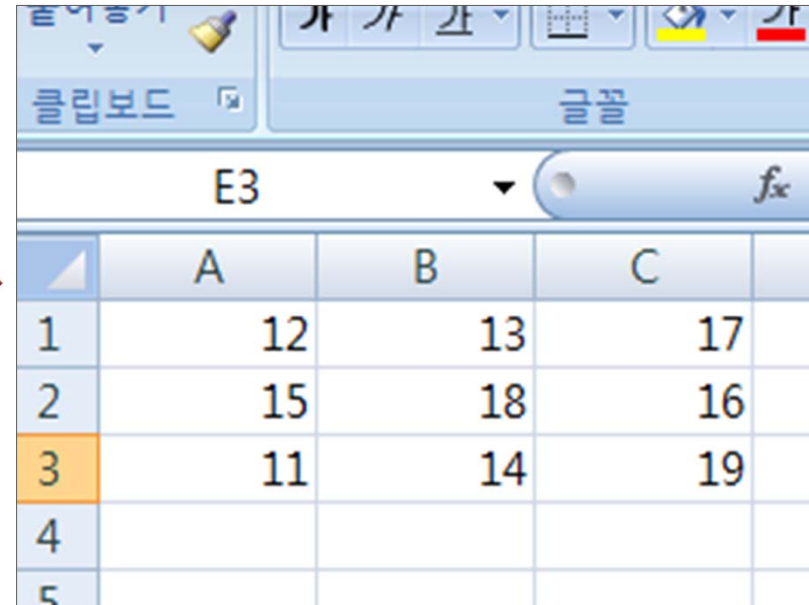
# Using VBA to Modify Contents of Cells

✓프로시저 실행하고 결과 확인하기



A screenshot of an Excel spreadsheet showing a grid of cells. The active cell is E3. The grid contains the following values:

	A	B	C	D
1	7	8	12	
2	10	13	11	
3	6	9	14	
4				
5				
6				
7				



A screenshot of an Excel spreadsheet showing the same grid of cells after VBA execution. The active cell is E3. The grid contains the following values:

	A	B	C	D
1	12	13	17	
2	15	18	16	
3	11	14	19	
4				
5				

```
Sub foreach()  
For Each c In Worksheets("sheet1").Range("A1:C3")  
c.Value = c.Value + 5  
Next c  
End Sub
```

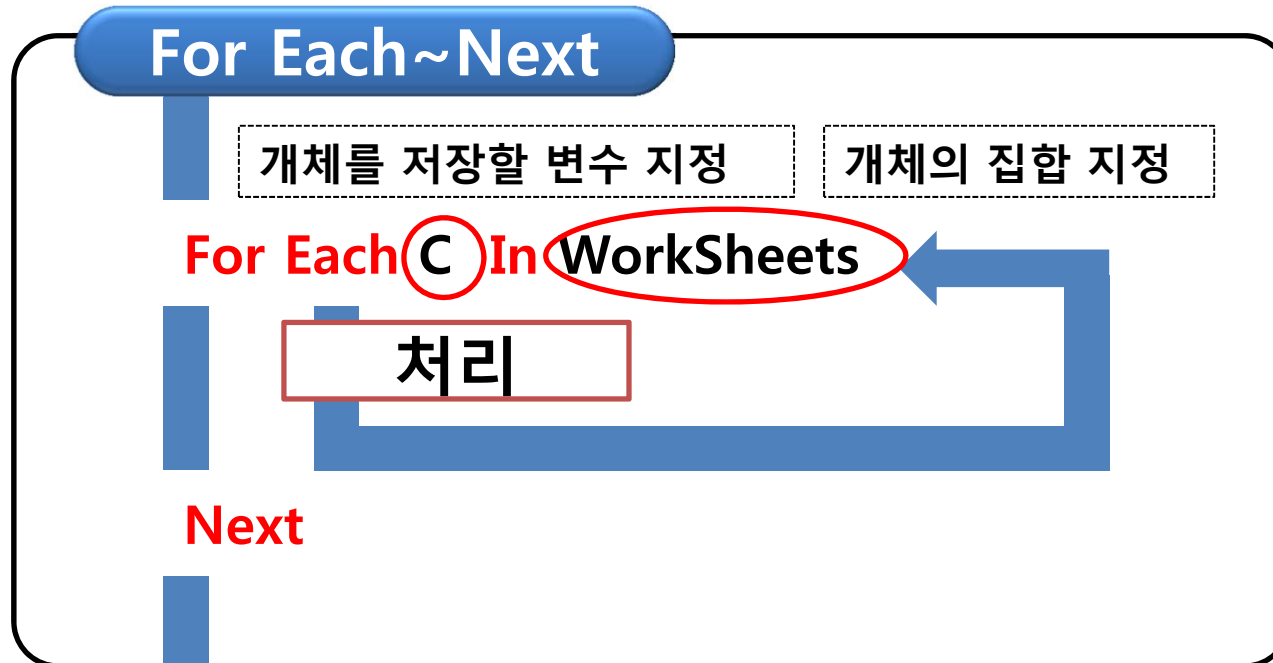


# Using VBA to Modify Contents of Cells

## ✓VBA 코드 이해하기

```
Sub foreach( )  
For Each c In Worksheets("sheet1").Range("A1:C3")  
c.Value = c.Value + 5  
Next c  
End Sub
```

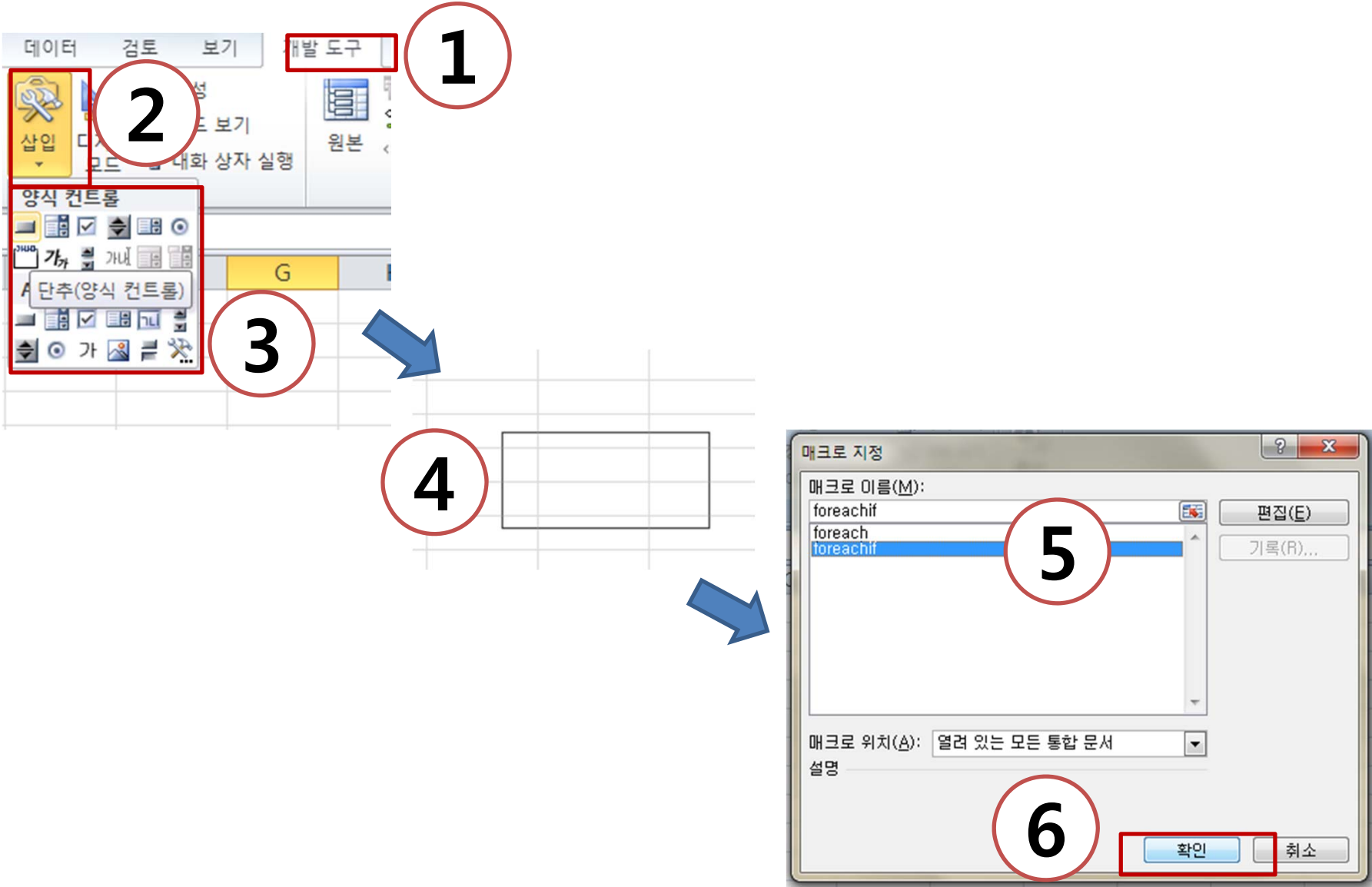
## ✓ For Each ~ Next 구문이란?



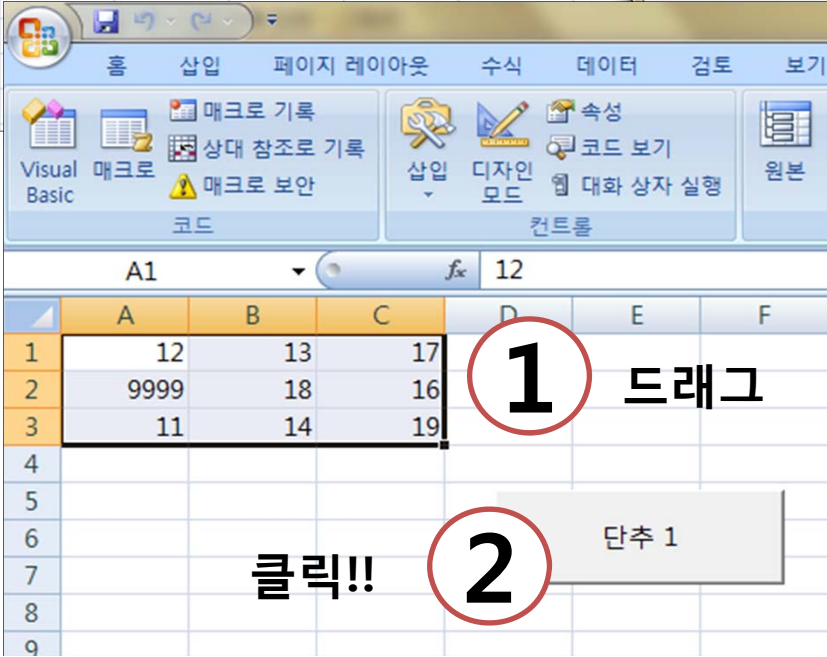
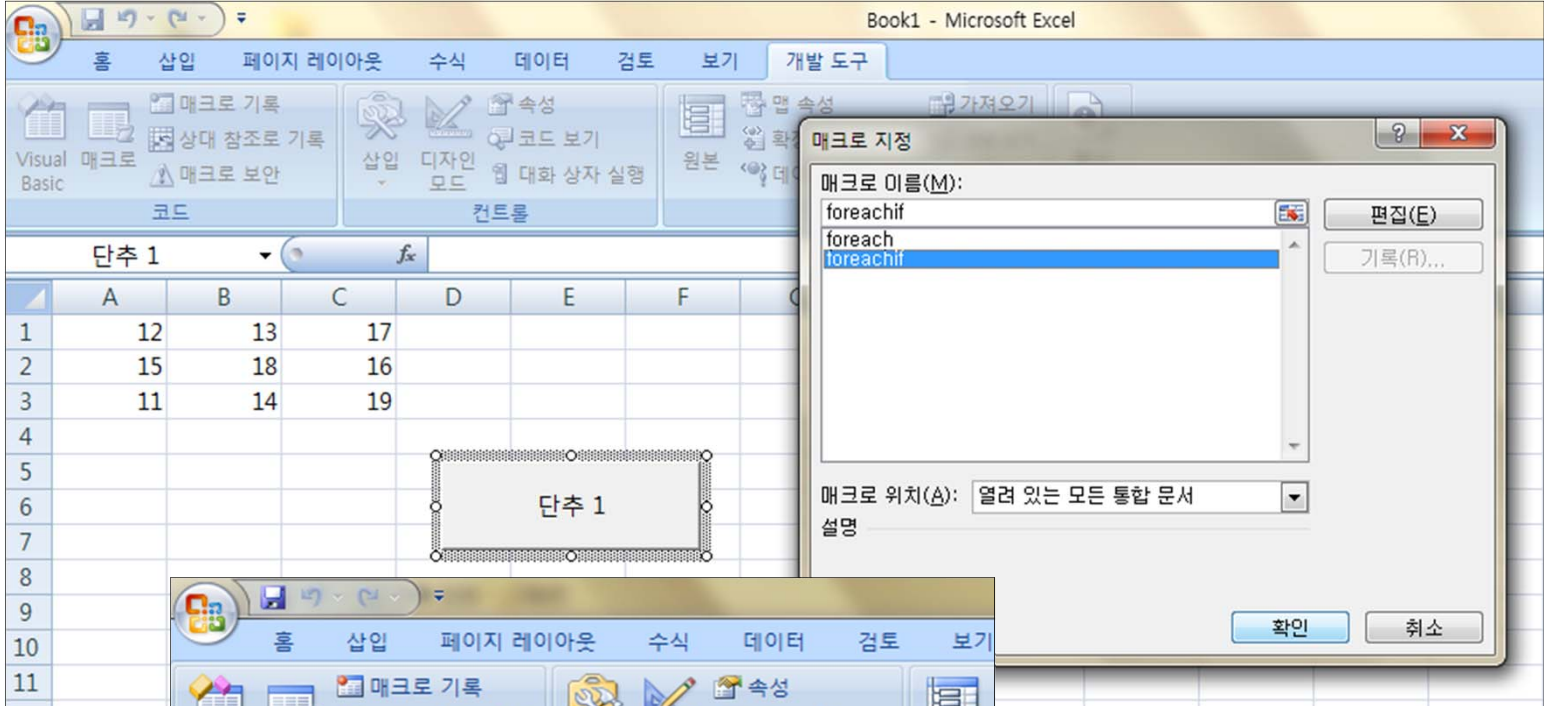
✓어떤 개체의 집합이 있고, 그 안의 모든 요소에 대해 처리를 수행할 때  
**For Each~Next** 구문 사용



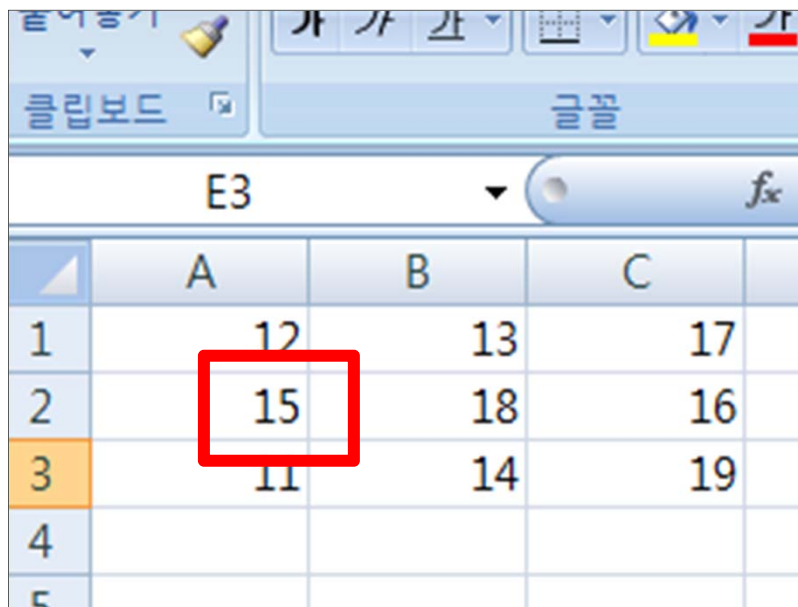
# Using VBA to Modify Contents of Cells



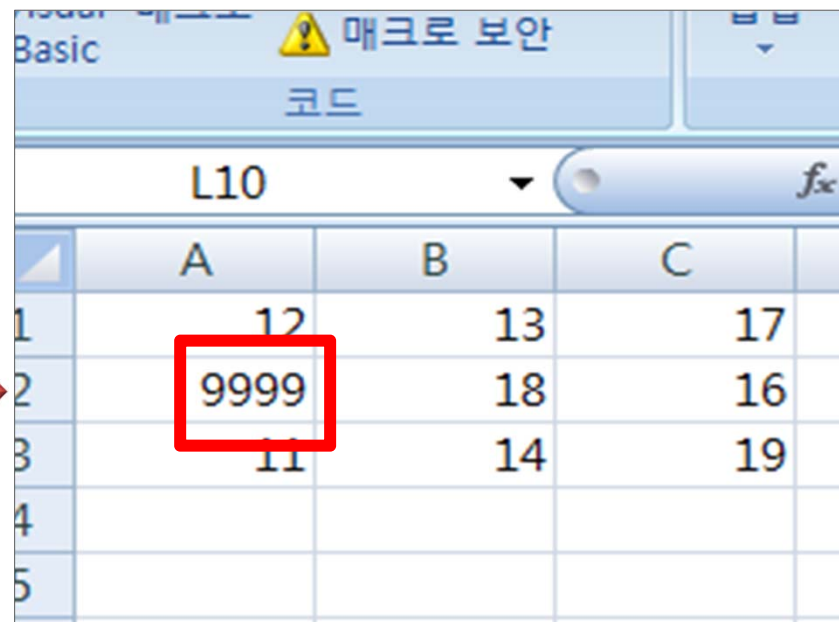
# Using VBA to Modify Contents of Cells



# Using VBA to Modify Contents of Cells



	A	B	C
1	12	13	17
2	15	18	16
3	11	14	19



	A	B	C
1	12	13	17
2	9999	18	16
3	11	14	19

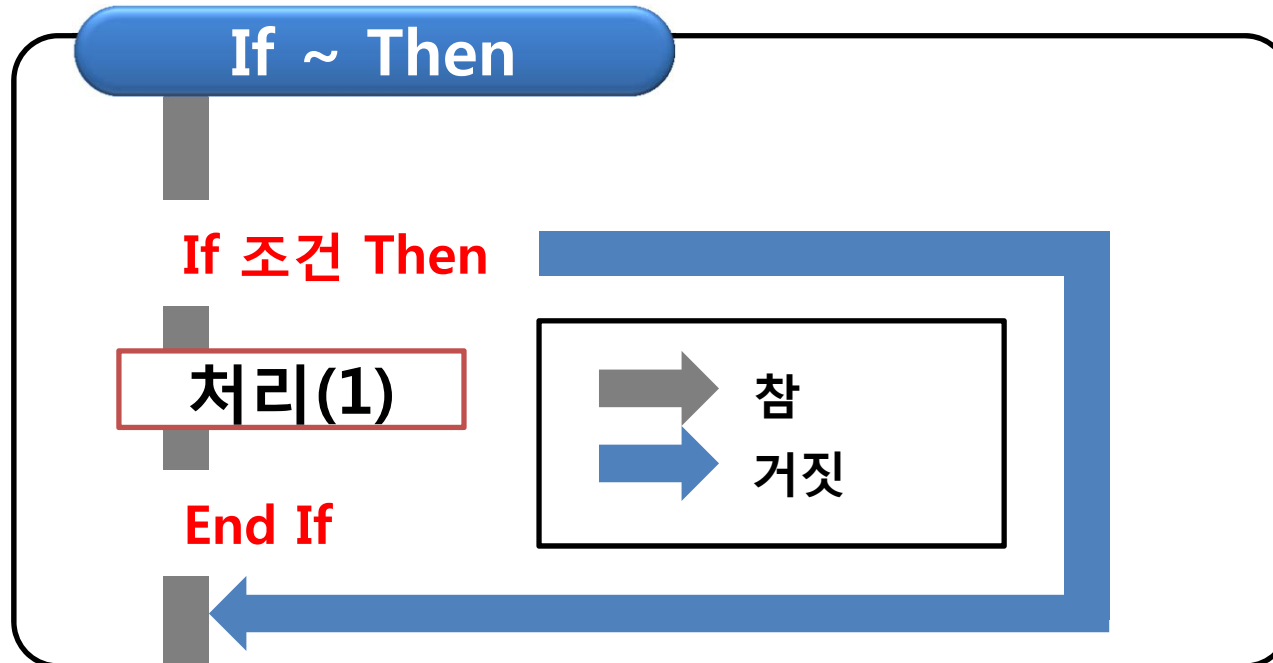
```
Sub foreachif()  
For Each c In Selection  
If c.Value = 15 Then  
    c.Value = 9999  
End If  
Next c  
End Sub
```

# Using VBA to Modify Contents of Cells

✓VBA 코드 이해하기

```
Sub foreachif()  
For Each c In Selection  
If c.Value = 15 Then  
    c.Value = 9999  
End If  
Next c  
End Sub
```

✓ If ~ Then구문이란?



✓조건이 성립할 때( 조건식이 참)는 처리(1)을 수행. 성립하지 않은 경우 아무것도 수행하지 않음.

# Using VBA to Modify Contents of Cells

✓VBA 코드 이해하기

```
Sub foreachif()  
For Each c In Selection  
If c.Value = 15 Then  
    c.Value = 9999  
End If  
Next c  
End Sub
```

IF c.Value =15  
c.Value = 9999

변수                  값  
                        ↪ 대입

✓ c.Value가 15인 경우 9999대입.  
조건이 성립하지 않는 경우는 그대로 두기.

# Summary



# Summary

## Referencing Excel Cells in VBA

➤ a = ? ("A5"). Value

✓ Range 개체는 셀 범위를 나타내는 개체.

✓ A5 cell 의미

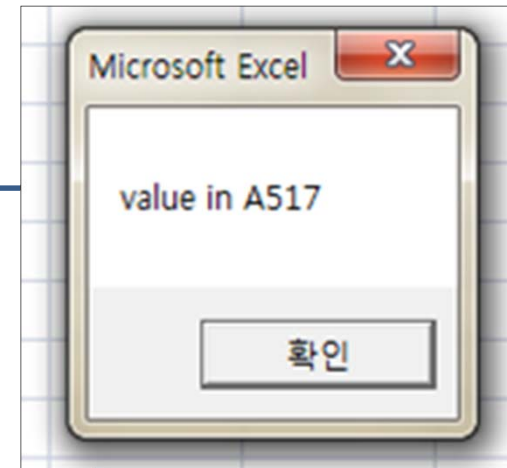
✓ 속성명

➤ ? ("Value in A5") & a

✓ MsgBox 함수는 사용자에게 메시지 창을 표시할 때 사용

✓ MsgBox(메시지 내용)

✓ 지정한 a의 값



- ✓ ? 함수: 사용자의 **입력 값을 받아 처리**하고자 할 때 사용.
- ✓ MsgBox 함수: 사용자에게 **전달한 내용을 메시지창에 표시**. 특정상황에 **사용자의 선택이 필요**한 경우 사용.

# Summary

## Importing Data from Excel to VBA

The screenshot shows an Excel spreadsheet with the following data:

	A	B	C	D	E	F	G
1	7	9	2				
2	5	4	8				
3	3	1	6				
4							
5	17			7	9	2	
6				5	4	8	
7				3	1	6	
8							
9							

Annotations in the image:

- A blue box highlights the range A1:C5 with the text: `✓ a = Range ("A1:C5")`  
A1에서 C5까지  
a로 지정
- A red box highlights the range D5:F7.
- A blue box contains the text: `✓ Range("D5:F7").Value=a`  
D5에서 F7까지 a의 값이 입력

```
Sub GetArray2( )  
a = Range("A1:C5")  
Range("D5:F7").Value = a  
End Sub
```

# Summary

- Using VBA to Modify Contents of Cells

