

# 엑셀과 VBA



# 차 례

<b>제 I 편 Excel</b>	<b>1</b>
<b>제 1 장 엑셀이란 무엇인가?</b>	<b>3</b>
제 1.1 절 엑셀 시작과 종료 . . . . .	4
제 1.2 절 엑셀 명칭 . . . . .	4
제 1.3 절 도움말 사용 . . . . .	7
<b>제 2 장 통합문서(워크북;Workbook)</b>	<b>9</b>
제 2.1 절 엑셀 파일 . . . . .	9
제 2.2 절 파일 열기 . . . . .	10
제 2.3 절 새로 만들기 . . . . .	11
제 2.4 절 파일 저장 . . . . .	12
제 2.5 절 시트(sheet) . . . . .	12
제 2.6 절 인쇄 . . . . .	15
제 2.7 절 페이지 레이아웃 . . . . .	15
제 2.8 절 Excel 옵션 . . . . .	16
2.8.1 일반 . . . . .	17
2.8.2 수식 . . . . .	18
2.8.3 언어 교정 . . . . .	19
2.8.4 저장 . . . . .	19
2.8.5 언어 . . . . .	20
2.8.6 고급 . . . . .	20

---

2.8.7 리본 사용자 지정 . . . . .	25
2.8.8 빠른 실행 도구 모음 . . . . .	25
2.8.9 추가기능 . . . . .	25
2.8.10 보안 센터 . . . . .	25
제 2.9 절 이미지 편집 . . . . .	28
2.9.1 이미지 일부분 저장하기 . . . . .	28
2.9.2 이미지 배경색 지우기 . . . . .	28
2.9.3 이미지 흐리게 하기 . . . . .	28
<b>제 3 장 셀(cell)에 대하여</b>	<b>31</b>
제 3.1 절 셀 번지 . . . . .	31
제 3.2 절 셀 입력, 셀 편집 . . . . .	32
제 3.3 절 셀 이동 . . . . .	34
제 3.4 절 셀 선택 . . . . .	34
제 3.5 절 이름상자 . . . . .	35
제 3.6 절 셀 서식 . . . . .	35
3.6.1 표시형식 . . . . .	36
3.6.2 맞춤 . . . . .	45
3.6.3 글꼴 . . . . .	46
3.6.4 테두리 . . . . .	47
3.6.5 채우기 . . . . .	47
3.6.6 셀 보호 . . . . .	47
제 3.7 절 조건부 서식 . . . . .	48
제 3.8 절 셀에 메모 사용 . . . . .	51
제 3.9 절 셀에 하이퍼링크(hyperlink) 사용 . . . . .	52
제 3.10 절 셀에 수식 입력과 연산 . . . . .	53
3.10.1 산술 연산자 . . . . .	54
3.10.2 비교 연산자 . . . . .	54
3.10.3 텍스트 연산자 . . . . .	55

3.10.4 참조 연산자 . . . . .	55
제 3.11 절 셀 채우기 . . . . .	56
3.11.1 숫자, 날짜, 기타 항목 연속 데이터 채우기 . . . . .	56
3.11.2 사용자 지정 목록 채우기 . . . . .	59
제 3.12 절 데이터 이동과 복사 . . . . .	59
제 3.13 절 셀 참조 . . . . .	61
제 3.14 절 시트 참조 . . . . .	63
제 3.15 절 통합문서 참조 . . . . .	64
<b>제 4 장 함수 사용하기</b>	<b>69</b>
제 4.1 절 데이터 불러 오기 . . . . .	69
제 4.2 절 함수 사용방법 . . . . .	73
4.2.1 함수를 직접 입력하는 방법 . . . . .	73
4.2.2 함수 마법사를 사용하는 방법 . . . . .	75
제 4.3 절 함수의 종류 . . . . .	75
제 4.4 절 논리 함수 . . . . .	76
4.4.1 TRUE, FALSE 함수 . . . . .	77
4.4.2 AND 함수 . . . . .	77
4.4.3 OR 함수 . . . . .	78
4.4.4 NOT 함수 . . . . .	78
4.4.5 IF 함수 . . . . .	79
4.4.6 중첩 함수 . . . . .	80
4.4.7 COUNT 함수 . . . . .	81
4.4.8 COUNTIF 함수 . . . . .	82
4.4.9 COUNTIFS 함수 . . . . .	83
4.4.10 SUM 함수 . . . . .	84
4.4.11 SUMIF 함수 . . . . .	84
4.4.12 SUMIFS 함수 . . . . .	85
4.4.13 AVERAGE 함수 . . . . .	86

---

4.4.14 AVERAGEIF 함수 . . . . .	87
4.4.15 AVERAGEIFS 함수 . . . . .	88
제 4.5 절 날짜/시간 함수 . . . . .	89
4.5.1 TODAY 함수 . . . . .	90
4.5.2 NOW 함수 . . . . .	90
4.5.3 YEAR 함수 . . . . .	90
4.5.4 MONTH 함수 . . . . .	91
4.5.5 DAY 함수 . . . . .	91
4.5.6 DATE 함수 . . . . .	92
4.5.7 HOUR 함수 . . . . .	92
4.5.8 MINUTE 함수 . . . . .	93
4.5.9 SECOND 함수 . . . . .	94
4.5.10 TIME 함수 . . . . .	94
4.5.11 WEEKDAY 함수 . . . . .	95
4.5.12 WORKDAY 함수 . . . . .	96
4.5.13 NETWORKDAYS 함수 . . . . .	97
제 4.6 절 IS 함수 . . . . .	97
4.6.1 ISBLANK 함수 . . . . .	98
4.6.2 ISERR 함수 . . . . .	98
4.6.3 ISERROR 함수 . . . . .	98
4.6.4 ISEVEN 함수 . . . . .	99
4.6.5 ISLOGICAL 함수 . . . . .	99
4.6.6 ISNA 함수 . . . . .	99
4.6.7 ISNONTEXT 함수 . . . . .	99
4.6.8 ISNUMBER 함수 . . . . .	99
4.6.9 ISODD 함수 . . . . .	99
4.6.10 ISREF 함수 . . . . .	99
4.6.11 ISTEXT 함수 . . . . .	99
제 4.7 절 텍스트 관련 함수 . . . . .	100

4.7.1	CODE 함수 . . . . .	100
4.7.2	CHAR 함수 . . . . .	100
4.7.3	EXACT 함수 . . . . .	100
4.7.4	FIXED 함수 . . . . .	101
4.7.5	LEFT 함수 . . . . .	102
4.7.6	LEN 함수 . . . . .	102
4.7.7	LOWER 함수 . . . . .	102
4.7.8	MID 함수 . . . . .	103
4.7.9	PROPER 함수 . . . . .	103
4.7.10	REPLACE 함수 . . . . .	104
4.7.11	REPT 함수 . . . . .	105
4.7.12	RIGHT 함수 . . . . .	105
4.7.13	TRIM 함수 . . . . .	106
4.7.14	UPPER 함수 . . . . .	106
4.7.15	VALUE 함수 . . . . .	106
제 4.8	절 행렬 . . . . .	107
4.8.1	전치행렬 . . . . .	107
4.8.2	두 행렬의 곱 . . . . .	108
4.8.3	행렬식 . . . . .	110
4.8.4	역행렬 . . . . .	111
제 4.9	절 수학·삼각 함수 . . . . .	113
4.9.1	삼각 함수에서 유용한 함수 . . . . .	113
4.9.2	SIN 함수 . . . . .	114
4.9.3	ASIN 함수 . . . . .	114
4.9.4	COS 함수 . . . . .	115
4.9.5	ACOS 함수 . . . . .	115
4.9.6	TAN 함수 . . . . .	116
4.9.7	ATAN 함수 . . . . .	116
4.9.8	COMBIN 함수 . . . . .	116

---

4.9.9 EXP 함수 . . . . .	117
4.9.10 GCD 함수 . . . . .	117
4.9.11 LCM 함수 . . . . .	117
4.9.12 INT 함수 . . . . .	118
4.9.13 로그 함수 . . . . .	118
4.9.14 자릿수 조정 함수 . . . . .	119
4.9.15 FACT 함수 . . . . .	120
4.9.16 SQRT 함수 . . . . .	120
4.9.17 합계 함수 . . . . .	121
제 4.10 절 통계 함수 . . . . .	122
4.10.1 대푯값에 대한 함수 . . . . .	122
4.10.2 자료의 흩어짐(dispersion)에 대한 함수 . . . . .	125
4.10.3 기술 통계량(descriptive statistics) . . . . .	126
4.10.4 두 변수의 관련성을 나타내는 함수 . . . . .	128
4.10.5 개수 함수 . . . . .	129
4.10.6 분포함수 . . . . .	131
4.10.7 통계적 추론 함수 . . . . .	146
제 4.11 절 찾기/참조 함수 . . . . .	151
4.11.1 순위 함수 . . . . .	151
4.11.2 찾기 함수 . . . . .	152
4.11.3 MATCH, INDEX, CHOOSE 함수 . . . . .	154
제 4.12 절 재무 함수 . . . . .	156
4.12.1 FV 함수 . . . . .	156
4.12.2 PMT 함수 . . . . .	157
제 4.13 절 사용자가 만든 함수 . . . . .	158
<b>제 5 장 차트 그리기</b>	<b>163</b>
제 5.1 절 차트 . . . . .	164
제 5.2 절 원 그래프 . . . . .	165

제 5.3 절 막대 그래프 . . . . .	166
제 5.4 절 산점도와 추세선 . . . . .	169
제 5.5 절 곡선 그래프와 산점도 . . . . .	174
제 5.6 절 도수분포표 . . . . .	177
제 5.7 절 히스토그램 . . . . .	179
제 5.8 절 상자 수염 그림 . . . . .	181
제 5.9 절 이중 축 혼합 차트 . . . . .	183
<b>제 6 장 데이터베이스</b>	<b>188</b>
제 6.1 절 정렬 . . . . .	188
제 6.2 절 필터 . . . . .	190
제 6.3 절 유효성 검사 . . . . .	194
제 6.4 절 부분합 . . . . .	196
제 6.5 절 목표값 찾기 . . . . .	199
제 6.6 절 해 찾기 . . . . .	202
제 6.7 절 시나리오 작성 . . . . .	204
<b>제 II 편 Visial Basic for Applications</b>	<b>207</b>
<b>제 7 장 매크로(Macro)</b>	<b>209</b>
제 7.1 절 매크로 기록과 실행 . . . . .	210
7.1.1 매크로 기록 . . . . .	210
7.1.2 매크로 실행 . . . . .	214
제 7.2 절 VBA . . . . .	217
연습문제 . . . . .	218
<b>제 8 장 컨트롤</b>	<b>219</b>
제 8.1 절 ActiveX 컨트롤 . . . . .	219
8.1.1 명령 단추(CommandButton) . . . . .	219
8.1.2 콤보 상자(ComboBox) . . . . .	226

8.1.3 확인란(CheckBox) . . . . .	231
8.1.4 목록 상자(ListBox) . . . . .	236
8.1.5 옵션 단추(OptionButton) . . . . .	240
8.1.6 텍스트 상자(TextBox) . . . . .	245
8.1.7 스크롤 막대(ScrollBar) . . . . .	248
8.1.8 스핀 단추(SpinButton) . . . . .	250
8.1.9 ActiveX 컨트롤을 이용한 인사관리 . . . . .	252
8.1.10 ActiveX 컨트롤을 이용한 자동필터 . . . . .	258
<b>제 8.2 절 양식 컨트롤 . . . . .</b>	<b>263</b>
8.2.1 명령 단추(CommandButton) . . . . .	263
8.2.2 콤보 상자(ComboBox) . . . . .	264
8.2.3 확인란(CheckBox) . . . . .	265
8.2.4 목록 상자(ListBox) . . . . .	266
8.2.5 옵션 단추(OptionButton) . . . . .	267
8.2.6 양식 컨트롤을 이용한 고급필터 . . . . .	268
<b>연습문제 . . . . .</b>	<b>272</b>
<b>제 9 장 사용자 정의 폼(User Form) . . . . .</b>	<b>275</b>
<b>제 9.1 절 사용자 정의 폼 작성과 실행 . . . . .</b>	<b>275</b>
<b>제 9.2 절 컨트롤 도구 상자 . . . . .</b>	<b>278</b>
<b>제 9.3 절 사용자 정의 폼 – 허수, 짹수 판별 . . . . .</b>	<b>284</b>
<b>제 9.4 절 사용자 정의 폼 – 월드컵 축구 프로그램 . . . . .</b>	<b>290</b>
<b>제 9.5 절 사용자 정의 폼 – 문자 입·출력 프로그램 . . . . .</b>	<b>295</b>
<b>제 10 장 비주얼 베이직 입문 . . . . .</b>	<b>301</b>
<b>제 10.1 절 Visual Basic 시작하기 . . . . .</b>	<b>301</b>
<b>제 10.2 절 Visual Basic 만들고 실행하기 . . . . .</b>	<b>302</b>
<b>제 10.3 절 Visual Basic 구성 . . . . .</b>	<b>303</b>
<b>제 10.4 절 엑셀과 관련된 개체, 메소드, 속성 . . . . .</b>	<b>304</b>
<b>10.4.1 통합문서 개체 . . . . .</b>	<b>305</b>

10.4.2 시트 개체 . . . . .	308
10.4.3 셀 개체 . . . . .	309
제 10.5 절 VBA에서 제공하는 함수 사용방법 . . . . .	319
10.5.1 MsgBox 함수 . . . . .	319
10.5.2 InputBox 함수 . . . . .	323
연습문제 . . . . .	326
<b>제 11 장 VBA 프로그래밍 기법</b>	<b>328</b>
제 11.1 절 변수와 상수 . . . . .	328
11.1.1 변수 선언 . . . . .	328
11.1.2 상수 선언 . . . . .	331
11.1.3 배열 선언 . . . . .	331
제 11.2 절 변수에 값 배정 . . . . .	331
제 11.3 절 Visual Basic 식별자 규칙 . . . . .	333
제 11.4 절 Visual Basic 문 작성 . . . . .	333
11.4.1 여러 줄에 계속되는 문 . . . . .	334
11.4.2 여러 문을 한 줄에 사용 . . . . .	334
11.4.3 주석 추가 . . . . .	335
11.4.4 구문 오류 검사 . . . . .	335
제 11.5 절 배열 사용 . . . . .	336
11.5.1 하한값 변경 . . . . .	337
11.5.2 배열에 값 저장 . . . . .	338
11.5.3 다차원 배열 사용 . . . . .	338
11.5.4 동적 배열 선언 . . . . .	339
제 11.6 절 제어문 . . . . .	341
11.6.1 If-Then-ElseIf-Then-Else 문 . . . . .	341
11.6.2 Select-Case 문 . . . . .	343
제 11.7 절 반복문 . . . . .	345
11.7.1 Do-Loop 문 . . . . .	345

---

11.7.2 For-Each-Next 문 . . . . .	348
11.7.3 For-Next 문 . . . . .	349
11.7.4 With 문 . . . . .	351
제 11.8 절 연산자 . . . . .	352
11.8.1 산술 연산자 . . . . .	353
11.8.2 비교 연산자 . . . . .	353
11.8.3 문자열 연결 연산자 . . . . .	354
11.8.4 논리 연산자 . . . . .	355
<b>제 12 장 VBA 프로시저</b>	<b>357</b>
제 12.1 절 Sub 프로시저 . . . . .	357
제 12.2 절 Function 프로시저 . . . . .	360
제 12.3 절 Property 프로시저 . . . . .	361
제 12.4 절 인수 전달 방법 . . . . .	367
12.4.1 ByVal 인수 전달 . . . . .	367
12.4.2 ByRef 인수 전달 . . . . .	368
제 12.5 절 이벤트 프로시저 . . . . .	369
12.5.1 Workbooks 개체와 이벤트 프로시저 . . . . .	370
12.5.2 Worksheets 개체와 이벤트 프로시저 . . . . .	372
제 12.6 절 VBA 프로젝트에 암호 설정 . . . . .	376
<b>찾아 보기</b>	<b>379</b>

# 제 I 편

## Excel



# 제 1 장

## 엑셀이란 무엇인가?

엑셀은 마이크로소프트(이하 MS)사에서 제작한 오피스에 포함된 응용프로그램의 한 종류로 스프레드시트<sup>1</sup>이자 분석프로그램이다. 여기서 엑셀은 버전 MicroSoft Office 2016을 기준으로 소개한다. 엑셀의 특징은

- ◉ 워크시트(WorkSheet<sup>2</sup>)를 이용한 작업.
- ◉ 차트제작
- ◉ 편리한 수식계산
- ◉ 함수를 이용한 자료분석
- ◉ 독자적인 응용프로그램 개발기능(매크로, macro)
- ◉ 엑셀을 웹에 연결
- ◉ 그림이나 클립아트 등 다양한 개체를 삽입
- ◉ 데이터베이스 기능

이다.

---

<sup>1</sup>일상 업무에 많이 발생하는 여러 가지 도표 형태의 양식으로 계산하는 사무 업무를 자동으로 할 수 있는 프로그램

<sup>2</sup>엑셀에서는 스프레드시트를 워크시트로 부른다.

## 제 1.1 절 엑셀 시작과 종료

엑셀을 시작하려면 당연히 엑셀 응용프로그램이 설치되어 있어야 한다. 엑셀은 MS 오피스 제품군에 포함되어 있으므로 MS 오피스를 설치할 때 엑셀을 포함하여 설치하면 엑셀 응용프로그램을 사용할 수 있다. 엑셀의 시작은

- 바탕화면에 Microsoft Excel 단축아이콘이 있으면 이것을 더블클릭하거나
- 시작→프로그램→Microsoft Excel을 누르면

실행할 수 있다. 종료하는 방법은 엑셀 프로그램이 활성화 상태에서

- 창 종료버튼을 누르거나
- ALT + F4 키를 누르거나
- 엑셀창에서 파일→닫기 선택하면

엑셀프로그램을 종료할 수 있다.

## 제 1.2 절 엑셀 명칭

엑셀의 명칭에 대하여 알아보자. 엑셀을 실행하면 그림 1.2.1과 같이 창이 열린다. 각 부분 명칭은

- 제목표시줄 : 윈도에서 프로그램을 실행하면 실행프로그램이름과 파일명이 나타나는 부분으로 프로그램이 활성화 상태이면 파란색, 비활성화 상태이면 회색이다.
- Office 단추 : 파일 열기, 저장, 다른이름으로 저장, 인쇄, Excel 옵션 등 엑셀에서 자주 사용하는 메뉴와 엑셀의 환경을 설정하는 메뉴들이 포함되어 있다.

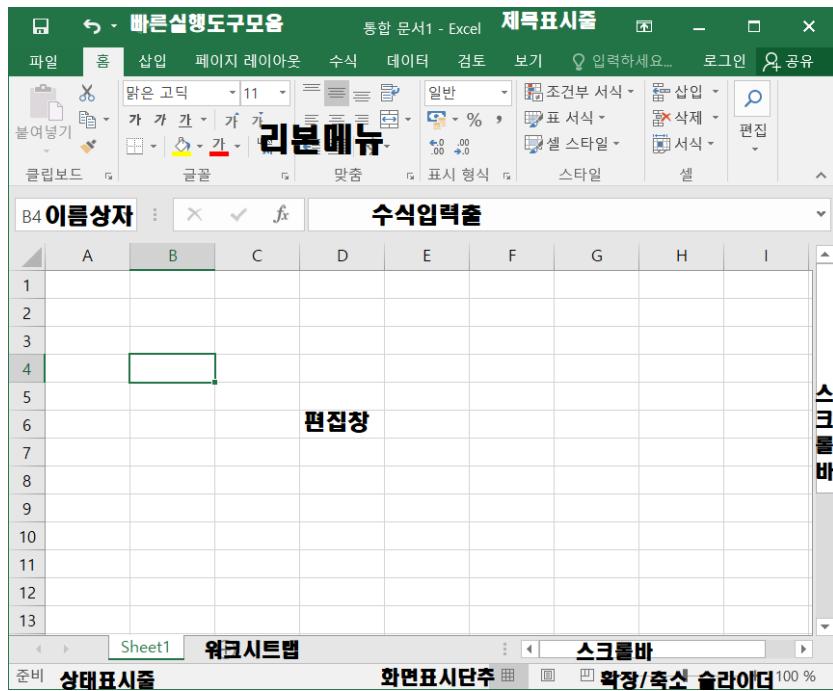


그림 1.2.1: 엑셀 명칭

- ① **빠른실행도구모음** : 자주 실행하는 메뉴들을 모아 놓은 것으로 파일 열기, 저장 등이 포함되어 있으며, 다른 메뉴도 추가할 수 있다.
- ② **리본메뉴** : 각 범주에 속하는 메뉴들을 모아 놓은 것으로 다른 프로그램에서는 메뉴나 도구모음에 해당된다. 리본메뉴는 기본값이 전체 리본메뉴를 볼 수 있도록 설정되어 있으며 최소화 상태의 메뉴 상태로 변경하려면 빠른 실행 도구모음에서 → 리본목록 최소화를 클릭한다. 반대로 전체 리본 메뉴를 보이도록 하려면 빠른 실행 도구모음에서 → 리본목록 최소화 체크를 해제한다.
- ③ **편집창** : 현재 편집중인 창으로 자료의 입력, 편집, 삭제를 할 수 있는 스프레드시트로 구성되어 있으며 배열은 열과 행에 셀로 구성되어 있다.

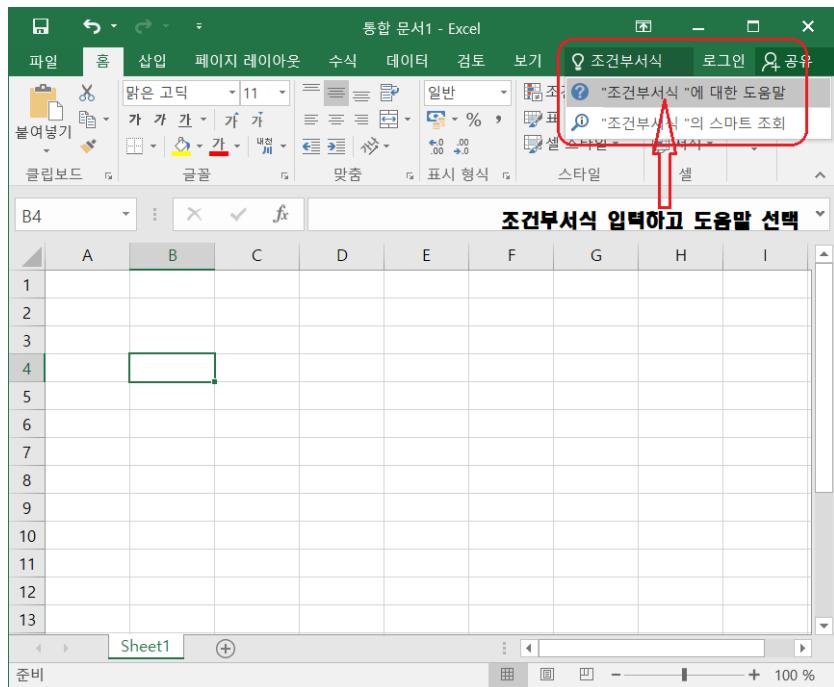


그림 1.2.2: 도움말 입력

- 상태표시줄 : 현재 엑셀의 상태를 보여주는 것으로 처음 시작하면 준비 상태이다.
- 셀(cell) : 엑셀에서 작업하는 공간으로 행의 개수는  $2^{20} = 1,048,576$ 이고 1 2 … 1,048,576으로 나타내며 열의 개수는  $2^{14} = 16,348$ 이고 A B … Z AA AB … ZY ZZ AAA AAB … XFC XFD로 나타낸다.
- 시트(sheet) : 셀이 모여서 이루어진 것으로 최대 255개까지 만들 수 있다.
- 화면표시단추 : 편집중인 화면을 기본, 페이지 레이아웃, 페이지 나누기 미리보기 중 하나를 선택하여 화면표시모드를 변경할 수 있다.
- 스크롤바(scroll bar) : 엑셀에서 보여지고 있는 공간이 전체 데이터의

어느정도인지 알려주는 정보이며 다른 영역으로 이동하는 기능도 제공한다.

- ◉ 확대/축소 슬라이더 : 현재 편집중인 워크시트의 확대/축소하여 화면에 보여주는 기능을 제공한다.
- ◉ 이름상자 : 셀의 위치 정보를 보여 주며, 특정 셀들의 영역을 선택한 후 이름을 설정할 수 있다. 또한 특정 셀로 이동하려면 셀 주소를 입력하고 엔터키를 누른다.
- ◉ 수식입력줄 : 이 곳에서 연산자와 함수를 사용하여 계산한다.

이다.

### 제 1.3 절     도움말 사용

응용프로그램의 사용법은 응용프로그램 안에 있는 도움말이 가장 유용한 정보를 준다. 도움말은 응용프로그램의 사용법 뿐만 아니라 예제도 제공하기 때문에 사용방법 습득에 많은 도움이 된다.

엑셀에서 도움말을 사용하려면

- ◉ 엑셀 프로그램이 활성화 상태에서 키보드의 F1을 누른다.

만일 엑셀 사용자가 도움말을 활용하여 조건부서식에 대하여 알아보려면 그림 1.2.2의 검색 입력창에 조건부서식을 입력하고 검색 버튼을 누른다. 그림 1.2.2의 화면에 조건부서식에 대하여 검색한 결과를 별도의 창에서 확인할 수 있다.



## 제 2 장

### 통합문서(워크북;Workbook)

통합문서(워크북, workbook)는 엑셀에서 자료를 입력한 후 저장하여 만든 파일로 한 개 이상 시트(sheet)로 구성되어 있다. 시트(sheet)는 엑셀에서 최소 저장 단위인 셀(cell)이 모여서 이루어진 데이터들의 집합체이다.

#### 제 2.1 절 엑셀 파일

엑셀에서 작업한 내용은 여러 가지 파일 형태로 저장하며 내용은 다음과 같다.

- ◉ Excel 통합 문서 : 일반적인 엑셀 파일이다. 파일 확장자는 .xlsx
- ◉ Excel 매크로 사용 통합 문서 : 매크로(macro)가 포함된 엑셀 파일을 의미한다. 파일 확장자는 .xlsm이다. 매크로는 여러 엑셀 명령을 한 번에 실행할 수 있게 만든 도구이다.
- ◉ Excel 바이너리 통합 문서 : 엑셀 파일을 신속하게 로드하고 저장할 수 있도록 바이너리 형태로 저장한다. 파일 확장자는 .xlsb
- ◉ Excel 97 - 2003 통합 문서 : 이전 엑셀 파일 버전으로 저장한다. 파일 확장자는 .xls

- ◉ OpenDocument 스프레드시트 : OpenDocument 파일 형태(확장자 .ods)로 저장한다. OpenDocument는 XML 기반 파일 형태로 스프레드시트, 차트, 프리젠테이션을 전자 문서의 형태로 표현한다.
- ◉ PDF 또는 XPS : PDF 파일이나 XPS(XML Paper Specification) 파일로 저장한다. XPS는 문서 서식이 유지되고 파일 공유가 가능한 고정된 레이아웃의 전자파일이다. 파일 확장자는 .xml
- ◉ 다른 형식 : 다른 이름으로 저장 대화 상자에서 열어서 사용 가능한 모든 파일 형식이다.

## 제 2.2 절 파일 열기

이미 저장된 엑셀 파일은 파일 → 열기 메뉴 실행으로 그 내용을 확인한다. 이 때 파일 형식이 엑셀 파일이 아니면, 파일 형식에서 적합한 파일 종류를 선택한다. 그림 2.2.1은 파일 열기 대화상자이다.

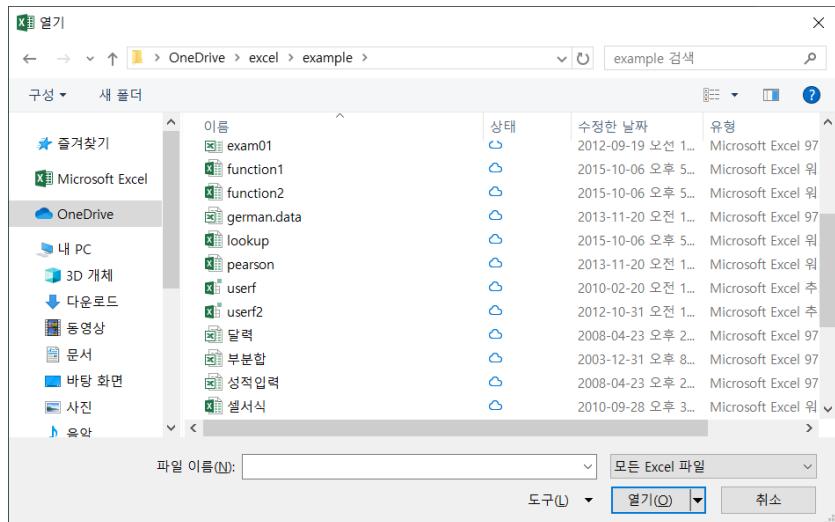


그림 2.2.1: 파일 열기

## 제 2.3 절 새로 만들기

새로 만들기 메뉴는

- ◉ 새 통합문서를 만들거나
- ◉ 사용자가 직접 만든 서식파일이나 최근에 사용한 서식파일, MS에서 이미 만들어진 서식파일 등을 이용할 수 있다.

새로 만들기 실행은 파일 → 새로 만들기 메뉴를 클릭한다. 이 메뉴는 이미 만들어진 서식파일을 열어 새로운 통합문서를 만들 수 있다. 그럼 2.3.1은 새로 만들기에서 사용 가능한 서식 파일들이다.

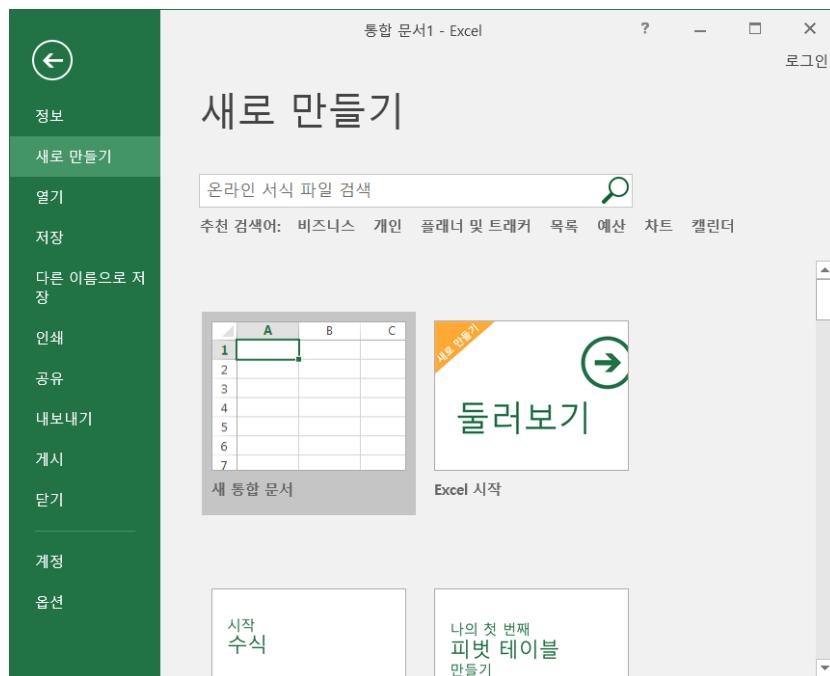


그림 2.3.1: 새로 만들기

## 제 2.4 절 파일 저장

엑셀에서 작업을 마치고 저장하는 방법은

- 파일 → 저장 버튼을 누르거나
- 이미 저장되어 있는 파일을 열어서 작업하고 파일→ 다른 이름으로 저장

하는 두 가지 방법이 있다. 다른 이름으로 저장할 때에는 저장하는 파일 형식은 2.1 절에 있다. 또한 엑셀 파일은 암호를 사용하여 저장할 수 있으며, 이후에 이 파일을 로드할 때에는 암호를 입력해야 그 내용을 읽을 수 있거나 수정 또는 저장 등을 할 수 있다. 파일에 암호 입력은 다른 이름으로 저장 버튼을 누르면 다른 이름으로 저장 창이 나오고(그림 2.4.1), 이 창에서 도구→일반 옵션 메뉴를 누르면 일반 옵션 창이 나타나며(그림 2.4.2) 여기에 암호를 설정할 수 있다.

- 열기암호 : 파일을 열 때 입력하는 암호
- 쓰기암호 : 파일을 저장할 때 입력하는 암호
- 읽기 전용 권장 : 파일 내용은 볼 수 있으나 파일을 수정할 수 없도록 설정할 때 사용

## 제 2.5 절 시트(sheet)

시트(sheet)는 셀이 모여서 구성이 된 엑셀의 저장 단위이며, 이 시트가 모여서 통합문서(workbook)가 구성된다.

시트는 기본 설정이 워크북에 세 개 생성되도록 되어 있지만, 필요에 따라 새 문서를 열 때 그 값을 변경할 수 있으며 설정 과정은 파일 → 옵션 → 일반 → 포함할 시트 수이며, 그림 2.1(a)에서 확인할 수 있다.

시트 이름 변경은 시트 탭을 더블 클릭하면 반전 상태가 되며 이때 사용자가 원하는 시트명을 입력한다.

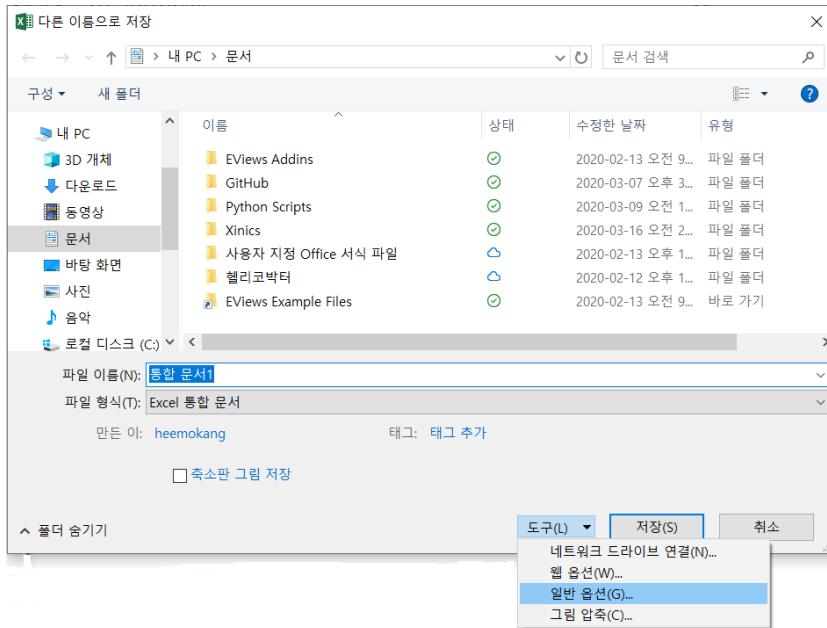


그림 2.4.1: 다른 이름으로 저장

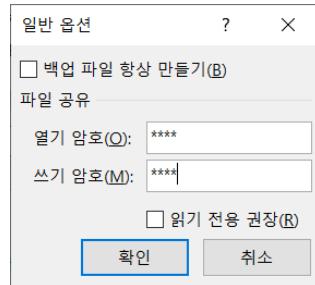


그림 2.4.2: 일반 옵션

### 시트 위치 이동은

- ① 이동할 시트 탭에서 시트 이름을 누른 후 옮길 위치에 드래그하거나
- ② 이동할 시트 탭에서 마우스 오른쪽을 클릭하여 이동/복사를 선택하여 할 수 있다.

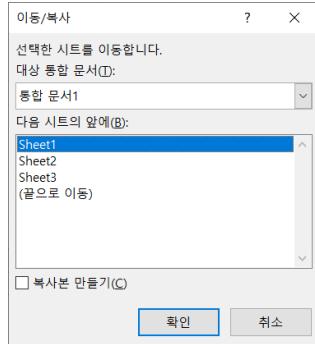


그림 2.5.1: 시트 이동 및 복사

시트에 배경화면 삽입은 페이지 레이아웃 → 배경 메뉴를 선택하고 사용자가 원하는 그림을 선택한다(그림 2.5.2). 시트의 배경화면 삭제는 페이지

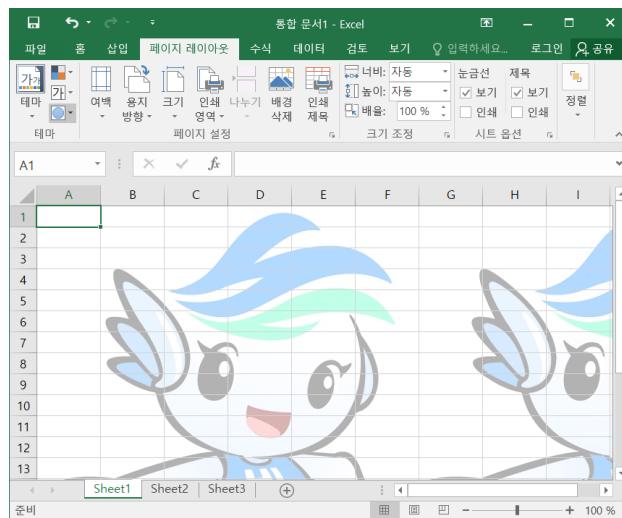


그림 2.5.2: 시트 배경 화면

레이아웃 → 배경삭제 메뉴를 선택한다.

시트의 모든 내용 복사는 복사할 시트 탭에서 마우스 오른쪽 버튼을 누르면 이동/복사 메뉴를 클릭한다.

- ◉ 대상 통합문서 : 복사/이동할 시트가 포함될 통합문서를 설정한다. 기본값은 현재 통합문서이다.
- ◉ 다음 시트의 앞에 : 복사/이동할 시트의 위치를 설정한다.
- ◉ 복사본 만들기 : 이 메뉴를 선택하면 원본과 시트 복사본이 만들어지고, 선택하지 않으면 시트가 이동한다.

## 제 2.6 절     인쇄

인쇄는 엑셀파일의 내용을 출력장치인 프린터로 출력하는 기능을 제공한다.  
인쇄 메뉴는

- ◉ 인쇄 : 이 메뉴를 실행하면 로컬 컴퓨터에서 인쇄할 수 있는 프린터의 목록이 인쇄 창에 나타나고 출력할 프린터를 지정하면 해당 프린터로 내용이 출력된다.
- ◉ 빠른인쇄 : 프린터로 엑셀 파일을 인쇄하는데 기존에 설정된 프린터로 출력하며 인쇄화면 창이 보이지 않는다.
- ◉ 인쇄미리보기 : 프린터로 출력하기 전 인쇄할 내용을 확인하는 메뉴로 인쇄 창에서도 인쇄미리보기를 이용할 수 있다. 이 메뉴를 실행하면 출력될 면이 화면에 점선으로 보인다.

## 제 2.7 절     페이지 레이아웃

페이지 레이아웃 리본 메뉴는 인쇄에서 출력할 때 사용할 메뉴들이 대부분 포함되어 있다.

- ◉ 여백 메뉴 : 출력될 면의 상, 하, 좌, 우에 해당되는 여백을 설정한다.
- ◉ 용지 방향 메뉴 : 용지 방향을 가로, 세로 중 선택할 수 있으며, 기본값은 가로이다.

- 크기 메뉴 : 용지 크기를 설정할 수 있으며, 기본값은 A4이다.
- 인쇄 영역 메뉴 : 시트의 특정한 부분을 출력하기 원할 때, 마우스나 키보드로 특정 영역을 블락으로 설정한 후, 인쇄 영역 설정을 하면 이 부분만 인쇄된다. 인쇄 영영 취소 버튼을 누르면 설정이 해제된다.
- 나누기 메뉴 : 인쇄할 영역이 현재 셀의 우측 상단을 기준으로 설정된다.
- 배경 메뉴 : 시트의 백그라운드를 설정하는 메뉴이다.
- 인쇄 제목 메뉴 : 이 메뉴는 페이지, 여백, 머리글/바닥글, 시트를 설정한다.
  - ✖ 페이지 : 용지방향, 용지의 확대 축소비율, 용지 크기, 인쇄 품질 등을 설정하다.
  - ✖ 여백 : 인쇄할 페이지의 머리글, 바닥글, 좌, 우, 상, 하의 여백을 설정한다.
  - ✖ 머리글/바닥글 : 인쇄할 때 출력될 머리글과 바닥글의 내용을 설정한다.
  - ✖ 시트 : 인쇄영역과 인쇄할 때 항상 내용이 보이도록 반복할 행과 반복할 열을 설정할 수 있다. 또한 인쇄할 때 별도로 표를 셀에 테두리를 그리지 않더라도 눈금선을 출력할 수 있다. 페이지 순서는 열 우선과 행 우선 중 선택할 수 있다.

## 제 2.8 절 Excel 옵션

Excel 옵션은 엑셀을 편리하게 사용할 수 있는 메뉴들을 모두 모아놓은 것으로 파일→옵션을 클릭하여 그 내용을 확인할 수 있다(그림 2.8.1, 그림 2.8.2).

### 2.8.1 일반

이 메뉴에는 엑셀에서 가장 많이 사용하는 옵션을 변경한다.

- ◉ 사용자 인터페이스 옵션

- ✖ 선택 영역에 미니 도구 모음 표시
- ✖ 실시간 미리 보기 사용
- ✖ 색 구성표
- ✖ 화면 설명 스타일

- ◉ 새 통합 문서 만들기

- ✖ 다음 글꼴 사용
- ✖ 글꼴 크기
- ✖ 새 시트의 기본 보기
- ✖ 포함할 시트 수

- ◉ Micorsoft Office 개인 설정

- ✖ 사용자 이름
- ✖ Office 테마

- ◉ 시작 옵션

- ✖ 기본적으로 Excel에서 열 확장명 선택
- ✖ Microsoft Excel이 스프레드시트를 보고 편집하는 기본 프로그램이 아닌 경우 알림
- ✖ 이 응용 프로그램을 시작할 때 시작 화면 표시

### 2.8.2 수식

수식 계산, 성능 및 오류 처리에 관련된 옵션을 변경한다.

- ◉ 계산 옵션

- ✖ 통합 문서 계산

- ✖ 반복 계산 사용

- ◉ 수식 작업

- ✖ R1C1 참조 스타일

- ✖ 수식 자동 완성 사용

- ✖ 수식에 표 이름 사용

- ✖ 피벗 테이블 참조에 GetPivotData 함수 사용

- ◉ 오류 검사

- ✖ 다른 작업을 수행하면서 오류 검사

- ◉ 오류 검사 규칙

- ✖ 오류를 반환하는 수식이 있는 셀

- ✖ 한 표에서 다른 계산된 열 수식이 사용된 셀 표시

- ✖ 2자리로 표시된 연도가 있는 셀

- ✖ 앞에 아포스트로피가 있거나 텍스트로 서시이 지정된 숫자

- ✖ 한 영역에서 다른 수식이 사용된 셀 표시

- ✖ 수식에 사용된 영역에 누락된 셀 있음 표시

- ✖ 수식을 포함한 셀 잠그지 않음

- ✖ 빈 셀을 참조하는 수식 사용

- ✖ 표 데이터 유효성 오류

### 2.8.3 언어 교정

엑셀에서 텍스트를 수정하거나 서식을 지정하는 방법을 변경한다.

- ◉ 자동 고침 옵션

- ✖ 자동 고침 옵션 : 텍스트를 수정하거나 서식을 지정하는 방법을 선택한다

- ◉ Microsoft Office 프로그램에서 맞춤법 검사

- ✖ 대문자로 된 단어 건너뛰기
  - ✖ 숫자가 있는 단어 건너뛰기
  - ✖ 인터넷 주소와 파일 경로 건너뛰기
  - ✖ 반복되는 단어에 플래그로 설정
  - ✖ 한국어:보조용언 붙여쓰기
  - ✖ 한국어:오용어 목록 검색
  - ✖ 한국어:복합 명사 처리
  - ✖ 기본사전에 있는 단어로만 추천 : 사용자 지정 사전을 클릭하여 새 단어도 추가할 수 있다.
  - ✖ 사용할 사전의 언어 : 기본값은 설정된 언어이다.

### 2.8.4 저장

통합 문서의 저장 방법을 사용자가 직접 설정한다.

- ◉ 통합 문서 저장

- ✖ 다음 형식으로 파일 저장
  - ✖ 자동 복구 정보 저장 간격
  - ✖ 자동 복구 파일 위치

✖ 기본 파일 위치

- ◉ 자동 복구 예외 항목
  - ✖ 이 통합 문서에만 자동 복구 사용 안 함
- ◉ 문서 관리 서버 파일에 대한 오프라인 편집 옵션
  - ✖ 체크 아웃 파일 저장 위치
    - 이 컴퓨터의 서버 임시 보관함
    - Office 문서 캐시
  - ✖ 서버 임시 보관함 위치
- ◉ 통합 문서의 시작적 모양 유지
  - ✖ 이전 버전의 Excel에 표시될 색 선택

### 2.8.5 언어

Office 언어 기본설정을 구성

- ◉ 편집 언어 선택
- ◉ 표시 및 도움말 언어 선택 : 언어 선택의 우선 순위
- ◉ 화면 설명 언어 선택

### 2.8.6 고급

엑셀에서 제공하는 고급옵션을 선택한다.

- ◉ 편집 옵션
  - ✖ <Enter> 키를 누른 후 다음 셀로 이동 : 이동할 위치는 아래쪽 위쪽, 왼쪽, 오른쪽에서 선택할 수 있다.

- ✖ 소수점 자동 삽입 : 소수점을 생성하는 기본값을 설정한다.
  - ✖ 채우기 핸들 및 셀 끌어서 놓기 사용 : 덮어쓸 때 경고 표시를 할 것인가를 결정한다.
  - ✖ 셀에서 직접 편집 허용 : 셀에서 직접 편집을 허용할 것인가를 결정한다.
  - ✖ 데이터 범위의 수식과 수식을 확장
  - ✖ 자동 % 입력 사용 : %를 입력할 때 수식에서 %로 전환할 것인가 결정한다.
  - ✖ 셀 내용을 자동 완성 : 이미 입력된 값은 자동 완성하여 사용할 것을 결정한다.
  - ✖ IntelliMouse로 화면 확대/축소 : 화면 확대/축소를 결정한다.
  - ✖ 시간이 오래 걸리는 작업이 발생할 경우 사용자에게 알림
  - ✖ 시스템 구분 기호 사용
  - ✖ 커서 움직임
- ◉ 잘라내기/복사/붙여넣기
    - ✖ 붙여넣기 옵션 단추 표시 : 붙여넣기 옵션 단추가 표시되도록 설정한다.
    - ✖ 삽입 옵션 단추 표시 : 삽입 옵션 단추가 표시되도록 설정한다.
    - ✖ 삽입한 개체를 상위 셀과 함께 잘라내기, 복사 및 정렬
  - ◉ 이미지 크기 및 품질
    - ✖ 편집 데이터 취소
    - ✖ 파일 이미지 압축 안 함
    - ✖ 기본 대상 출력 설정
  - ◉ 인쇄

✖ 그래픽의 고품질 모드 : 인쇄를 고품질 모드로 설정한다.

● 차트

✖ 마우스가 위치하면 차트 요소 이름 표시

✖ 마우스가 위치하면 데이터 요소 값 표시

● 표시

✖ 표시할 최근 문서 수 : 최근에 사용한 최근 문서 수를 보여준다.  
기본값은 25개이다.

✖ 눈금자 단위

✖ 작업 표시줄에 모든 창 표시

✖ 수식 입력줄 표시

✖ 함수 화면 설명 표시

✖ 하드웨어 그래픽 가속 사용 안 함

✖ 메모가 있는 셀 표시

- 메모와 표식 모두 표시 안 함
- 표식만, 마우스만 위치하면 메모 표시
- 메모와 표식

✖ 기본방향

- 오른쪽에서 왼쪽
- 왼쪽에서 오른쪽

● 이 동합 문서의 표시 옵션

✖ 가로 스코롤 막대 표시

✖ 세로 스크롤 막대 표시

✖ 시트 탭 표시

✖ 자동 필터 메뉴에서 날짜 그룹화

✖ 개체표시

- 모두

- 표시 암 함(개체 숨기기)

◉ 이 워크시트의 표시 옵션

✖ 행 및 열 머리글 표시

✖ 계산 결과 대신 수식을 셀에 표시

✖ 시트 방향 바꾸기

✖ 페이지 나누기 표시

✖ 0 값이 있는 셀에 0 표시

✖ 윤곽을 설정한 경우 윤곽 기호 표시

✖ 눈금선 표시

✖ 눈금선 색

◉ 수식

✖ 다중 스레드 계산 허용

✖ 계산 드레드 수

- 이 컴퓨터의 모든 프로세서 사용
- 수동

◉ 이 통합 문서의 계산 대상

✖ 다른 문서에 대한 링크 업데이트

✖ 표시된 정밀도 설정

✖ 1904 날짜 체계 사용

✖ 외부 연결값 저장

◉ 일반

- ✖ 작업 완료 후 소리로 알림
- ✖ 작업 완료 후 애니메이션으로 알림
- ✖ DDE(동적 데이터 교환)를 사용하는 다른 응용 프로그램 무시
- ✖ 자동 연결 업데이트 확인
- ✖ 추가 기능의 사용자 인터페이스 오류 표시
- ✖ A4 또는 8.5 × 11" 용지 크기에 맞게 내용 조정
- ✖ 고객 제출 Office.com 콘텐츠 표시
- ✖ 시작 시 다음 위치의 모든 파일 열기
- ✖ 웹 옵션
- ✖ 다중 스레드 처리
- ✖ 큰 피벗 테이블의 경우 새로 고침 작업을 실행 취소할 수 없도록 설정하여 새로 고침 시간을 출입니다.
- ✖ 데이터 원본 행 수가 다음 이상인 피벗 테이블의 경우 취소할 수 없도록 설정(천 단위):
- ✖ 정렬 및 채우기 순서에서 사용할 목록 만들기

◉ Lotus 호환성

- ✖ Microsoft Office Excel 메뉴 키
- ✖ 키보드 명령 바꾸기
- ✖ Lotus 1-2-3 수식평가
- ✖ Lotus 1-2-3 수식입력

### 2.8.7 리본 사용자 지정

리본 메뉴를 사용자 지정합니다. 즉 리본 메뉴에 자주 사용하는 것들을 추가하거나 레본 메뉴에 없는 명령을 추가하는 기능.

### 2.8.8 빠른 실행 도구 모음

빠른 실행 도구 모음을 사용자가 지정합니다. 즉 많이 사용하는 메뉴들은 제 목표시줄 오른쪽에 놓고 간편하게 사용할 수 있도록 설정하는 기능.

### 2.8.9 추가기능

Microsoft Office 추가 기능을 보고 관리한다.

- ◉ 추가기능
  - ✖ 활성 응용 프로그램 추가 기능 : 이미 추가 기능이 적용된 메뉴들
  - ✖ 비활성 응용 프로그램 초기 기능 : 추가할 수 있는 메뉴들
- ◉ 관리 : 여러 가지 추가 기능을 관리하는 것으로 기본값은 Excel 추가 기능으로 설정되어 있다.

### 2.8.10 보안 센터

문서 및 컴퓨터를 안전하고 보안이 유지된 상태로 관리한다.

- ◉ 개인 정보 보호 : Microsoft의 개인 정보 보호 정책 관련
- ◉ 보안 및 기타 : Microsoft Office Online의 개인 정보 보호 및 보안에 대하여 자세히 알아본다.
- ◉ Microsoft Office Excel 보안 센터 : 보안 센터에는 개인정보가 설정이 있으므로 컴퓨터를 안전하게 유지할 수 있으며 설정값을 변경하지 않는 것이 좋다.



그림 2.8.1: Excel 옵션

HEEMO KANG

엑셀 VBA

## 2.8. EXCEL 옵션

27

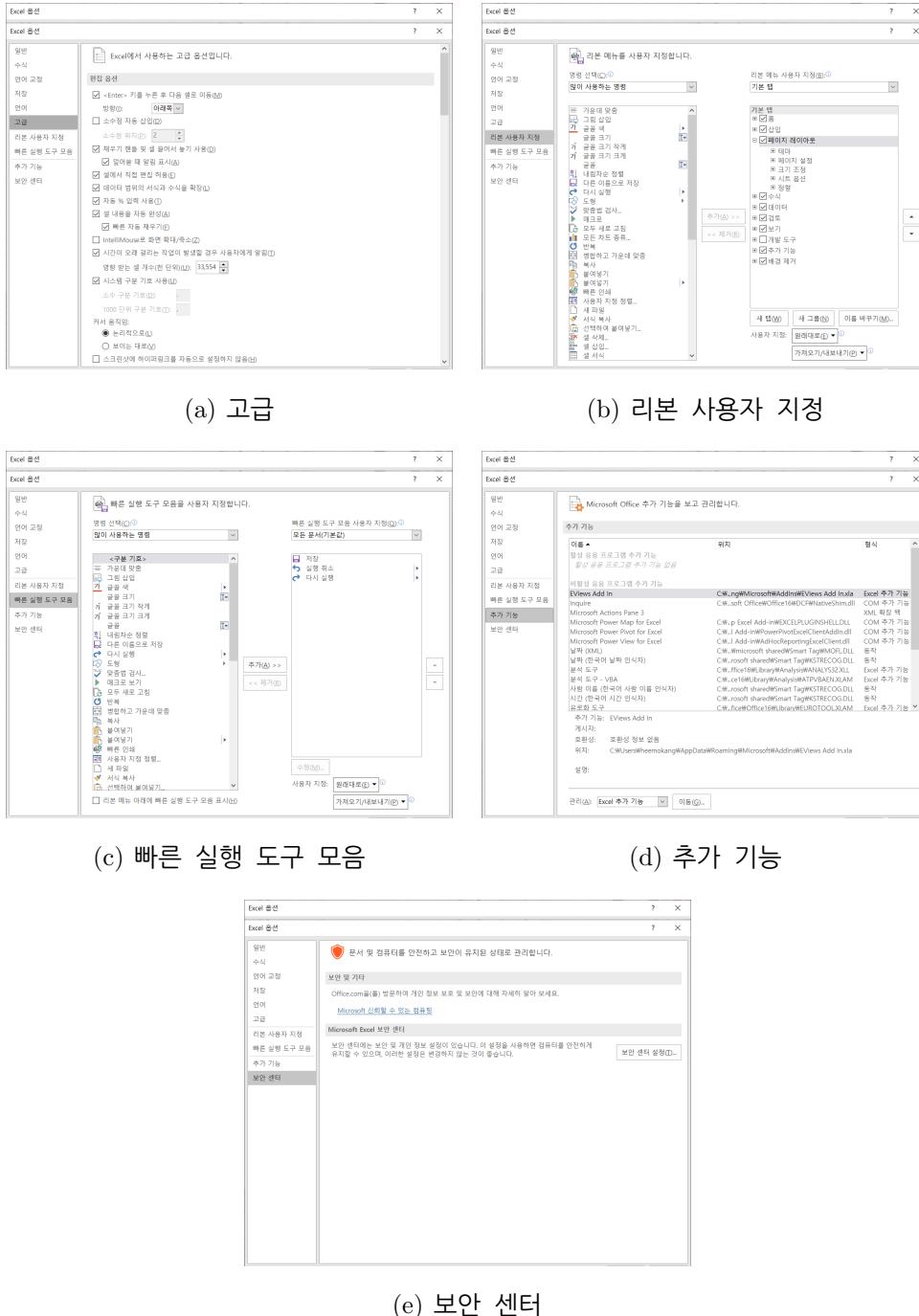


그림 2.8.2: Excel 옵션

엑셀 VBA

HEEMO KANG

## 제 2.9 절 이미지 편집

그림파일은 파일 전체를 사용하기도 하지만 경우에 따라 일부를 사용할 경우도 발생한다. 따라서 전문적인 그림 편집 능력보다 간단한 그림 편집 기술에 대하여 알아보자.

### 2.9.1 이미지 일부분 저장하기

Windows 10에서 제공하는 그림판(mspaint)으로 이미지 일부분 저장에 대하여 알아보자.

### 2.9.2 이미지 배경색 지우기

웹페이지 <https://pixlr.com>에서 제공하는 프로그램으로 이미지 배경색을 제거할 수 있다.

### 2.9.3 이미지 흐리게 하기

이미지 파일에 배경색이 진하면 의미가 없다. 따라서 그림 2.5.2와 같이 그림 파일을 흐리게 해야 은은한 효과가 있다. 웹에서 photoscape 프로그램을 다운로드하여 설치한 후 그림 파일을 흐리게 만들어 보자.

## 연습문제

1. 새로운 엑셀파일을 열고 A1 셀에 “파일 저장”을 입력한 후 파일명 ‘cells.xlsx’로 저장한다. 저장할 때 읽기 암호는 ‘1234’, 쓰기 암호는 ‘5678’로 설정 한다.
2. 현재 세금계산서를 발행하려고 한다. 엑셀에 기존에 만들어진 서식파일이 있는지 찾아 보고 그 과정을 모두 적어라.

3. 인터넷에서 태극기가 있는 그림을 검색한다. 그 중 자신이 마음에 드는 것을 선택하여 다운로드 한 후 시트의 뒤배경(background)로 설정한다.
4. 새 통합 문서를 열면, 시트가 50개 열리도록 설정하여라.
5. 새 통합 문서를 열고 50번째 마지막 시트로 이동한다. 그 과정을 적어라.
6. 현재 리본메뉴에서 개발도구 리본메뉴가 보이지 않으면 나타나도록 설정하여라.
7. 계산의 정확도를  $10^{-8}$ 이 되도록 설정하여라.
8. 현재 시트에서 행 개수와 열 개수가 몇 개인지 적어라.
9. 엔터 키를 누르면 현재 셀이 오른쪽으로 이동하게 설정하고 과정을 적어라.
10. 실행한 엑셀 파일을 화면 캡쳐하여 그림판에 옮기고 각 명칭을 적어라. 완성한 그림은 현재 작업중인 한글 워드프로세서에 붙여넣기 한다.
11. 새 문서를 편집한 후 저장을 누르면 C:\temp 폴더에 파일이 저장될 수 있도록 설정하여라.
12. 마우스로 채우기 핸들을 실행하면 “아이린”, “슬기”, “웬디”, “조이”, “예리”가 계속 순환하여 나타나게 설정하여라.



# 제 3 장

## 셀(cell)에 대하여

엑셀에서 자료 입·출력 최소단위는 셀이다. 즉 셀에서 모든 데이터를 입력, 삭제, 수정할 수 있기 때문에 셀 기능을 잘 활용해야 엑셀을 능숙하게 사용할 수 있다. 이 장에서는 셀에 자료 입력 방법과 셀 값의 출력 변경 과정을 알아본다.

### 제 3.1 절 셀 번지

현재 셀은 현재 검정색 테두리 셀로 하나 이상의 셀이 될 수 있다. 현재 셀 변경은 셀을 클릭하거나 키보드에 있는 화살표 키를 이용하여 원하는 셀로 이동한다. 현재 셀은 그 위치가 행열의 형태로 셀 번지를 가지는데 그 위치가 이름상자에 보인다. 셀 번지는 열(column) 이름과 행(row) 이름이 결합하여 표현한다. 셀 번지 기본 설정은 열은 영문 대문자, 행은 아라비아 숫자로 표시 하며 A1 참조 스타일이라고 부른다.

- ◉ 행 이름은 1~2절에서 언급하였듯이 숫자로 표현하며 1에서  $2^{20} = 1,048,576$  이다.
- ◉ 열 이름은 영문 알파벳으로 표현하며 A B ... Z AA AB ... ZY ZZ AAA AAB ... XFC XFD까지  $2^{14} = 16,384$ 개이다.

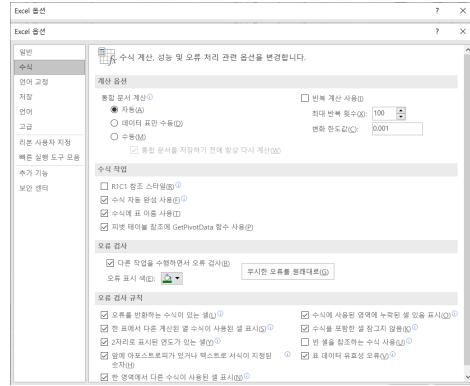


그림 3.1.1: R1C1 참조 스타일

예를 들어 A10 셀 번지는 A열 10행이다. 또 다른 셀 번지 표현방법으로 열과 행을 모두 아라비아 숫자로 표시하는 R1C1 참조 스타일이 있으며 Excel 옵션 → 수식에서 R1C1 참조 스타일을 선택한다(그림 3.1.1).

## 제 3.2 절 셀 입력, 셀 편집

셀에 값이 입력되면 그 값은 크게 숫자, 문자로 분류된다. 입력된 값이 숫자인지 문자인지 분류하는 방법은

- ◉ 화면을 보고 분류하는 방법 : 입력값이 오른쪽 맞춤이면 숫자로 왼쪽 맞춤이면 문자로 인식한다.

[조건] 해당 셀 가로 맞춤이 일반으로 설정되어 있어야 된며, 셀 맞춤은 셀 서식 → 맞춤 → 텍스트 맞춤 → 가로가 일반으로 설정한다.

- ◉ 함수를 이용하여 분류하는 방법 : ISNUMBER<sup>1</sup>, ISNONTEXT 함수를 이용하여 숫자와 문자를 구분할 수 있다.

<sup>1</sup>함수의 사용법은 4 장에 소개되어 있다.

현재 셀에 값을 입력하면 이전에 있던 값은 지워지고 새로운 내용이 셀에 입력된다. 따라서 셀 편집은

- ◉ 마우스로 해당 셀을 더블 클릭하거나
- ◉ 해당 셀에서 F2 키를 누르고

편집 상태에서 수정해야 한다.

다음 값 a와 01과 '01을 셀에 입력해 보자.

- ◉ a는 입력결과가 그대로 셀에 보여지고 왼쪽 정열되며 문자로 인식한다.
- ◉ 01은 1이 셀에 보여지고 오른쪽 정열되며 숫자로 인식한다.
- ◉ '01은 01이 셀에 보여지고 왼쪽에 정열되며 문자로 인식한다. 이 경우는 숫자를 문자로 저장할 때 사용하며 휴대폰 번호와 같이 맨 앞에 0을 반드시 입력할 경우 유용하게 사용할 수 있다.

셀에 날짜 출력은 다음과 같다.

- ◉ [yyyy]-mm-dd : 셀에 년도- 월-일을 입력한다. 년도를 생략하면 입력 년도의 월-일이 출력된다(예 : 2019-2-18).
- ◉ [yyyy]/mm/dd : 셀에 년도- 월-일을 입력한다. 년도를 생략하면 입력 년도의 월-일이 출력된다(예 : 2019/2/18).

셀에서 단축 키를 사용하여 특수한 결과를 얻을 수 있다.

- ◉ CTRL + :(콜론) : 현재 날짜 출력
- ◉ CTRL + SHIFT + :(콜론) : 현재 시간 출력

특수 기호의 삽입은

- ◉ 삽입→ 기호 메뉴에서 사용자가 원하는 것을 삽입한다.
- ◉ 한글 자음 + 한자 키를 누르고 사용자가 원하는 특수 기호를 삽입한다.

### 제 3.3 절 셀 이동

셀의 위치를 이동하려면

- ENTER 키를 누르면 아래로
- TAB 키를 누르면 오른쪽으로

이동한다. 만일 enter 키를 눌렀을 때 이동하는 방향을 변경하려면 파일 → 옵션 → 고급에서 <Enter> 키를 누른 후 다음 셀로 이동(M)에서 원하는 방향을 설정한다. 또한 화살표 키를 이용하여 좌, 우, 상, 하로도 이동할 수 있다.

현재 셀에서 특정한 셀로 이동은

- 이미 등록된 이름을 이름상자에서 선택한다.
- 이름상자에 등록된 이름을 입력한다.
- 현재 셀에서 CTRL + →, ←, ↑, ↓를 누르면 화살표 방향으로 셀 값이 있는 셀로 찾아가며 빈 셀은 무시하고 지나친다. 만일 화살표 방향에 모두 빈 셀이라면 시트의 맨 끝에 있는 셀로 이동한다.

### 제 3.4 절 셀 선택

엑셀에서 작업은 선택한 현재 셀에서 할 수 있도록 기본값이 설정되어 있다. 그러나 경우에 따라서 여러 개의 셀을 선택하여 작업할 수도 있다. 다수의 셀 선택은 다음과 같다.

- 마우스를 이용하는 방법 : 선택한 셀을 누르고(press) 드래그(drag)한 후 마우스를 떼면(release) 그 안에 있는 셀이 모두 선택된다.
- 키보드를 이용하는 방법 : 선택을 원하는 셀에 현재 셀을 위치시키고 SHIFT + →, ←, ↑, ↓를 누르면 그 안에 있는 모든 셀이 선택된다.
- 키보드와 마우스를 이용하는 방법 : CTRL 키 + 선택을 원하는 셀을 클릭 한다.

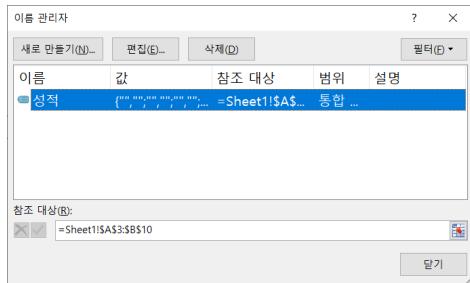


그림 3.5.1: 이름관리자

## 제 3.5 절 이름상자

이름상자는

- ◉ 현재 셀을 특정한 셀로 이동하거나
- ◉ 여러 셀에 이름을 지정할 수 있다.

이름상자 위치는 5 페이지 그림 1.2.1에서 확인한다. 예를 들어 B2000 셀 번지로 이동하려면 이름상자에 B2000를 입력하여 이동할 수 있다. 또한 A3:B10 셀을 마우스로 선택하고 이름상자에 성적을 입력하면 선택한 셀 이름은 성적으로 등록된다. 이후 이름상자에 등록된 여러 셀 성적의 평균은 =AVERAGE(성적)으로 구한다. 사용자가 등록한 이름상자는 수식 → 정의된 이름 → 이름관리자에서 새로 만들기, 편집 및 삭제가 가능하다.

## 제 3.6 절 셀 서식

셀 서식은 셀에서 제공하는 여러 기능을 사용하여 셀들을 보기 좋게 꾸민다. 셀 서식은 6 가지가 있으며 각 기능에 대하여 알아보자. 셀 서식 실행은

- ◉ 셀을 선택한 후 마우스 오른쪽 버튼 → 셀 서식이나
- ◉ 홈 리본 메뉴 → 표시형식 메뉴에서 대화 상자 표시 아이콘( )을 클릭한다.

셀 서식 창은 여섯 가지 기능이 있으며 각 탭 선택하면 해당 메뉴로 이동한다. 각 메뉴 기능은

- 표시형식 : 입력된 셀 값은 변경되지 않고 화면에 보이는 내용을 설정 한다. 그림 3.1(a)는 출력 형식을 범주에서 일반에서 사용자 지정까지 선택할 수 있다. 범주 → 사용자 지정은 사용자가 직접 설정한 값이 화면 출력 내용에 적용된다.
- 맞춤 : 글자의 가로 세로에 대한 정열, 글자의 방향, 들여쓰기, 텍스트 조정(텍스트 바꿈, 셀에 맞춤, 셀병합)을 조절한다.
- 글꼴 : 셀에 입력된 값에 대하여 글자체, 글자크기, 글자 색 등을 변경할 수 있다. 워드프로세서 한글에서 글자모양 메뉴와 동일한 기능이다.
- 테두리 : 워드프로세서 한글에서 표그리기와 동일한 기능이다.
- 무늬 : 셀의 색이나 무늬를 지정한다.
- 보호 : 잠금된 셀을 보호하는 기능으로 암호를 지정한다. 이 기능은 여러 가지 제약사항이 있으므로 다른 작업을 모두 하고 이 기능을 수행하는 것이 좋다.

이다.

### 3.6.1 표시형식

표시형식은 셀 값의 출력 설정으로 종류는 숫자, 날짜 또는 시간, 통화, 백분율, 지수 표기와 텍스트나 공백이 있다. 특별히 사용자가 원하는 표시형식 설정은 사용자 지정을 사용한다. 표시형식 사용 방법은

1. 셀 서식을 적용할 셀을 선택하고
2. 마우스 오른쪽 버튼을 클릭하여 셀 서식 메뉴를 클릭한다.
3. 셀 서식 창에서 표시 형식 탭을 클릭한다.



그림 3.6.1: 셀 서식

4. 표시 형식 탭의 범주 상자에서 사용할 표시 형식의 종류를 선택한다.

표시형식 중 자주 사용하는 것들은

- 표시형식의 범주를 리스트의 풀다운 메뉴에서 선택
- 소숫점 늘이기와 줄이기
- 원 통화량 표시
- 숫자 세 자리마다 컴마 표시

등이 리본메뉴의 홈 탭에 있다. 범주 상자에서 사용자기 직접 서식을 만드려면 사용자 지정은 위에서 소개한 표시형식 사용방법에 다음과 같은 과정을 추가하면 된다. 그 과정은 셀 서식 → 표시형식에서

1. 범주 상자에서 사용자 지정을 클릭한다.
2. 형식 상자에서 서식 코드를 편집하여 원하는 서식을 만든다.

여기서 기본적으로 설정된 표시형식은 삭제할 수 없으나 사용자 지정 서식은 삭제가 가능하다. 만일 사용자 지정 서식이 많이 설정되면 더 이상 서식이 추가되지 않는다. 이 경우가 엑셀에서 허용하는 사용자 지정 서식이 모두 추가한 경우이므로 이미 설정한 사용자 지정 서식에서 일부 목록을 삭제해야 다른 서식을 추가 할 수 있다.

서식 코드는 숫자나 문자 항목의 서식 설정으로, 최대 네 항목까지 지정할 수 있고, 각 항목 구분자는 세미콜론(;)을 사용한다. 서식 코드의 순서는

- 양수에 대한 서식 설정
- 음수에 대한 서식 설정
- 영(0)에 대한 서식 설정
- 텍스트가 입력될 경우 적용할 서식 설정

으로 정의한다. 서식 코드 사용방법은

$\overbrace{\#,\#\#\#.00\_}^{\text{양수 형식}}$ ; $\overbrace{[\text{빨강}] (\#, \#\#\#.00)}^{\text{음수 형식}}$	$\overbrace{0.00}^{\text{영(0) 형식}}$ ; $\overbrace{"\text{판매량"}@}^{\text{텍스트 형식}}$
--	---

이다. 이 네 가지 서식 코드 항목을 항상 모두 설정하지 않고 필요한 것만 사용할 수 있다. 서식 코드 항목을 하나만 사용하면 모든 숫자에 적용되는 서식이고, 숫자 서식 코드를 두 개 사용하면 양수와 음수에 적용되는 서식이다. 만일 네 개의 서식 코드 중 사용하지 않으면 세미콜론(;)만 쓰고 다음 서식 코드를 입력한다.

첫 번째 숫자 서식 코드는

- ◉ 정수 부분의 유효 자릿수
- ◉ 정수 부분의 세 자리마다 콤마(,)로 구분하기
- ◉ 소수 부분의 유효 자릿수
- ◉ 소수 부분의 분수 사용
- ◉ 지수 사용
- ◉ 날짜, 시간, 통화, 백분율 등
- ◉ 색깔
- ◉ 비교문 사용

숫자의 유효 자릿수 지정에서 서식에 있는 형식보다 실제값의 소수점 아래 자릿수가 더 많으면 자릿수가 반올림되고 서식에 있는 형식보다 실제값의 소수점 위 자릿수가 더 많은 경우에는 그대로 나타난다. 숫자 서식 코드는 예약된 세 개 문자로 다음과 같이 서식 형식을 지정한다.

- ◉ # : 유효 자릿수만 나타내고 유효하지 않은 0은 보이지 않는다.
- ◉ ? : 소수점 위나 아래에 있는 유효하지 않은 0 대신 공백을 추가한다. Courier New와 같은 너비가 고정된 글꼴 서식이 지정될 때 소수점을 맞준다. 자릿수가 다른 분수에도 ?를 사용할 수 있다.
- ◉ 0 : 서식에서 설정한 0의 수보다 입력값의 자릿수가 적으면 입력한 숫자를 출력하면서 유효하지 않은 자리수에는 0을 채운다.

이 숫자 서식이 사용된 예를 표 3.6.1에 정리하였다. 천 단위 구분 기호는 쉼

입력	출력	서식 코드
3.1415	3.1	0.#
1234.56	1234.6	#####.#
19	19.0	#.0#
1234.567	1234.57	
45.246, 100.21, 5.4	소수점을 맞춤	????.???
1.5	1 1/2	
4.6	4 6/10	# ???/????
8.9	8.900	#.000

표 3.6.1: 숫자 코드 사용

표만 삽입하거나 천의 배수 형태에 쉼표를 사용한다. 쉼표만 사용한 경우는 자리수의 단위수가 줄어들고 천의 배수형태에 쉼표를 같이 사용한 경우는 천의 자리마다 구분하여 보여준다(표 3.6.2).

서식 영역에 색 지정은 다음 여덟 개 색상 중 하나를 대괄호로 묶어 입력 한다. 색 코드는 서식에서 맨 앞이고 색을 대괄호([ ]) 안에 색을 입력한다. 색 종류는 다음과 같이 한글로

- ◉ [검정]
- ◉ [파랑]

입력	출력	서식 코드
14000	14	#,
14000	14,000	#,###
14400000	14.4	0.0,,

표 3.6.2: 천 단위 숫자 사용

◉ [자홍]

◉ [빨강]

◉ [청록]

◉ [녹색]

◉ [흰색]

◉ [노랑]

으로 설정한다. 다음 서식은 60보다 크거나 같으면 파란색 글꼴로, 60보다 작은 숫자는 빨간색 글꼴로 나타낸다. 이 예가 적용된 것은 37 면의 그림 3.1(a)에서 확인할 수 있다.

### 보기 3.6.1. 【색깔 서식】

[파랑] [>=60]#; [빨강] [<60]#

[주의] : 때로는 도움말에 색을 영어로 설정해도 된다는 경우가 있으나 한글 버전의 엑셀에서 영어로 색을 입력하면 에러가 발생하므로 반드시 색은 한글로 입력해야 올바로 실행된다.

두 번째로 날짜와 시간에 대한 서식 코드 형식에 대하여 알아보자. 날짜와 시간에 대한 서식 코드는

- ◉ 년
- ◉ 월
- ◉ 일
- ◉ 시
- ◉ 분
- ◉ 초
- ◉ AM, PM 등

총 여덟 개의 코드를 사용한다. 날짜 서식 코드는 표 3.6.3에서, 시간 서식코드는 표 3.6.4에서 확인한다. 날짜 서식 코드는 표 3.6.3과 같이 년, 월, 일을 표현하며 시간 코드는 PM, AM을 사용하면 12 시간제, 그 이외인 경우는 24 시간제이다. 시간에서 분 서식 코드 `m`, `mm`은 시 서식 코드 `h`나 `hh` 뒤나 초 서식 코드 `ss` 앞에 `m`을 사용한다. 그렇지 않으면 분 대신 월이 나타난다.

세 번째로 통화, 백분율, 공학용 표기에 대한 숫자 서식 코드에 대하여 알아보자. 숫자 서식에 통화 기호 입력은 기호에 해당하는 ANSI 코드를 입력한다.

[참고] 사용자 지정 서식은 통합 문서에 저장된다. 특정 통화 기호 사용은 Excel을 시작하기 전에 제어판의 국가별 설정에서 선택한 통화 기호를 바꾼다.

백분율 서식 코드는 백분율로 표현하는 숫자 서식에 백분율 기호(%)를 넣는다. 예를 들어

`0.08은.....8%`로,  
`2.8은.....280%`로  
설정한다. 숫자의 지수 표현은 셀에 `E-`, `E+`, `e-`, `e+` 등의 지수 코드를 입력한다. 이들 코드 오른쪽에 0(영)이나 #(숫자 기호)이 들어 있으면 숫자에 E나 e가 삽입되어 지수형 서식으로 나타난다. 여기서 코드 오른쪽의 0이나 #는 지수에서 자릿수이다. E-나 e-는 음의 지수일 때 -(음수 기호)를 사용하고, 양의

표 3.6.3: 날짜 서식 코드

표시	서식 코드
두 자리 연도 : 00–99	Yy
네 자리 연도 : 1900–9999	YYYY
세 문자 요일 : Sun–Sat	Ddd
요일 : Sunday–Saturday	Dddd
두 자리 월 : 01–12	Mm
한, 두자리 월 : 1–12	M
한 문자 월 : J – D	MMMMM
세 문자 월 : Jan–Dec	MMM
월 : January–December	MMMM
두 자리 일 : 01–31	Dd
한, 두자리 일 : 1–31	D

지수일 때 +(양수 기호)를 각각 사용한다. 셀에 다음과 같이

3867000000000000 을 입력하면 ..... 3.867E+14

지수 형태로 보여진다. 이와 같은 출력 형태는 입력된 숫자가 셀에 모두 보여줄 수 없기 때문에 지수 형태로 화면에 보여준다.

네 번째로 텍스트에 대하여 알아보자. 텍스트 서식은

- ◉ 숫자 서식 코드에 텍스트를 추가로 입력하거나
- ◉ 서식코드 네 번째 영역에 추가하여 사용할 수 있다.

먼저 숫자 서식 코드와 함께 문자를 사용하려면,

- ◉ 문자열인 경우에는 큰 따옴표(“ ”)로 텍스트 문자로 묶고
- ◉ 단일 문자인 경우에는 앞에 백슬래시(\)를 입력한다.

예를 들어, 숫자 다음에 음수는 순해 문자열, 양수는 이익 문자열이 보이려면

표시	서식 코드
한 자리 분 : 00~59	Mm
한, 두 자리 분 : 0~59	M
경과 분 : 분:초 (예 63:46)	[mm]:ss
두 자리 시간 : 00~23	Hh
한, 두자리 시간 : 0~23	H
경과 시간 : 시.분 (예 25.02)	[h]:mm
시간 : 시 AM/PM (예 4 AM)	h AM/PM
시간: 시:분 AM/PM (예 4:36 PM)	h:mm AM/PM
시간을 4:36:03 P로	h:mm:ss A/P
두 자리 초 : 00~59	Ss
한, 두 자리 초 : 0~59	S
경과 초 : 초 (예 : 89)	[ss]

표 3.6.4: 시간 코드 사용

1. 표시형식에 \$0.00" 이익";\$-0.00" 손해"로 설정한다.
2. -125.74를 입력하면 ..... \$-125.74 손해
3. 125.74를 입력하면 ..... \$125.74 이익

로 셀에 보인다. 서식에 사용할 문자 중에서 공백 문자( )와 \$ - + / ( ) : ! ^ & ‘(왼쪽 작은 따옴표) ’(오른쪽 작은 따옴표) ~ { } = < > 등의 문자는 따옴표를 사용하지 않고도 나타낼 수 있다.

서식 코드 네 번째 영역에 문자열을 나타내려면

"문자열" @ 또는 @ "문자열"

이다. 여기서 @는 값을 입력할 경우 입력 값의 위치이다. 만일 텍스트 영역에서 @ 문자를 생략하면 입력하는 텍스트가 나타나지 않으며 서식에 텍스트 영역이 없으면 입력하는 텍스트는 서식의 영향을 받지 않는다. 예를 들어 셀에 총 수익 문자열이 항상 나타나도록 하려면 서식코드에 "총 수익 " @으로 입력한다. 그리고 설정된 셀에

56을 입력하면 ..... 총 수익 56  
이 보이게 된다.

숫자 서식에 공백이나 문자를 반복하여 삽입할 수도 있다. 이 기능은 공백을 추가하여 자릿수 조정에 사용하면 양수와 음수가 섞여 있어도 보기 좋다. 숫자 서식에 문자 서식 추가는 다음과 같이 설정한다.

- ◉ \_(밑줄) : 특정 문자 앞에 \_를 넣으면 공백이 특정 문자의 길이만큼 공간이 삽입된다.
- ◉ \*(별표) : 특정 문자 앞에 \*를 넣으면 셀 크기에서 입력한 값을 제외하고 특정 문자가 반복하여 삽입된다.

예를 들어, 표시형식을 (#\_)로 설정하면, 괄호(()) 안에 숫자(#)가 나타나며 음수인 경우에는 음수 기호가 왼쪽 괄호 밖에 위치하고 양수 음수 모두 오른쪽 괄호 앞에 ) 길이 만큼 빈 공간이 추가된다.

셀에 5를 입력하면 ..... 5\_

또한 숫자 뒤에 -를 넣어 셀을 채우려면 0\*-를 입력한다.

셀에 5를 입력하면 ..... 5-----

가 보인다. 숫자이지만 문자처럼 항상 왼쪽에 정렬되게 하려면 표시형식을 0\*\_로 설정한다.

셀에 5를 입력하면 ..... 5\_\_\_\_\_

가 출력되어 실제로 숫자이지만 문자처럼 왼쪽에 정렬된다.

### 3.6.2 맞춤

셀 서식에서 맞춤 탭은 셀 값의 출력을 변경한다. 그 종류는 텍스트 맞춤, 텍스트 방향, 텍스트 조정 등 세 가지가 있다.

- 텍스트 맞춤은 셀 값을 가로와 세로에 대하여 정열한다. 그 종류는 일반, 왼쪽(들여쓰기), 가운데, 오른쪽, 채우기, 양쪽 맞춤, 선택영역의 가운데로, 균등 분할 등이 있다.

✖ 선택영역의 가운데로는 한 셀에 값을 입력하고 오른쪽으로 5 개 셀을 선택한 후 셀 서식에서 이 맞춤 메뉴를 실행하면 세 번째에서 값이 보인다. 하지만 셀 값은 첫 번째에 있다.

✖ 균등 분할은 모든 셀 공간을 균일하게 사용하므로 셀 너비를 넓게 하면 한 글자가 차지하는 공간은 같다.

한 셀에 두 줄 이상 입력은 ALT + ENTER하면 다음 줄에 커서가 위치하게 된다.

- 텍스트 방향은 입력된 글자를 -90 ~ 90까지 조정할 수 있다.

- 텍스트 조정은 텍스트 줄바꿈, 셀에 맞춤, 셀 병합이 있다.

✖ 텍스트 줄바꿈은 셀 안 길이보다 더 많은 글자를 입력하면 자동으로 줄이 바뀐다.

✖ 셀에 맞춤은 셀에 입력되는 문자를 한 셀에 보이게 한다. 입력되는 문자가 많으면 글자가 작게 보인다. 텍스트 줄바꿈과 셀에 맞춤은 서로 반대되는 텁으로 같이 사용할 수 없다.

✖ 셀 병합은 여러 개의 셀을 하나의 셀로 합친다. 셀 병합은 셀에 내용이 있으면 맨 왼쪽 셀의 내용만 남아있고 나머지 셀 내용은 모두 지워진다. 반대로 셀 병합을 해제하면 하나로 되었던 셀이 원래의 셀 개수만큼 분할된다.

### 3.6.3 글꼴

셀 서식에서 글꼴은 셀 내용의 글자 모양, 크기, 굵기, 색 등을 설정한다.

- ◉ 글꼴은 글자체를 선택한다. 사용자 컴퓨터에 등록된 글자체는 모두 사용할 수 있다. 컴퓨터에 사용가능한 글자체 확인는 제어판 → 글꼴에서 한다.
  - ◉ 글꼴 스타일은 보통, 기울임꼴, 굵게, 굵게 기울임꼴 등 네 가지가 있다.
  - ◉ 크기는 글자의 크기로 자연수를 입력해야 하며 단위는 픽셀(pixel)이다.
  - ◉ 밑줄은 글자의 밑줄을 긋는 것으로 없음, 실선, 이중 실선, 실선(회계용), 이중 실선(회계용) 등이 있다.
  - ◉ 색은 글자 색을 설정한다.
  - ◉ 효과는 취소선, 아래 첨자, 위 첨자이 있다.
  - ◉ 미리보기는 설정된 내용을 바로 볼 수 있는 기능이다.
- 등이 있다.

#### 3.6.4 테두리

테두리는 워드프로세서의 표에 해당되는 것으로 선택한 셀의 윤곽선, 안쪽선, 사선을 긋고 선의 굵기와 선의 색깔을 조정한다.

#### 3.6.5 채우기

채우기는 선택한 셀에 대하여 음영을 주는 기능으로 색과 무늬로 설정한다. 색은 색상표에서 색을 선택하여 변경하고, 무늬는 무늬 리스트에서 셀에 표현할 무늬 모양을 선택하여 변경한다.

#### 3.6.6 셀 보호

셀 서식 → 보호 탭은 선택한 셀에 대하여 시트 보호를 설정할 때 적용된다. 이 기능은 해당 시트 또는 특정한 셀의 수정이나 변경 또는 저장 등을 제약하므로 작업 종료 전에 설정하는 것이 좋다. 시트 보호 설정은 홈 리본메뉴 → 셀

→ 서식 → 보호 → 시트보호로 한다. 설정된 시트 보호 해제는 홈 리본메뉴 → 셀 → 서식 → 보호 → 시트보호해제로 한다. 다음은 시트 보호 설정 과정이다.

1. 모든 셀은 잠금이 기본값으로 설정되어 있어 시트 보호를 실행하면 모든 시트의 셀에 적용된다. 따라서 특정한 셀만 보호하려면 시트의 모든 셀을 선택한 후 셀 서식 → 보호 탭에서 잠금을 해제한다(그림 3.1(f)).
2. 보호 할 셀을 선택한다.
3. 셀 서식 → 보호 탭에서 잠금을 선택한다.
4. 도구 → 보호 → 시트 보호 메뉴를 선택하면 그림 3.6.2의 창이 나타난다. 시트 보호 해제 암호에 암호를 입력하고, 잠긴 셀 선택과 잠기지 않은 셀 선택을 선택한 후 확인 버튼을 눌러 시트보호를 설정한다. 이 때 암호는 생략이 가능하다.

시트 보호 결과 보호된 셀은 여러 기능을 사용할 수 없게 설정되었으며 특히 잠김으로 설정된 셀은 편집할 수 없고 그 이외의 셀은 부분적으로 편집이 허용된다.

## 제 3.7 절 조건부 서식

조건부 서식은 어떤 조건을 만족하는 값에 셀 서식을 설정하는 것으로 셀 서식에서 소개한 것보다 더 동적으로 화려하게 표현한다. 조건부 서식 종류는 다음과 같다.

- 셀 강조 규칙 : 특정한 범위에 대하여 강조하려는 셀에 배경색을 칠한다.

- ✖ 보다 큼
- ✖ 보다 작음
- ✖ 다음 값의 사이에 있음

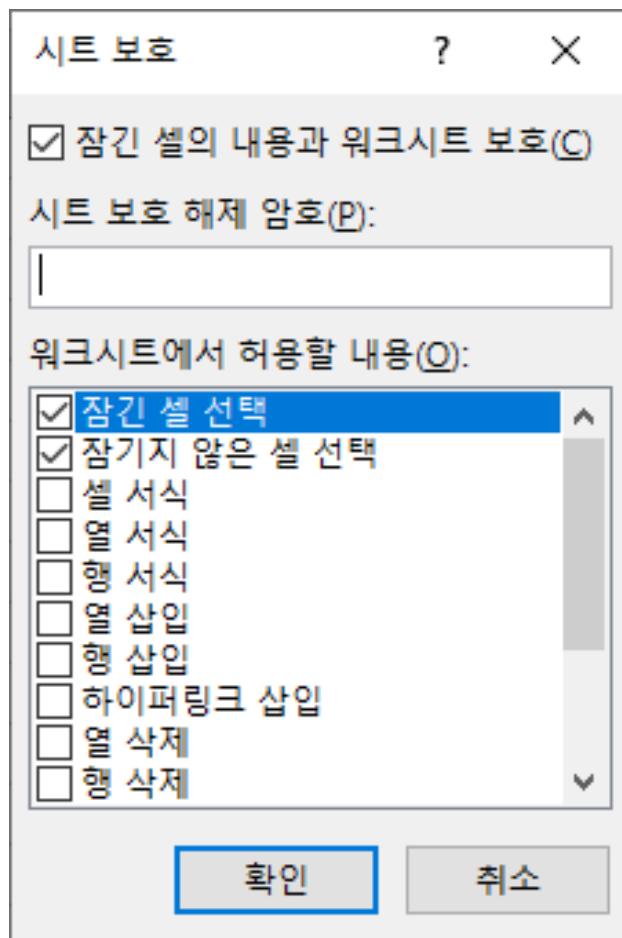


그림 3.6.2: 시트 보호

- ✖ 같음
- ✖ 텍스트 포함
- ✖ 발생 날짜
- ✖ 중복 값

- ◉ 상위/하위 규칙 : 상위 또는 하위 항목을 강조하려고 셀에 배경색을 칠 한다

- ✖ 상위 10개 항목
- ✖ 상위 10%
- ✖ 하위 10개 항목
- ✖ 하위 10%
- ✖ 평균 초과
- ✖ 평균 미만

◉ 아이콘 집합 : 선택한 셀의 값을 아이콘 집합으로 표시한다.

- ✖ 방향
- ✖ 도형
- ✖ 표시기
- ✖ 추천

◉ 데이터 막대 : 색이 지정된 데이터 막대를 셀 값에 추가하여 표시한다.

- ✖ 그라데이션 채우기
- ✖ 단색 채우기

◉ 색조 : 셀 범위에 단색 그라데이션을 적용한다.

조건부 서식을 적용해보자. 조건부 서식 실행은 홈 → 스타일 조건부 → 서식 메뉴를 선택한다. 이후 조건부 서식 종류를 선택하면 그림 3.7.1이 나타난다. 그림 3.7.2는 5 개 조건부 서식을 각 열마다 적용하였고 내용은 다음과 같다.

- ◉ B 열은 색조 → 녹색-노랑-빨강 색조
- ◉ C 열은 셀 강조 규칙 → 보다 큼 → 775
- ◉ D 열은 데이터 막대 → 그라데이션 채우기 → 파랑데이터 막대
- ◉ E 열은 아이콘 집합 → 추천 → 평점 5



그림 3.7.1: 조건부 서식 종류

- ⑥ F 열은 상위/하위 규칙 → 평균 초과 → 적용할 서식 : 진한 노랑 텍스트가 있는 노랑 채우기

## 제 3.8 절 셀에 메모 사용

메모는 특정한 셀에 설명하는 기능이다. 메모 실행은

마우스 오른쪽 버튼 → 메모 삽입

이나

검토 → 새 메모

로 한다. 실행 결과 나타난 메모 창에 설명할 메시지를 입력한다. 메모가 삽입된 셀은 오른쪽 위 부분에 빨간색 삼각형 표시가 나타난다(그림 3.8.1). 메모는 기본값이 메모가 있는 셀에 마우스가 놓여지면 내용이 나타나고, 셀을 벗어나면 내용이 나타나지 않는다. 메모를 항상 보이려면

	A	B	C	D	E	F
1	날짜	종가	전일비	등락률	거래량	순매매량(기)
2	2020-02-03	57,200	800	1.42%	23,995,260	1,076,953
3	2020-02-04	58,900	1,700	2.97%	21,800,192	-3,797,514
4	2020-02-05	59,500	600	1.02%	19,278,165	-4,459,274
5	2020-02-06	61,100	1,600	2.69%	14,727,159	2,483,655
6	2020-02-07	60,400	-700	-1.15%	16,402,493	-2,436,611
7	2020-02-10	59,700	-700	-1.16%	13,107,121	-263,054
8	2020-02-11	59,900	200	0.34%	11,071,231	1,889,247
9	2020-02-12	60,500	600	1.00%	12,904,207	1,516,828
10	2020-02-13	60,700	200	0.33%	18,449,775	-1,916,497
11	2020-02-14	61,800	1,100	1.81%	13,276,067	699,682
12	2020-02-17	61,500	-300	-0.49%	8,740,596	471,048
13	2020-02-18	59,800	-1,700	-2.76%	16,674,266	-2,458,387
14	2020-02-19	60,200	400	0.67%	12,951,496	697,276

그림 3.7.2: 조건부 서식 적용

마우스 오른쪽 클릭 → 메모 표시

를 선택한다(그림 3.8.1). 메모 편집은

마우스 오른쪽 클릭 → 메모 편집

을 실행하고 내용을 편집한다. 메모 삭제는

마우스 오른쪽 클릭 → 메모 삭제

로 한다.

## 제 3.9 절 셀에 하이퍼링크(hyperlink) 사용

셀에 하이퍼링크 삽입은

HEEMO KANG

엑셀 VBA

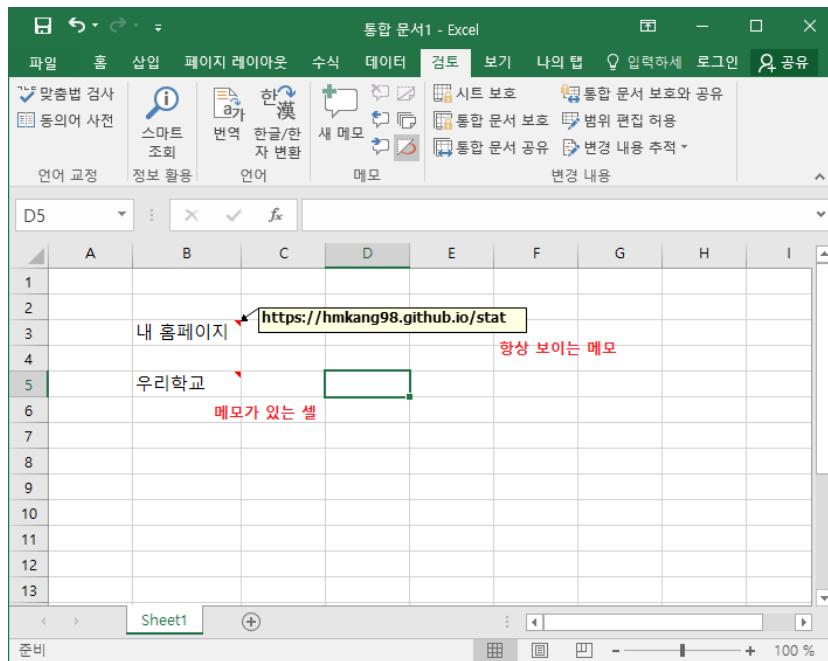


그림 3.8.1: 메모 삽입

- ◉ 셀에 직접 URL 주소나 메일 주소를 입력한다.
- ◉ 해당 셀을 선택하고 마우스 오른쪽 버튼 → 셀 서식 → 하이퍼링크를 클릭하면 하이퍼링크 추가 창이 나타난다. 이 창에서
  - ✖ 주소에 연결할 URI 주소를 입력하고
  - ✖ 표시할 텍스트에 하이퍼링크에 보여지는 글자를 입력한다.

이후 설정한 하이퍼링크는 셀을 클릭하면 연결된 주소로 이동한다.

## 제 3.10 절 셀에 수식 입력과 연산

셀은 수식을 입력하여 연산하는 기능이 있다. 셀에서 연산 실행은 맨 처음 등호 (=)를 입력하고 이후 식을 입력한다. 예를 들어 셀에

$5+3$ 은 ..... 5+3이 출력되고  
 $=5+3$ 은 ..... 8이 출력된다.  
 수식은 연산자와 피연산자가 있는데, 연산자는 수식의 필수요소로 계산의 종류를 나타낸다. 엑셀에서 연산자는 산술 연산자, 비교 연산자, 텍스트 연산자, 참조 연산자 등이 있다.

### 3.10.1 산술 연산자

산술 연산자는 더하기, 빼기, 곱하기, 나누기와 같은 기본 수치 연산을 수행하고 숫자를 결합하여 결과를 얻는데 사용한다. 산술 연산자와 의미, 예제 사용법은 표 3.10.1에 있다.

산술 연산자	의미	예
*	곱하기	$=3*3$
/	나누기	$=3/3$
+	더하기	$=3+3$
%	백분율	$=20\%$
-	빼기	$=3-1$
	음수	-1
$^$	지수	$=3^2(3*3과 같음)$

표 3.10.1: 산술 연산자

### 3.10.2 비교 연산자

비교 연산자는 두 개 이상 논리값을 비교한다. 결과는 TRUE와 FALSE 중 하나이다. 비교 연산자에 피연산자는 셀 번지, 숫자, 문자 등을 사용할 수 있다. 비교 연산자와 의미, 예제 사용법은 표 3.10.2에 있다.

비교 연산자	의미	예
=	같음	=A1=B1
<>	같지 않음	=A1<>B1
<=	작거나 같음	=A1<=B1
<	작음	=A1<B1
>=	크거나 같음	=A1>=B1
>	큼	=A1>B1

표 3.10.2: 비교 연산자

### 3.10.3 텍스트 연산자

텍스트 연산자는 여러 텍스트를 결합에 사용한다. 텍스트 연산자 기호는 &이다. 비교 연산자와 의미, 예제 사용법은 표 3.10.3에 있다.

텍스트 연산자	의미	예
&	두 값을 연결하여 연속하는 하나의 텍스트 값으로 만든다.	=대한 & 민국는 대한민국이 된다.

표 3.10.3: 텍스트 연산자

### 3.10.4 참조 연산자

참조 연산자는 여러 셀을 참조하여 연산 실행에서 사용한다. 참조 연산자는 콜론(:)과 컴마(,)가 있다. 사용법은 표 3.10.4에 있다. 또한 셀 참조 스타일에 따라 셀 자동채우기에서 계산이 다르게 되며 그 내용은 3.13 절에 소개되어 있다.

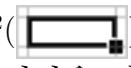
참조 연산자	의미	예
:	두 참조를 포함하여 두 참조 사이에 있는 셀을 모두 한 참조로 만드는 범위 연산자	=SUM(B5:B15)
,	여러 참조를 한 참조로 만드는 결합 연산자	=SUM(B5:B15, D5:D15)

표 3.10.4: 참조 연산자

## 제 3.11 절 셀 채우기

셀 채우기는 자동으로 데이터의 감소/증가 값을 계산하여 인접된 셀에 적용한다. 그 종류는

- ◉ 숫자, 날짜, 기타 항목 연속 데이터 채우기
- ◉ 사용자 지정 연속 데이터 채우기
- ◉ 행이나 열에서 데이터 복사

등이 있다. 이 기능 실행은 채우기 핸들<sup>2</sup>( )에 마우스 포인터를 놓고 마우스 끌기나 더블 클릭을 실행하여 결과값을 얻는다. 그림 3.1(a)에 연속 데이터를 채운 예가 있으니 확인하자.

### 3.11.1 숫자, 날짜, 기타 항목 연속 데이터 채우기

연속 데이터 채우기는 일정한 차이나 비율로 증가나 감소하는 값을 연속된 셀에 채운다. 연속 데이터의 기본 설정은 일정한 차이의 증가/감소이며 그 사용법은 다음과 같다.

<sup>2</sup>선택영역 오른쪽 하단에 있는 작고 검은 사각형으로, 채우기 핸들에 마우스가 위치하면 마우스 포인터가 검은 십자가 모양으로 바뀐다. 이 때 인접한 셀로 내용을 복사하거나 연속된 데이터를 채우려면 채우기 핸들을 마우스로 끈다.

1. 연속 데이터는 왼쪽, 오른쪽, 위 쪽, 아래 쪽 방향으로 채우기를 적용할 수 있다.
2. 연속 데이터 설정은 첫 셀에 시작 값을 입력하고 채우기가 적용될 방향으로 일정한 차이 값을 입력한다. 두 셀 값의 차가 연속 데이터의 증감값이 된다. 예를 들어 A 열 방향으로 연속 데이터를 적용하여 값을 2 만큼 증가하려면 A1 셀에 1을 입력하고 A2 셀에 3을 입력한다.
3. 시작 값이 있는 셀과 증감치를 적용할 셀을 선택한다. 예를 들어 설정한 셀 범위가 A1, A2이면 이 셀을 모두 선택한다.
4. 연속 데이터 실행은 선택된 셀의 채우기 핸들 포인터에서 연속 데이터 채우기가 적용될 방향의 범위까지 마우스로 드래그한다. 만일 채우기 핸들에서 마우스로 드래그 할 때 인접 셀에 데이터가 있으면 채우기 핸들에서 더블 클릭할 때 셀 값이 있는 길이까지 연속 데이터가 자동으로 실행된다.

예를 들어 A1에서 A1000까지 셀 값이 모두 채워져 있으면 B1열에서 B1000까지 연속 데이터 채우기하는 편리한 실행방법은 채우기 핸들 포인터에서 마우스를 더블 클릭하면 B1000 셀까지 자동으로 연속 데이터 실행으로 값이 채워진다.

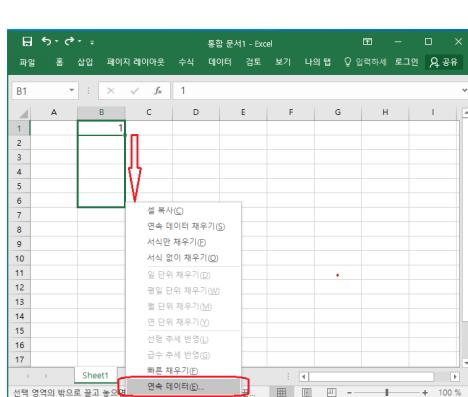
등차 수열을 포함한 등비수열 자료의 연속 데이터 실행은

1. 연속데이터가 적용될 첫 셀에 초기값을 입력하고
2. 채우기 핸들에서 마우스 오른쪽 버튼을 누르고 값을 채울 셀만큼 마우스 드래그 한 후 마우스를 떼면 여러 가지 메뉴가 나오며(그림 3.1(a)) 연속 데이터 메뉴를 선택한다.
3. 연속 데이터 창에서 실행가능한 연속 데이터 유형은
  - ① 선형 : 등차 수열(arithmetic sequence)
  - ② 급수 : 등비 수열(geometric sequence)

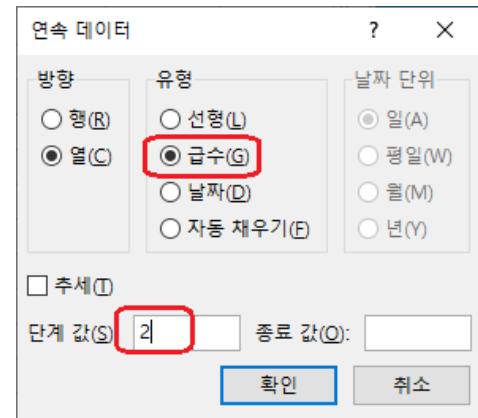
◉ 날짜 : 등차 수열

◉ 자동 채우기 : 등차가 1인 수열

이고(그림 3.1(b)) 단계값과 종료값을 입력하고 실행하면 연속 데이터가 채워진다. 그림 3.1(c)는 등비가 2인 급수의 연속 데이터를 적용하였다.



(a) 연속 데이터 실행



(b) 연속 데이터 창

	B	C	D	E	F	G	H	I
1	1							
2		2						
3			4					
4				8				
5					16			
6						32		

(c) 연속 데이터 결과

그림 3.11.1: 연속 데이터

### 3.11.2 사용자 지정 목록 채우기

이 기능은 기존 워크시트에 저장되어 있는 사용자 지정 목록이나 사용자가 직접 입력한 사용자 지정 목록을 설정하고 사용한다. 이미 설정된 사용자 지정 목록은

- ◉ 파일 → 옵션 → 고급 → 일반 → 사용자 지정 목록 편집

을 실행하여 그림 3.2(a)의 사용자 지정 목록에서 내용을 확인한다. 사용자가 원하는 목록은 목록 항목에 직접 입력한다. 새 목록 추가는

- ◉ 그림 3.2(a)에서 목록 항목에 값을 입력한다. 목록 구분은 콤마 또는 엔터이다. 입력을 마치면 추가 버튼을 클릭한다.
- ◉ 사용자 지정 목록에 셀 값 추가는 목록 가져올 범위에서 대화 상자 확장 버튼 ( ) 클릭, 셀 값 선택, 대화 상자 축소 버튼 ( ) 클릭한다. 목록이 목록 가져올 범위에 보인다. 가져오기 버튼을 클릭하면 목록이 추가된다.

그림 3.2(a)는 사용자가 추가한 사용자 지정 목록이고 그림 3.2(b)는 추가한 목록을 연속 데이터 채우기로 실행한 결과이다.

## 제 3.12 절 데이터 이동과 복사

셀은 수식, 값, 서식, 메모, 유효성검사, 테두리 등 많은 정보를 가지고 있다. 셀 값을 클립보드에 복사하고 붙여넣기를 실행하면 셀 값의 모든 정보가 복사된다. 셀 값을 옮기려면

- ◉ 복사 → 붙여넣기 : 원본과 대상본이 모두 존재한다.
- ◉ 잘라내기 → 붙여넣기 : 원본은 사라지고 대상본이 존재하며 마우스로 선택영역을 선택하고 이동하는 것과 동일하다.

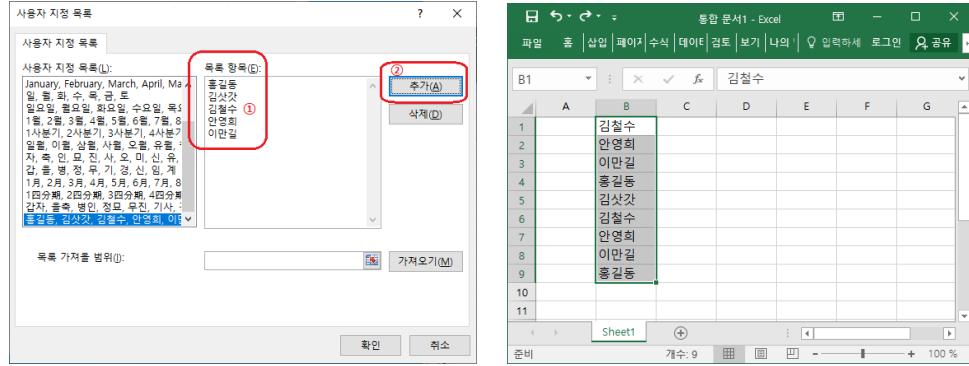


그림 3.11.2: 사용자 정의 연속 데이터

복사( )는 클립보드에 선택영역을 복사한다. 단축키는 CTRL + C이다. 잘라내기( )는 클립보드에 선택영역을 복사한다. 단축키는 CTRL + X이다. 붙여넣기( )는 클립보드에 복사나 잘라내기한 것을 붙여넣는다. 단축키는 CTRL + V이다. 붙여넣기는 선택한 복사 영역에 숨긴 셀이 있어도 모두 복사된다.

복사한 셀 정보에서 일부분만 붙여넣기를 할 수 있다. 선택하여 붙여넣기는 클립보드에 복사한 셀 값의 일부분만 붙여넣기한다. 즉 복사한 셀 값, 수식, 메모, 서식 중 선택한 정보를 붙여넣기한다. 선택하여 붙여넣기 기능은

- **붙여넣기** : 모두, 수식, 값, 서식, 메모, 유효성 검사, 테두리만 제외, 열너비에서 한 가지 붙여넣기한다.
- **연산** : 기본적으로 붙여넣기는 해당 셀 값이 있으면 지워지고 붙여넣기한 값이 복사된다. 선택하여 붙여넣기는 셀에 있는 값과 선택하여 붙여넣기 할 값이 서로 연산한다. 연산은 더하기, 빼기, 곱하기, 나누기가 있다. 사칙연산이므로 붙여넣기 할 값과 붙여넣을 값은 모두 숫자이다.
- **행열 바꿈** : 행과 열을 바꿔 붙여넣기한다. 행렬에서 전치행렬(transpose)이다.

- ◉ 공백 건너뛰기 : 공백은 잘라내고 붙여넣기한다.  
이다.

## 제 3.13 절 셀 참조

참조는 워크시트의 셀이나 셀 범위를 정의하고 수식에서 사용할 값이나 데이터를 찾는다. 참조는 수식에 데이터나 셀 값을 사용한다. 참조 종류는

- ◉ 같은 통합 문서에 있는 같은 시트의 셀
- ◉ 같은 통합 문서에 있는 다른 시트의 셀
- ◉ 다른 통합 문서에 있는 시트의 셀
- ◉ 다른 프로그램 데이터

가 있다. 다른 통합 문서의 셀 참조는 외부 참조라고 하고 다른 프로그램의 데이터 참조는 원격 참조라고 한다.

셀 참조 스타일은 A1 참조 스타일과 R1C1 스타일 두 종류가 있다.

- ◉ A1 참조 스타일은

❖ 문자로 열(A부터 XFD까지 전체 16,384열)을 나타내고 숫자로 행(1부터 1,048,576까지)

를 나타내며 엑셀에 기본으로 설정된 참조 스타일이다. 여기에서 문자는 열, 숫자는 행을 나타낸다. 셀 참조는 열 문자와 행 숫자를 입력하거나 해당 셀을 마우스로 선택한다. D50은 열 D와 행 50의 교점에 있는 셀을 참조한다. 셀 범위 참조는

범위의 첫 셀, 콜론(:), 범위의 마지막 셀

을 차례로 입력한다. 다음 그림은 선택한 셀 범위의 합이다.

	A	B	C	D	E	
1		국어	영어	수학	총점	
2	강감찬	100	100	100	=SUM(B2:D2)	
3	홍길동	90	90	90		
4	총점					

- R1C1 참조 스타일은 워크시트의 행과 열에 모두 숫자 번호인 참조 스타일로 매크로에서 행과 열 위치를 계산에 사용한다. 셀 위치는 R 다음에 행 번호와 C 다음에 열 번호를 지정한다. 이 참조 스타일 설정은 3.1 절을 참조하자.

또한 셀 참조는 수행할 작업에 따라

- 수식의 상대 위치로 셀을 참조하는 상대 참조
- 항상 고유의 셀 위치를 참조하는 절대 참조
- 두 참조를 모두 사용하는 혼합 참조가 있다.

참조하는 방법에 대하여 좀 더 자세히 알아보면

- 상대 참조는 참조하는 셀 범위가 상대적인 셀 주소이다. 현재 셀을 기준으로 좌, 우로 몇 번째 앞, 뒤 몇 번째에 해당되는지를 참조한다. 상대참조 값을 자동 채우기하면 셀 주소가 증가 또는 감소한다. 상대참조는 A1과 같이 설정한다.
- 절대 참조는 셀 범위가 절대로 변하지 않는 셀 주소이다. 이 참조는 셀의 절대적인 위치 참조로 자동 채우기나 연속 데이터를 실행할 때 참조하는 셀 위치가 변하지 않는다. 절대 참조 설정은 셀 행과 열 앞에 \$ 기호를 붙인다. 예를 들어 절대참조는 \$A\$1과 같이 설정한다.
- 혼합 참조는 절대 참조와 상대 참조를 혼용하여 사용하는 셀 참조이다.

A1 참조 스타일인 경우 다음과 같이 설정되어 있다면

- ◉ \$A\$1 : 열과 행 모두 절대참조로 자동채우기를 실행해도 참조하는 셀 주소가 변하지 않는다.
- ◉ \$A1 : 열은 절대참조 행은 상대참조로 설정되어 있기 때문에 열 방향 자동채우기는 셀 주소가 변하지 않고 행 방향 자동채우기는 행 주소가 하나씩 증가한다. 즉 행 앞에 \$를 붙이면 행에 대한 절대참조, 열 앞에 \$를 붙이면 열에 대한 절대참조가 설정된다.
- ◉ A\$1 : 열은 상대참조 행은 절대참조로 설정되어 있기 때문에 열 방향 자동채우기는 열 주소가 증가하고 행 방향은 변하지 않는다.

R1C1 참조 스타일이 다음과 같이 설정되어 있다면

- ◉ R1C1 : 행과 열 모두 절대참조로 1열 1행이다.
- ◉ R1C[1] : 행은 절대참조 열은 상대참조로 현재셀 기준 오른쪽 1열 1행이다.
- ◉ R[1]C1 : 행은 상대참조 열은 절대참조로 1열 현재셀 기준으로 아래쪽 1열이다.
- ◉ R[1]C[1] : 행과 열 모두 상대참조로 현재셀 기준으로 오른쪽 1열 아래쪽 1행이다.

상대 참조와 절대 참조의 설정을 변경하려면 현재셀에서 F4 키를 누른다. 그 때마다 절대 열과 절대 행(예: R1C1), 상대 열과 절대 행(R1C[1]), 절대 열과 상대 행(R[1]C1) 상대 열과 상대 행(R[1]C[1]) 순서로 조합이 변경된다.

## 제 3.14 절 시트 참조

시트 참조는 현재 셀에서 다른 시트의 셀을 직접 참조하거나 함수의 인수 값으로 사용한다. 현재 활성화 시트 셀에서 다른 시트 셀 참조는

- 현재 셀에서 참조하는 다른 시트를 마우스로 클릭한 후 해당 셀을 클릭하거나
- 현재 셀에서 '와' 사이에 시트 이름을 입력하고 !를 입력한 후 참조하는 셀 주소를 입력한다.

예를 들어 Sheet1의 A1에서 Sheet2 B1 참조는

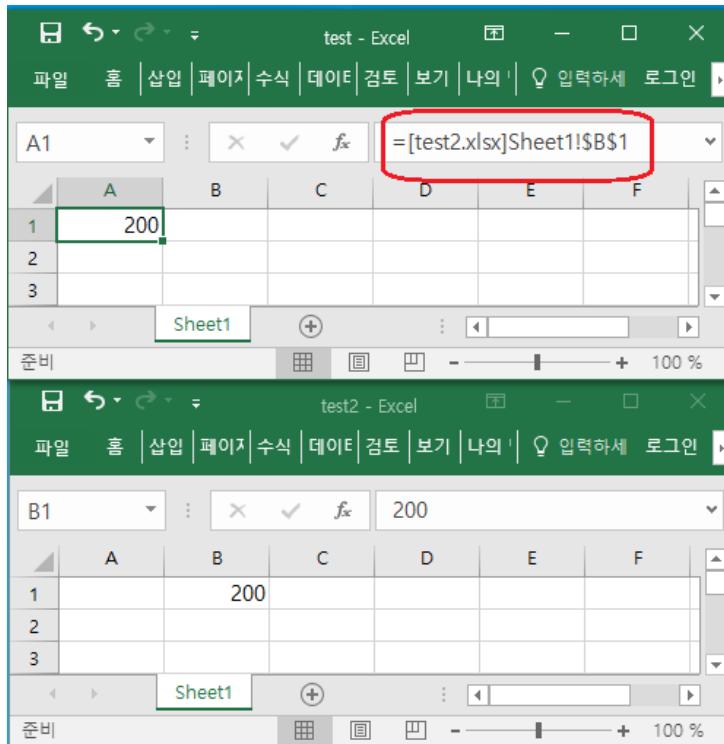
- 시트 Sheet1의 A1 셀을 클릭하고 등호(=) 입력, Sheet2 탭을 클릭, B1 셀을 클릭한다.
- 현재 셀에서 참조하는 셀이 속한 시트와 셀 번지 'Sheet2'!B1을 입력한다(그림 3.14.1).

The screenshot shows a portion of an Excel spreadsheet. The top row has column headers A, B, C, D, E. Row 1 has cell A1 highlighted in yellow and containing the value '100'. Row 2 has cell A2 containing '2'. Row 3 has cell A3 containing '3'. The formula bar at the top shows the formula '=Sheet2!B1'. The status bar at the bottom indicates 'A1'.

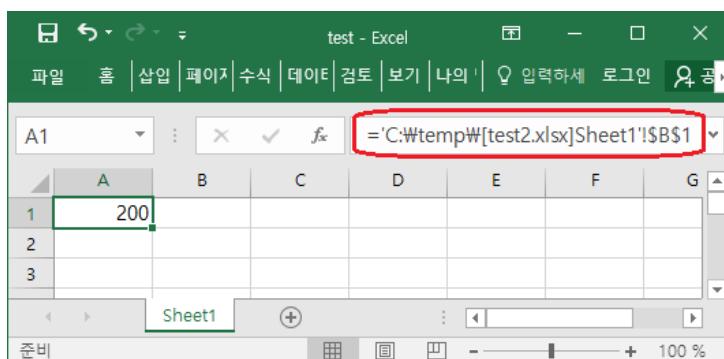
그림 3.14.1: 시트 참조

### 제 3.15 절 통합문서 참조

통합문서 참조는 다른 엑셀 파일의 셀을 참조한다. 통합문서 참조 실행과정은 시트 참조와 매우 비슷하다. 현재 셀에서 다른 엑셀 파일의 통합문서 참조는 '와' 사이에 파일경로[파일이름]시트이름을 입력하고 !를 입력한 후 참조하는 셀 주소를 입력한다. 여기서 파일경로는 참조하는 파일이 있는 경로(path)로 통합문서 참조하는 엑셀 파일이 있는 곳의 위치이다. 파일이름은 참조하는 통합문서를 가르킨다.



(a) 통합문서 참조(참조할 파일이 열린 경우)



(b) 통합문서 참조(참조할 파일이 닫힌 경우)

그림 3.15.1: 통합문서 참조

예를 들어 엑셀 파일 test.xlsx의 Sheet1 시트의 A1 셀에서 통합문서 참조할 셀이 엑셀 파일 c:\temp\test2.xlsx의 Sheet2 시트의 B1 셀이면

- ① 엑셀 파일 test.xlsx의 Sheet1 시트의 A1 셀에 현재 셀을 위치한 후 등호(=)를 입력하고 엑셀 파일 test2.xlsx의 Sheet2 시트의 B1 셀을 마우스로 선택한 후 엔터키를 누른다(그림 3.1(a)).
- ② 또는 현재 셀에서 참조하는 셀이 속한 파일의 절대경로, 시트와 셀 번지를 다음과 같이 등호(=)를 입력한 다음 'C:\temp\[test2.xlsx]Sheet2'!B1 을 입력한다(그림 3.1(b)).

## 연습문제

1. 현재 사용중인 엑셀의 한 시트에 열의 개수와 행의 개수가 몇 개인지 확인 후 적어라.
2. 엑셀에서 열 번호가 알파벳으로 되어 있을 때, 열 번호를 숫자로 나타나게 하려면 어떻게 해야 되는가?
3. 셀에 대하여 좌우 정렬을 적용한 경우가 아닐 때, 셀 값이 숫자인지 문자인지 구분하는 방법을 적어라.
4. 현재 시트의 맨 아래쪽 끝 셀로 이동하려면 어떻게 하는가?
5. 현재 시트의 맨 오른쪽 끝 셀로 이동하려면 어떻게 하는가?
6. “우리나라 대한민국”을 두 줄에 입력하려면 어떻게 하는가?
7. 셀에 숫자를 입력하였을 때 그 값이 80이상이면 파란색이면서 입력된 값과 “합격”, 80 미만이면 빨간색이면서 입력된 값과 “불합격” 문자가 나타나도록 설정하라.
8. 세 숫자 “0.631, 1234.59, 12”를 세 개의 셀에 입력한다. 모두 소수점 첫째자리까지 보이게 하고 유효하지 않은 자릿수는 0으로 채운다.

9. 위의 문제에서 사용한 숫자를 다른 셀에 복사한 후, 모두 소수점 첫째 자리까지 보이도록 설정하고 유효하지 않는 자릿수는 공간이 차지하게 한다.
10. 숫자를 입력하였을 때, 좌우 정렬을 하지 않아도 셀 값이 왼쪽에 정렬되도록 설정하라. 힌트는 셀 서식을 이용한다.
11. 현재 시트를 다른 사람이 편집하지 못하도록 설정하라. 읽기 암호는 1234, 쓰기 암호는 5678로 설정한다.
12. 현재 시트에서 “A1:B10” 셀까지는 편집할 수 있고, 나머지는 편집할 수 없도록 설정한다. 암호는 위의 문제와 동일하다.
13. C1 셀에 “한림대학교”를 입력한다. 이 셀을 클릭하면 한림대학교 홈페이지 “<https://www.hallym.ac.kr>”로 연결되도록 설정한다. 또한 마우스가 이 셀에 놓이면 “한림대학교 홈페이지로 연결됩니다” 메모가 나타나고, 마우스가 그 셀을 벗어나면 메모가 사라지도록 설정하라.
14.  $2^{20}$ 은 얼마인지 엑셀로 계산하라.
15. D1 셀에 “우리”, E1 셀에 “나라”를 입력하고 F1 셀에 두 문자를 합한 결과를 나타내라.
16. G1:G20 셀에 1에서 20까지 입력한다. H1:H20 셀에  $2^1$ 에서  $2^{20}$ 을 계산한다. 계산은 자동 채우기를 사용한다. I1:I20 셀에는  $2^1$ 에서  $2^{20}$ 을 계산한다. 이 계산은 셀 참조 기능을 사용하라.



## 제 4 장

# 함수 사용하기

엑셀에서는 함수는 여러 가지 연산을 수행하여 그 결과를 구하는 도구이다. 이 장은 다양한 함수의 사용법과 그 함수의 적용사례를 소개한다.

### 제 4.1 절 데이터 불러 오기

함수는 반드시 데이터가 필요한 것은 아니지만 많은 함수들이 데이터를 필요로 한다. 데이터가 대용량인 경우 DB에 저장하지만 보통 데이터는 파일로 제공한다.

데이터가 엑셀 파일이면 파일을 열어 직접 사용할 수 있지만 그 이외 파일은 엑셀 형식에 맞게 불러야 한다. 엑셀 파일 이외의 데이터 파일 중 많이 사용되는 파일 종류는 텍스트 파일이 있다. 엑셀에서 텍스트 파일을 열면 한 줄에 있는 데이터가 한 셀에 모두 들어간다. 따라서 한 셀에 들어간 데이터들은 각 데이터마다 공간이나 탭 같은 구분자를 사용하여 각각의 셀로 옮겨야 한다.

이 과정은 해당 셀을 선택하고 데이터 → 텍스트 나누기를 실행하면 각 셀에 데이터가 옮겨진다. 공개된 데이터 파일 중 german credit 데이터가 있다. 이 자료는 인터넷에서 얻을 수 있다. 이 파일은 텍스트 파일로 엑셀에서 이 파일을 불러보자.

1. 다운로드 한 german credit 텍스트 파일은 파일 → 파일 열기로 열거나

엑셀을 실행하고 텍스트 파일을 마우스로 누른 상태에서 엑셀 시트로 드래그하여 옮긴다.

2. 전에 불렀던 파일은 엑셀 형식에 맞게 열리기도 하지만 그렇지 않은 경우에는 모든 데이터가 한 열에 있다(그림 4.1.1). 한 열에 있는 여러

	A1	B	C	D	E	F	G	H
1	good.bad	checking	duration	history	purpose	amount	savings	employed
2	1 2 9.00	5 3 1919.00	1 4 4.00	3 1 3.00	3 35.00	3 1 1.00	3 1.00	2 1
3	2 1 18.00	3 1 1216.00	1 2 4.00	2 1 3.00	3 23.00	3 1 1.00	3 1.00	2 1
4	1 4 6.00	5 4 1382.00	1 3 1.00	2 1 1.00	3 28.00	3 2 2.00	3 1.00	2 1
5	1 4 12.00	3 1 1101.00	1 3 3.00	4 1 2.00	1 27.00	3 2 2.00	3 1.00	2 1
6	1 1 18.00	1 1 2249.00	2 4 4.00	3 1 3.00	3 30.00	3 2 1.00	4 2.00	2 1
7	1 4 36.00	5 2 10477.00	5 5 2.00	3 1 4.00	4 42.00	3 3 2.00	3 1.00	1 1
8	1 2 18.00	3 4 1113.00	1 3 4.00	2 3 4.00	1 26.00	3 2 1.00	2 2.00	1 1
9	2 1 40.00	6 7 5998.00	1 3 4.00	3 1 3.00	4 27.00	1 2 1.00	3 1.00	2 1
10	1 4 12.00	4 1 2133.00	5 5 4.00	2 1 4.00	4 52.00	3 3 1.00	4 1.00	2 1

그림 4.1.1: 엑셀에서 텍스트 파일을 부른 경우

개 데이터를 엑셀 형식에 맞게 각 셀로 옮기려면 그림 4.1.1과 같이 여러 데이터들이 있는 열을 선택하고 데이터 → 텍스트 나누기를 실행한다. 그러면 그림 4.1.2와 같은 텍스트 마법사 창이 나타난다.

3. 텍스트 마법사 1단계 : 그림 4.1.2와 같이 원본 데이터 형식을 결정하는 것으로 두 종류가 있다.

- 구분 기호가 분리됨 : 한 열에 있는 데이터는 쉼표나 탭, 공백 등과 같은 특수문자를 구분자로 각 셀에 옮긴다.
- 너비가 일정함 : 열 데이터를 일정한 너비로 분리한다.

4. 텍스트 마법사 2단계 – 그림 4.1.3

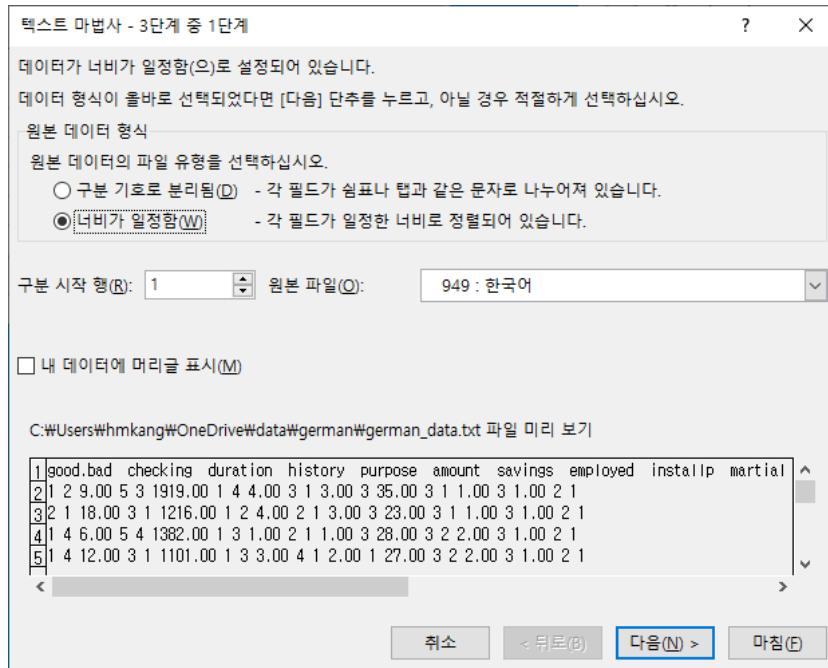
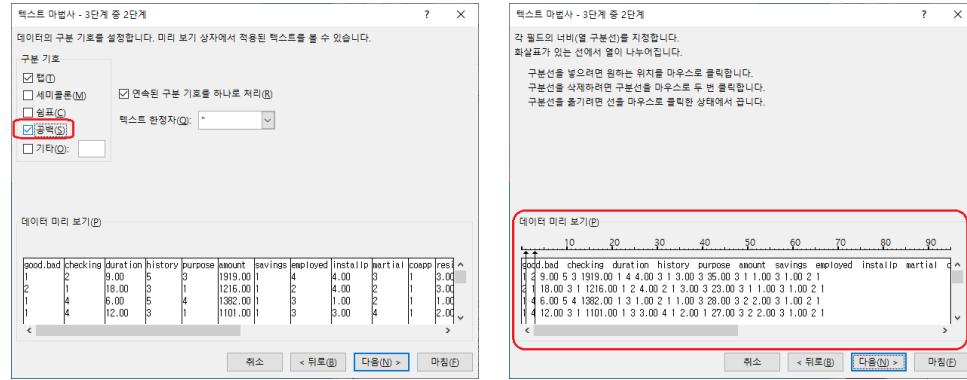


그림 4.1.2: 텍스트 마법사 1단계

- **구분 기호가 분리됨** : 텍스트로 된 셀은 구분자를 기준으로 나누어 각 셀에 데이터를 옮긴다. 구분기호는 탭, 세미콜론, 쉼표, 공백, 기타 등이며 여러 개 선택할 수 있다(그림 4.3(b)).
- **너비가 일정함** : 텍스트로 된 셀을 열 구분선 ↑로 구분하여 각 셀에 옮긴다. 열 구분선은 데이터 미리 보기에서 나눌 데이터를 클릭하면 ↑가 추가된다(그림 4.3(b)).

## 5. 텍스트 마법사 3단계 – 그림 4.1.4

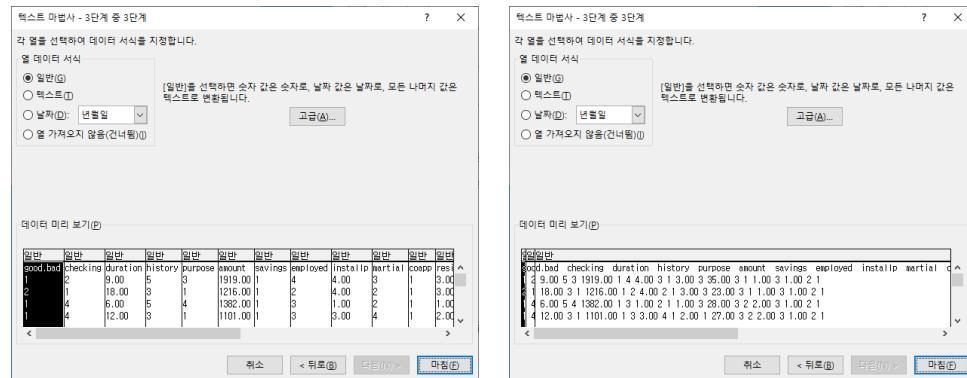
- **구분 기호가 분리됨** : 고급 버튼은 각 데이터의 숫자 서식을 설정하고 열 데이터 서식은 셀에 적용할 서식을 설정한다(그림 4.4(a)).
- **너비가 일정함** : 구분 기호가 분리됨과 설정방법이 동일하다(그림 4.4(b)).



(a) 구분 기호가 분리됨

(b) 너비가 일정함

그림 4.1.3: 텍스트 마법사 2단계



(a) 구분 기호가 분리됨

(b) 너비가 일정함

그림 4.1.4: 텍스트 마법사 3단계

6. 텍스트 마법사 과정이 모두 완료되면 데이터가 각 셀로 옮겨진다(그림 4.1.5).

	A	B	C	D	E	F	G	H
1	good.bad	checking	duration	history	purpose	amount	savings	employed in
2	1	2	9	5	3	1919	1	4
3	2	1	18	3	1	1216	1	2
4	1	4	6	5	4	1382	1	3
5	1	4	12	3	1	1101	1	3
6	1	1	18	3	1	2249	2	4
7	1	4	36	5	2	10477	5	5
8	1	2	18	3	4	1113	1	3
9	2	1	40	5	7	5998	1	3
10	1	4	12	3	1	2133	5	5

그림 4.1.5: 텍스트 마법사 완료

## 제 4.2 절 함수 사용방법

엑셀에 제공하는 함수는 약 400여개이다. 이 함수들은 여러 범주(category)로 나누어 있어 어떤 함수가 속한 범주를 알면 사용자가 쉽게 찾는다. 함수 사용법은

- ◉ 셀에 함수를 직접 입력하거나
- ◉ 함수 마법사  $f_x$ 를 사용한다.

### 4.2.1 함수를 직접 입력하는 방법

함수 사용은 셀에 다음과 같이 입력한다.

= 함수명(인수1, 인수2, ..., 인수n)

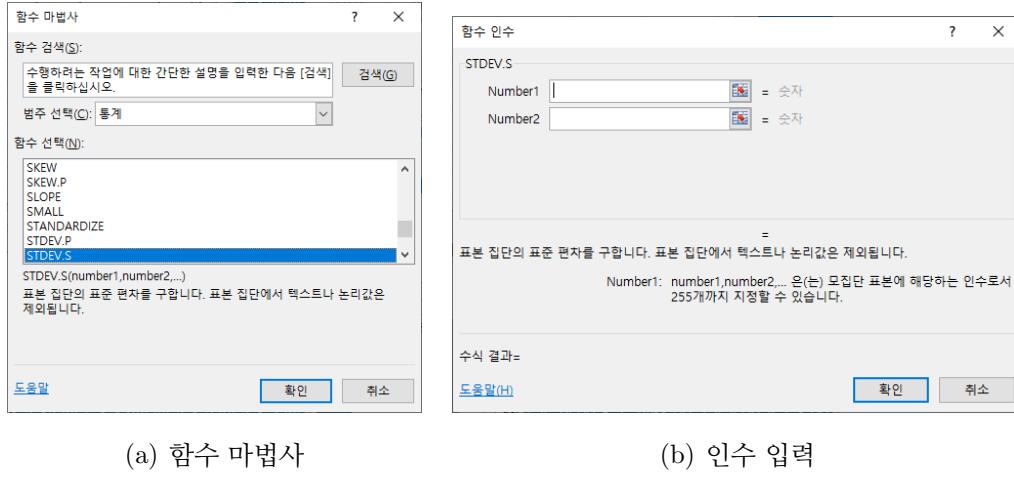


그림 4.2.1: 함수 마법사를 이용한 함수 사용방법

사용방법에서 괄호 안 인수는 입력하지 않아도 되는 함수부터 최대 255개까지 입력하고, 인수에 입력 값은 셀 참조나 직접 값을 입력한다. 인수에 셀 참조는 해당 셀을 마우스로 클릭하면 자동으로 입력된다. 통계에서 널리 알려진 표본의 표준편차는

$$\hat{\sigma} = \sqrt{\frac{\sum_{i=1}^n (X_i - \bar{X})^2}{n-1}}$$

이다. 엑셀에서 표준편차는 STDEV.S 함수로 구한다.

#### 보기 4.2.1. [함수 사용 예]

$$\begin{aligned}
 & \underbrace{=STDEV.S}_{\substack{\text{함수명} \\ }} \underbrace{(10, 20, 30, 40, 50)}_{\substack{\text{인수들} \\ }} \\
 & \underbrace{=STDEV.S}_{\substack{\text{함수명} \\ }} \underbrace{(A1:A5, D1:D5)}_{\substack{\text{인수들} \\ }}
 \end{aligned}$$

### 4.2.2 함수 마법사를 사용하는 방법

셀에 함수를 직접 입력하려면 사용할 함수 이름을 알아야 하지만 잘 모르면 함수 마법사를 실행하고 함수 리스트에서 사용할 함수를 클릭한다. 함수 마법사 실행은

- ◉ 수식 →  $f_x$  함수 삽입 메뉴를 선택하거나
- ◉ 도구모음에서  버튼을 클릭하거나
- ◉ 함수명을 직접 셀에 입력한 후 CTRL + A를 누른다.

그림 4.2.1은 함수 마법사로 표준편차를 구하는 과정이다. 함수에 사용할 데이터는 인수를 입력하는 입력창에 마우스를 클릭한 후

- ◉ 데이터를 직접 입력하거나
- ◉ 대화 상자 확장 버튼 을 클릭한다. 사용할 데이터가 있는 셀을 마우스로 드래그하여 셀을 선택한다. 대화 상자 축소 버튼 을 클릭하고

확인 버튼을 누르면 사용한 함수에 대한 결과를 얻는다.

## 제 4.3 절 함수의 종류

함수 마법사에 있는 함수 종류는 범주(category)로 제공한다. 함수 범주는

- ◉ 최근에 사용한 함수 : 최근에 사용한 함수 목록이다.
- ◉ 모두 : 함수 마법사에 등록된 모든 함수 목록이다.
- ◉ 재무 : 재무에 관련된 함수 목록이다.
- ◉ 날짜/시간 : 날짜와 시간에 관련된 함수 목록이다.
- ◉ 수학/삼각 : 수학에 관련된 함수와 삼각함수에 관련된 목록이다.

- 통계 : 통계에 사용되는 함수와 관련 목록이다.
- 찾기/참조영역 : 입력된 값이나 셀 번지에 대한 참조영역과 조건에 맞는 자료를 찾아주는 함수들과 관련된 목록이다.
- 데이터베이스 : 데이터베이스에 관련된 함수 목록이다.
- 텍스트 : 문자열에 관련된 함수 목록이다.
- 논리 : 논리값에 관련된 함수 목록이다.
- 정보 : 셀에 대한 정보 즉 숫자인지 문자인지, 셀이 비어 있는지 채워져 있는지, 숫자가 짝수인지 홀수인지, 에러가 발생하였는지, 발생된 에러는 어떤 것인지에 대한 정보로 대부분 is에 관련된 함수이다.
- 공학 : 공학에 관련된 함수들이 포함되어 있다.
- 호환성 : 기능이 향상된 함수의 정확성이나 예측 가능한 일관성이 있게 기존에 사용하던 함수의 범주를 변경했거나 이름이 바뀐 함수들이 있다.
- 큐브 : 큐브는 다차원의 셀로 이루어진 것으로 셀은 측정값의 그룹과 차원으로 구성되어 있다.
- 사용자 정의 : 사용자가 정의한 함수들 목록이다.
- 웹 : 웹 관련 함수 목록이다.

함수 범주에서 자주 사용하는 함수들을 소개한다.

## 제 4.4 절 논리 함수

논리 함수에서 가장 기본적인 TRUE, FALSE, AND, OR, NOT 함수에 대하여 알아보자.

#### 4.4.1 TRUE, FALSE 함수

먼저 TRUE와 FALSE는 논리값으로 다음과 같이 인수가 없는 함수이다.

TRUE() 또는 FALSE()

이 함수들은 직접 사용하지 않고 TRUE와 FALSE로 값을 셀에 입력하여 사용하는 경우가 더 많다.

#### 4.4.2 AND 함수

AND 함수는 인수의 논리값이 모두 TRUE이면 결과가 TRUE이고 하나라도 FALSE이면 결과가 FALSE이다. 사용 방법은

AND(*logical1, logical2, ...*)

이며, 인수 *logical1, logical2, ...*에는 1개에서 255개까지 사용할 수 있다. 인수를 사용할 때 주의사항은

- ◉ 인수는 TRUE 또는 FALSE와 같은 논리값으로 평가되거나 논리값이 들어 있는 배열<sup>1</sup> 또는 참조되어야 한다.
- ◉ 배열이나 참조 인수에 텍스트 또는 빈 셀이면 그 값들은 무시한다.
- ◉ 지정한 범위에 논리값이 없으면 #VALUE! 오류 값이 반환된다.

표 4.4.1의 예를 보자.

<sup>1</sup>여러 가지 결과나 행과 열로 구성되는 인수 그룹에 연산할 수 있게 한 개 수식을 작성한 것

수식	결과	설명
=AND(TRUE, TRUE, TRUE)	TRUE	인수가 모두 TRUE이다.
=AND(TRUE, TRUE, FALSE)	FALSE	인수 하나는 FALSE이다.
=AND(3+2=5, 5+3=8)	TRUE	인수가 모두 TRUE이다.
=AND(1<A2, A2<70)	TRUE	만일 A2=20이면 1과 70 사이에 있다.

표 4.4.1: AND 예

#### 4.4.3 OR 함수

OR 함수는 인수의 논리값이 모두 FALSE이면 결과가 FALSE이고 하나라도 TRUE이면 결과는 TRUE이다. 사용 방법은

`OR(logical1, logical2, ...)`

이며, 인수 *logical1, logical2, ...*에 1개에서 255개까지 사용한다. 인수 사용에서 주의사항은 AND 함수와 같다. 표 4.4.2의 예를 보자.

수식	결과	설명
=OR(FALSE, FALSE, FALSE)	FALSE	인수가 모두 FALSE이다.
=OR(TRUE, FALSE, FALSE)	TRUE	인수 하나는 TRUE이다.
=OR(3+2=7, 5+3=8)	TRUE	인수 하나는 TRUE이다.
=OR(1>A2, A2>70)	FALSE	A2=20이면 인수는 모두 FALSE이다.

표 4.4.2: OR 예

#### 4.4.4 NOT 함수

NOT 함수는 인수 논리값과 결과가 반대 논리이다. 이 함수는 주로 어떤 값이 특정 값과 같지 않은지 확인할 때 사용한다.

**NOT(*logical*)**

이면, *logical*에는 TRUE나 FALSE 값이나 식이다. 따라서 *logical*이 FALSE이면 결과가 TRUE이고, TRUE이면 결과는 FALSE이다. 표 4.4.3의 예를 보자.

수식	결과	설명
=NOT(FALSE)	TRUE	FALSE의 반대
=NOT(1+2=3)	FALSE	TRUE로 계산되는 수식의 반대

표 4.4.3: NOT 예

#### 4.4.5 IF 함수

논리 함수로 널리 사용되는 IF 함수에 대하여 알아보자.

**IF(*logical\_test*,*value\_if\_true*,*value\_if\_false*)**

*logical\_test* 인수 값이 TRUE나 FALSE 중 하나이다. 예를 들어 A10>50과 같은 논리식에서 A10 셀이 50 보다 크면 결과는 TRUE이면 50 보다 작거나 같으면 결과는 FALSE이다. 이 인수에는 대개 비교 연산자를 사용한다.

*value\_if\_true* 이 인수는 *logical\_test*가 TRUE이면 실행한다. 예를 들어 이 인수가 장년이라는 텍스트 문자열이면 *logical\_test* 인수가 TRUE일 때 IF 함수는 장년을 표시한다. *logical\_test*가 TRUE이고 *value\_if\_true* 가 공백이면 결과는 없다. *value\_if\_true*는 다른 수식을 사용할 수 있다.

*value\_if\_false* *logical\_test*가 FALSE이면 실행한다. 예를 들어 이 인수가 청년 문자열이면 *logical\_test* 인수가 FALSE일 때 IF 함수는 청년을 표시한다. *value\_if\_false*는 수식을 사용할 수 있다.

IF 함수를 사용한 예를 보자.

#### 보기 4.4.1. [IF 예]

=IF(A10>=50, "장년", "청년")

보기 4.4.1에서 A10 셀 값이 50보다 크면 장년, 그렇지 않으면 결과는 청년이다. 성적 입력에서 값이 올바르게 입력되었는지 IF로 알아보자.

#### 보기 4.4.2. [IF 예]

=IF(AND(0<A3, A3<100), A3, "범위를 벗어난 값입니다.")

보기 11.6.2는 A3 값이 0과 100 사이에 있으면 그 숫자를 표시하고 그렇지 않으면 결과는 세 번째 인수이다.

### 4.4.6 중첩 함수

함수 안에 인수로 함수를 여러 번 사용할 수 있다. 중첩함수는 함수 안에 다른 함수를 사용하는 것을 말한다. 함수 안에서 여러 번 함수를 사용할 때, 직접 입력해도 되지만, 함수 마법사로도 사용이 가능하다. 그림 4.4.1은 IF 함수 안에 AND 함수를 사용하는 방법으로 네 셀의 값이 모두 80 이상이면 “성적우수”, 아니면 “노력바람”이 출력된다.

1. 수식 → 함수 라이브러리 → 논리에서 IF를 선택한다. 중첩 함수가 추가될 IF 함수에서 첫 번째 인수를 선택하고 이름상자에서 AND 함수를 선택한다(그림 4.1(a)). 이름상자에 함수가 없으면 끝에 함수 추가를 클릭하여 추가할 함수를 선택한다.
2. 추가할 중첩 함수인 AND 함수 마법사에 인수 값 입력(그림 4.1(b)).
3. AND 함수에 인수를 모두 입력하였으면 수식입력줄에서 IF를 마우스로 클릭한다(그림 4.1(c)). IF 함수의 첫 번째 인수에 추가한 중첩함수인 AND 함수의 내용이 보인다.
4. 나머지 인수에 올바른 값을 입력한다(그림 4.1(d)).

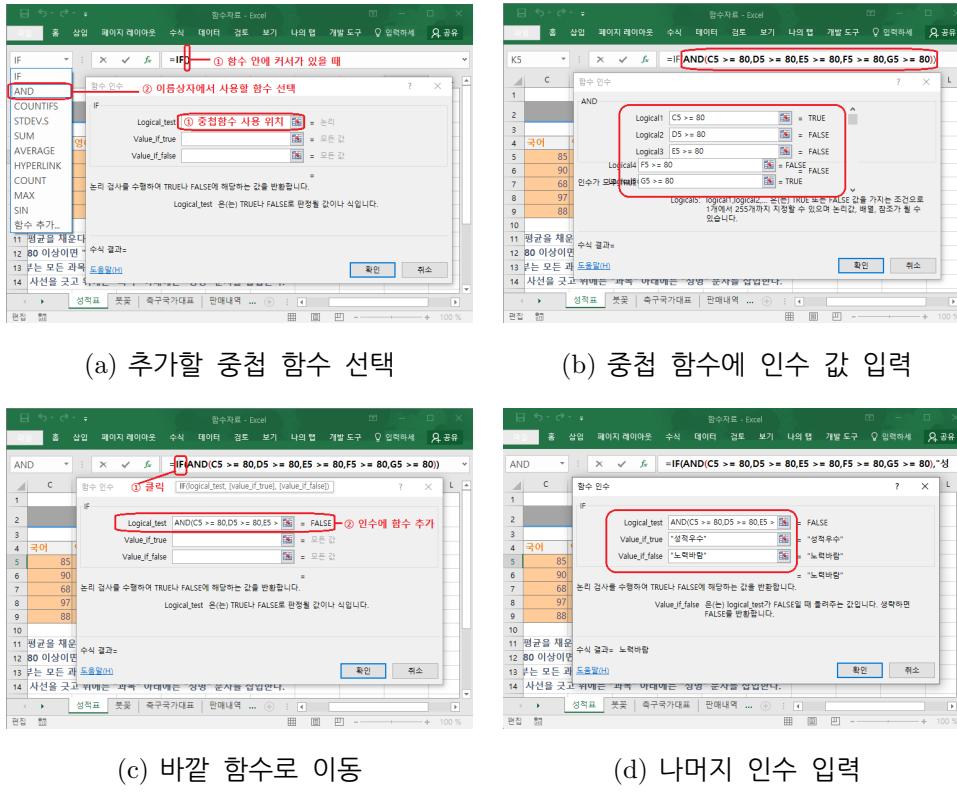


그림 4.4.1: 중첩 함수 사용

#### 4.4.7 COUNT 함수

COUNT 함수는 논리 함수는 아니지만 이후에 소개할 응용된 논리함수 COUNTIF 와 관련있기에 여기서 먼저 소개한다. 함수 사용법은

```
COUNT(value1,value2,...)
```

이며, 이 함수는 *value1, value2, ...*에 여러 데이터 형식이나 참조하는 인수를 255개까지 사용할 수 있고 숫자로 된 인수의 갯수를 계산한다. 다음 예를 보자.

#### 보기 4.4.3. [COUNT 예]

=COUNT(A1:A10)

A1에서 A10까지 숫자로 된 셀 개수를 계산한다. 또한 COUNTA 함수는 입력한 인수에 비어있지 않은 셀 개수를 계산한다.

#### 4.4.8 COUNTIF 함수

COUNTIF 함수는 조건에 맞는 셀의 개수를 반환한다. 이 함수 사용법은

`COUNTIF(range, criteria)`

이며, 함수에 사용하는 인수값은 다음과 같다.

*range* 셀 개수를 세려는 셀 범위이다.

*criteria* 개수를 세려는 셀을 정의하는 숫자, 식, 셀 참조 또는 텍스트의 형식으로 된 조건이다. 예를 들어 *criteria*는 46, "46", ">46", "바나나", "강??", "김\*" 등으로 입력한다.

다음 예를 보자.

#### 보기 4.4.4. [COUNTIF 예]

=COUNTIF(B2:B5, ">46")

보기 4.4.4는 B2에서 B5 셀까지 46보다 큰 셀 개수를 계산한다. IF가 포함된 함수는 SUMIF도 있으며 결과는 조건에 맞는 셀 값을 모두 합한다.

함수에서 인수 입력값이 논리 판별이고 문자열로 논리가 입력가능하면

- COUNTIF 함수처럼 셀 참조나 숫자를 직접 사용하거나
- 논리 판별에 문자열을 사용하려면 “와” 사이에 논리 조건을 입력한다. 예를 들어 논리 조건이 46 보다 크다면 ..... "> 46" 을 입력한다. 또한 셀 참조나 숫자, 문자 등이 혼합되었으면 문자열 결합

연산자 &으로 피연산자들을 연결한다. 또한 논리 조건이 A1에 5를 곱한 값보다 크면 ..... ">" & A1 \* 5로 입력한다.

- 논리 조건이 텍스트이면 와일드 카드 문자 \*, ?를 사용한다.

✖ \* : 문자 수에 제한이 없을 때 사용한다. 예를 들어 이름 중에서 강으로 시작하는 사람을 찾으려면 ..... "강\*"을 입력한다.

✖ ? : 정확한 문자 수를 찾을 때 사용한다. 예를 들어 강으로 시작하는 세 개 문자를 찾으려면 ..... "강??"를 입력한다.

#### 4.4.9 COUNTIFS 함수

COUNTIF 함수는 한 조건에 맞는 셀 갯수를 계산하고 COUNTIFS 함수는 여러 조건에 맞는 셀 갯수를 계산한다. 이 함수 사용법은

```
COUNTIFS(range1, criteria1, range2, criteria2, ...)
```

이며, 함수에 사용하는 인수값은 다음과 같다.

range1, range2, ... 셀 개수를 세려는 셀 범위이다.

criteria1, criteria2, ... 개수를 세려는 셀을 정의하는 숫자, 식, 셀 참조 또는 텍스트의 형식으로 된 조건이다.

다음 예를 보자.

##### 보기 4.4.5. [COUNTIFS 예]

```
=COUNTIFS(B2:B5, ">46", C2:C5, ">500")
```

보기 4.4.5는 B2:B5 셀 값이 46 초과하고 C2:C5 셀 값이 500 초과하는 셀의 갯수를 계산한다.

#### 4.4.10 SUM 함수

SUM 함수는 인수에 입력된 값이나 참조 값을 모두 합한다. SUM 함수도 논리 함수는 아니지만 SUMIF 함수와 관련있어 먼저 소개한다. 함수 사용법이다.

**SUM(*number1, number2, ...*)**

이 함수는 *number1, number2, ...*에 숫자나 참조하는 인수를 255개까지 입력하며 계산은 숫자만 포함된다. 표 4.4.4는 SUM 함수를 사용한 수식, 결과, 설명이다.

수식	결과	설명
=SUM(2,4,3)	9	2,4,3의 합계
=SUM(A2:A8)	7	모든 셀에 있는 값이 1이라면 합은 7

표 4.4.4: SUM 예

#### 4.4.11 SUMIF 함수

SUMIF 함수는 주어진 조건에 맞는 셀들의 합을 구한다. 함수 사용법이다.

**SUMIF(*range, criteria, sum\_range*)**

인수에 대한 설명이다.

*range* 조건을 적용할 셀 범위

*criteria* 숫자, 수식 또는 텍스트 형태로 된 조건으로 *criteria*는 46, "46", ">46", "바나나", "강??", "김\*" 등으로 입력한다.

	A 열	B 열	C 열
1 행	100,000	500	10
2 행	150,000	1,000	20
3 행	200,000	1,500	30
4 행	250,000	2,000	40
=SUMIF(A1:A4, ">150000", B1:B4)		계산 결과 3,500	
=SUMIFS(A1:A4, B1:B4, ">=1000", C1:C4, ">=30")		계산 결과 450,000	

표 4.4.5: SUMIF, SUMIFS 함수 사용 예

*sum\_range* 조건에 맞는 합을 구하는 셀이다. *range*의 셀이 찾을 조건과 일치하는 셀 값만 합한다. 만일 이 인수를 생략하면 *range*에 있는 셀 값을 합한다.

표 4.4.5는 SUMIF 함수를 사용한 수식, 결과, 설명이다. 표 4.4.5는 셀 A1:A4 범위에서 150000보다 큰 셀값을 B1:B4 범위에서 모두 합한다. B3이 1,500 B4가 2,000이므로 결과는 3,500이다.

#### 4.4.12 SUMIFS 함수

SUMIF는 하나의 조건에 맞는 셀들의 합을 구하고, SUMIFS 함수는 주어진 여러 개 조건에 맞는 셀들의 합을 구한다. 함수 사용법이다.

```
SUMIFS(sum_range, criteria_range1, criteria1, criteria_range2, criteria2, ...)
```

SUMIF와 SUMIFS 함수는 인수 배열순서가 다르므로 사용할 때 주의해서 사용해야한다. 인수에 대한 설명이다.

*sum\_range* 조건에 맞는 합을 구하는 셀이다. *criteria\_range1*, ...의 셀이 찾을 조건과 일치하는 셀 값만 합한다. 만일 이 인수를 생략하면 *range1*, ...에 있는 셀 값을 합한다.

*range1*, ... 조건을 적용할 셀 범위

*criteria1*, ... 숫자, 수식 또는 텍스트 형태로 된 조건으로 *criteria1*은 46, "46", ">46", "바나나", "강???", "김\*" 등으로 입력한다.

표 4.4.5에서 B1:B4셀 중 1,500 이상이고 C1:C4 셀 중 30 이상인 조건을 만족하는 A1:A4에서 합은 450,000이다. 그림 4.4.2에 함수마법사를 사용한 인수 입력과 계산결과가 있다.

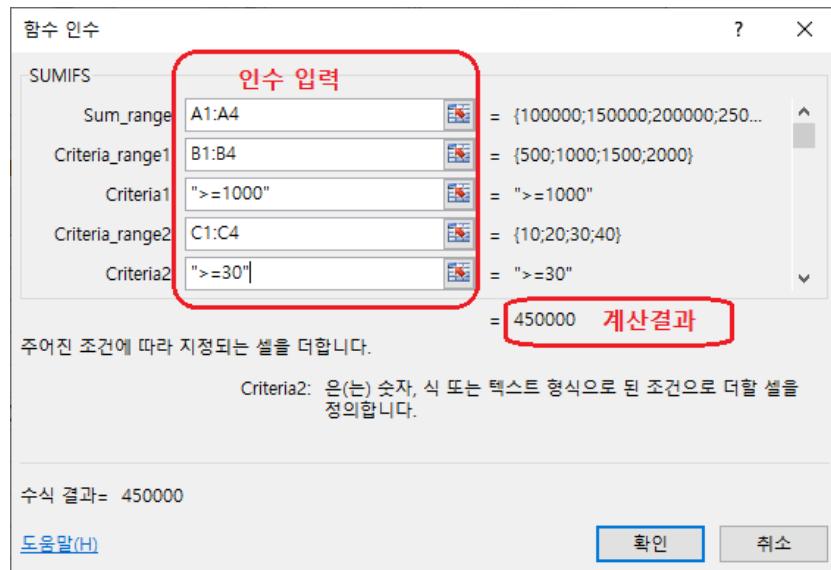


그림 4.4.2: SUMIFS 인수 입력

#### 4.4.13 AVERAGE 함수

AVERAGE 함수는 인수에 입력된 값이나 참조 값의 평균이다. AVERAGE 함수도 논리 함수는 아니지만 AVERAGEIF 함수와 관련있어 먼저 소개한다. 함수 사용법이다.

**AVERAGE(*number1, number2, ...*)**

이 함수는 *number1, number2, ...*에 숫자나 참조하는 인수를 255개까지 입력하며 계산은 숫자만 포함된다. 표 4.4.6은 AVERAGE 함수를 사용한 수식, 결과, 설명이다.

수식	결과	설명
=AVERAGE(2,4,3)	3	2,4,3의 평균
=AVERAGE(A2:A8)	7	모든 셀에 있는 값이 1이라면 평균은 1

표 4.4.6: AVERAGE 예

#### 4.4.14 AVERAGEIF 함수

AVERAGEIF 함수는 주어진 조건에 맞는 셀들의 평균을 구한다. 함수 사용법이다.

**AVERAGEIF(*range, criteria, average\_range*)**

인수에 대한 설명이다.

*range* 조건을 적용할 셀 범위

*criteria* 숫자, 수식 또는 텍스트 형태로 된 조건으로 *criteria*는 46, "46", ">46", "바나나", "강??", "김\*" 등으로 입력한다.

*average\_range* 조건에 맞는 평균을 구하는 셀이다. *range*의 셀이 찾을 조건과 일치하는 셀 값의 평균이다. 만일 이 인수를 생략하면 *range*에 있는 셀 값의 평균이다.

	A 열	B 열	C 열
1 행	100,000	500	10
2 행	150,000	1,000	20
3 행	200,000	1,500	30
4 행	250,000	2,000	40
=AVERAGEIF(A1:A4, ">150000", B1:B4)		계산 결과 1,500	
=AVERAGEIFS(A1:A4, B1:B4, ">=1000", C1:C4, ">=30")		계산 결과 225,000	

표 4.4.7: AVERAGEIF, AVERAGEIFS 함수 사용 예

표 4.4.7은 AVERAGEIF 함수를 사용한 수식, 결과, 설명이다. 표 4.4.7은 셀 A1:A4 범위에서 150000보다 큰 셀값을 B1:B4 범위에서 평균을 구한다. B2가 1,000, B3이 1,500, B4가 2,000이므로 결과는 이 값들의 평균인 1,500이다.

#### 4.4.15 AVERAGEIFS 함수

AVERAGEIF는 하나의 조건에 맞는 셀들의 평균을 구하고, AVERAGEIFS 함수는 주어진 여러 개 조건에 맞는 셀들의 평균을 구한다. 함수 사용법이다.

**AVERAGEIFS**(*average\_range*, *criteria\_range1*, *criteria1*, *criteria\_range2*,  
*criteria2*, ...)

AVERAGEIF와 AVERAGEIFS 함수는 인수 배열순서가 다르므로 사용할 때 주의 해서 사용해야한다. 인수에 대한 설명이다.

*average\_range* 조건에 맞는 평균을 구하는 셀이다. *criteria\_range1*, ...의 셀이 찾을 조건과 일치하는 셀 값의 평균이다.

*criteria\_range1*, ... 조건을 적용할 셀 범위

*criteria1, ...* 숫자, 수식 또는 텍스트 형태로 된 조건으로 *criteria1*은 46, "46", ">46", "바나나", "강???", "김\*" 등으로 입력한다..

표 4.4.7에서 B1:B4셀 중 1,500 이상이고 C1:C4 셀 중 30 이상인 조건을 만족하는 A1:A4에서 평균은 225,000이다. 그림 4.4.3에 함수마법사를 사용한 인수 입력과 계산결과가 있다.

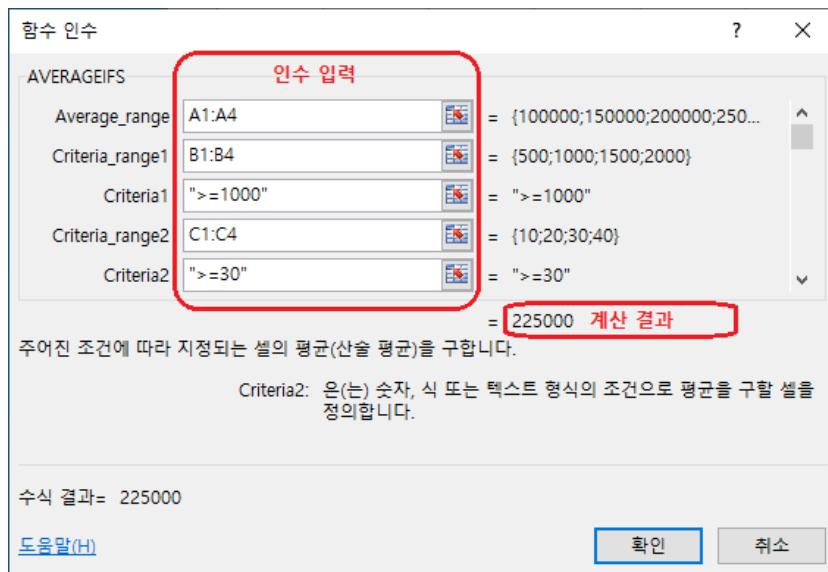


그림 4.4.3: AVERAGEIFS 인수 입력

## 제 4.5 절 날짜/시간 함수

엑셀에서 날짜와 시간에 대한 함수를 제공한다. 날짜는 년-월-일을 하이픈 (-)이나 쉼표(,), 슬래쉬(/)로 구분하여 입력한다. 날짜는 1900년 1월 1일부터 1씩 증가한다. 날짜에 대한 표시형식을 숫자로 변경하면 1900년 1월 1일부터 지난온 날짜수가 계산된다. 따라서 날짜도 가감 연산이 가능하다.

시간은 시:분:초를 콜론(:)으로 구분하여 입력한다. 시간도 숫자로 변경 할 수 있다. 날짜는 24시간이 1로 표현한다. 즉 12:00 AM은 0이고 12:00 PM

은 0.5이다. 1시간은 1/24로 약 0.0417이다.

### 4.5.1 TODAY 함수

TODAY 함수는 오늘 날짜를 보여준다. 오늘 날짜는 CTRL + : 키를 눌러도 셀에 나타난다. 이 함수는 입력 값이 없다.

### 4.5.2 NOW 함수

NOW 함수는 오늘 날짜와 시간을 보여준다. 오늘 날짜와 시간은 CTRL + SHIFT + : 키를 눌러도 셀에 나타난다. 이 함수는 입력 값이 없다.

### 4.5.3 YEAR 함수

YEAR 함수는 입력한 날짜에서 년도를 가져오는 함수이다. 함수 사용법이다.

`YEAR(serial_number)`

인수에 대한 설명이다.

*serial\_number* 년,월,일, 시간을 포함한 날짜 형식 자료

다음 예를 보자.

#### 보기 4.5.1. 【YEAR 예】

=YEAR(A1)

보기 4.5.1에서 A1 셀에 날짜가 ..... 2020-04-13  
이면 결과는 ..... 2020  
이다.

#### 4.5.4 MONTH 함수

MONTH 함수는 입력한 날짜에서 년도를 가져오는 함수이다. 함수 사용법이다.

```
MONTH(serial_number)
```

인수에 대한 설명이다.

*serial\_number* 년,월,일, 시간을 포함한 날짜 형식 자료

다음 예를 보자.

##### 보기 4.5.2. 【MONTH 예】

```
=MONTH(A1)
```

보기 4.5.2에서 A1 셀에 날짜가 ..... 2020-04-13  
이면 결과는 ..... 4  
이다.

#### 4.5.5 DAY 함수

DAY 함수는 입력한 날짜에서 년도를 가져오는 함수이다. 함수 사용법이다.

```
DAY(serial_number)
```

인수에 대한 설명이다.

*serial\_number* 년,월,일, 시간을 포함한 날짜 형식 자료

다음 예를 보자.

##### 보기 4.5.3. 【DAY 예】

```
=DAY(A1)
```

보기 4.5.3에서 A1 셀에 날짜가 ..... 2020-04-13  
 이면 결과는 ..... 13  
 이다.

#### 4.5.6 DATE 함수

DATE 함수는 인수에 입력한 년, 월, 일에 해당하는 특정 날짜를 계산한다. 함수 사용법이다.

`DATE(year, month, day)`

인수에 대한 설명이다.

*year* 년

*month* 월

*day* 일

다음 예를 보자.

##### 보기 4.5.4. [DATE 예]

=DATE(B1,C1,D1)

보기 4.5.4에서 B1 셀에 년도가 ..... 2020  
 C1 셀에 월이 ..... 4  
 D1 셀에 일이 ..... 13  
 이면 결과는 ..... 2020-04-13  
 이다.

#### 4.5.7 HOUR 함수

HOUR 함수는 입력한 시간 데이터에서 시간을 가져오는 함수이다. 함수 사용법이다.

HOUR(*serial\_number*)

인수에 대한 설명이다.

*serial\_number* 시, 분, 초를 포함한 시간 형식 자료

다음 예를 보자.

#### 보기 4.5.5. [HOUR 예]

=HOUR(A1)

보기 4.5.5에서 A2 셀에 시간이 ..... 12:59.00 PM  
이면 결과는 ..... 12  
이다.

#### 4.5.8 MINUTE 함수

MINUTE 함수는 입력한 시간 데이터에서 분을 가져오는 함수이다. 함수 사용 법이다.

MINUTE(*serial\_number*)

인수에 대한 설명이다.

*serial\_number* 시, 분, 초를 포함한 시간 형식 자료

다음 예를 보자.

#### 보기 4.5.6. [MINUTE 예]

=MINUTE(A2)

보기 4.5.6에서 A2 셀에 시간이 ..... 12:59.00 PM  
이면 결과는 ..... 59  
이다.

### 4.5.9 SECOND 함수

SECOND 함수는 입력한 시간 데이터에서 초를 가져오는 함수이다. 함수 사용법이다.

```
SECOND(serial_number)
```

인수에 대한 설명이다.

*serial\_number* 시, 분, 초를 포함한 시간 형식 자료

다음 예를 보자.

#### 보기 4.5.7. [SECOND 예]

```
=SECOND(A2)
```

보기 4.5.7에서 A2 셀에 시간이 ..... 12:59.00 PM  
이면 결과는 ..... 00  
이다.

### 4.5.10 TIME 함수

TIME 함수는 인수에 입력한 시, 분, 초에 해당하는 특정 시간을 계산한다. 함수 사용법이다.

```
TIME(year, month, day)
```

인수에 대한 설명이다.

*hour* 시

*minute* 분

*second* 초

다음 예를 보자.

#### 보기 4.5.8. 【TIME 예】

=TIME(B1,C1,D1)

보기 4.5.8에서 B2 셀에 시간이 .....	12
C2 셀에 분이 .....	59
D2 셀에 초가 .....	0
이면 결과는 .....	12:59:00 PM
이다.	

#### 4.5.11 WEEKDAY 함수

WEEKDAY 함수는 인수에 입력한 날짜에서 요일을 계산한다. 함수 사용법이다.

`WEEKDAY(serial_number, return_type)`

인수에 대한 설명이다.

*serial\_number* 날짜

*month* 요일에 대한 숫자를 결정하는 인수로 1,2,3 중 하나 사용

- ✖ 1은 일요일(1)에서 토요일(7)
- ✖ 2는 월요일(1)에서 일요일(7)
- ✖ 3은 월요일(0)에서 토요일(6)

다음 예를 보자.

#### 보기 4.5.9. 【WEEKDAY 예】

=WEEKDAY(A1,1)

보기 4.5.9에서 A1 셀에 날짜가 ..... 2020-04-13  
 이면 결과는 ..... 2  
 이다. CHOOSE 함수와 함께 사용하면 요일을 얻을 수 있다. CHOOSE(WEEKDAY(A1,1),"일","월","화","수","목","금","토")를 입력하면 결과는 ..... 월  
 이다. CHOOSE(WEEKDAY(A1,1),"일","월","화","수","목","금","토") 함수  
 식은 WEEKDAY(A1,1)가 1이면 "일", 2이면 "화", ..., 7이면 "토" 문자가 계산  
 된다. CHOOSE 함수 사용법은 154 면 4.11.3 절에 있다.

#### 4.5.12 WORKDAY 함수

WORKDAY 함수는 특정 요일에서 시작하여 주말과 지정한 공휴일을 제외한 일  
 수를 입력하면 해당 날짜를 계산한다. 함수 사용법이다.

`WORKDAY(start_date, days, holidays)`

인수에 대한 설명이다.

*start\_date* 시작 날짜

*days* 해당 날자일 수로 양수는 이후 날짜수이고 음수는 이전 날짜 수이다.

*holidays* 국경일, 공휴일, 임시 공휴일은 작업일수에서 제외한다. 생략하면 토,  
 일만 제외한다.

다음 예를 보자.

##### 보기 4.5.10. 【WORKDAY 예】

=WORKDAY(A1,40,A3:C3)

보기 4.5.10에서 A1 셀에 날짜가 ..... 2020-04-13  
 이고 A3에 총선 4월 15일, B3에 부처님 오신날 4월 30일, C3에 어린이날 5월  
 5일이 있으면 결과는 근무일수가 40일 지난 ..... 2020-06-11  
 이다.

#### 4.5.13 NETWORKDAYS 함수

NETWORKDAYS 함수는 특정 요일 범위에서 시작하여 주말과 지정한 공휴일을 제외한 일수를 계산한다. 즉 회사에서 일한 날짜수를 계산해준다. 함수 사용 법이다.

```
NETWORKDAYS(start_date, end_date, days, holidays)
```

인수에 대한 설명이다.

*start\_date* 시작 날짜

*end\_date* 종료 날짜

*holidays* 국경일, 공휴일, 임시 공휴일은 작업일수에서 제외한다. 생략하면 토, 일만 제외한다.

다음 예를 보자.

##### 보기 4.5.11. 【NETWORKDAYS 예】

```
=NETWORKDAYS(date(2020,4,1),date(2020,4,30),A3:A4)
```

보기 4.5.11에서 A3에 총선 4월 15일, B3에 부처님 오신날 4월 30일이 있으면 결과는 4월 근무일수는 ..... 20 일이다.

## 제 4.6 절 IS 함수

IS 함수는 인수 값에 대한 결과가 TRUE나 FALSE 중 하나이다. 또한 이 함수는 함수 사용중 오류발생도 감지한다. 다음은 함수 사용에서 발생하는 오류이다.

- ◉ #VALUE! : 인수나 피연산자 형식이 잘못된 경우 오류발생
- ◉ #REF! : 유효하지 않은 셀을 참조하는 경우 오류 발생

- #DIV/0! : 숫자를 0으로 나눈 경우 오류 발생(불능, impossible)
- #N/A : 함수나 수식에 사용할 수 없는 값을 입력하면 오류 발생
- #NUM! : 수식이나 함수 인수에 잘못된 숫자가 포함되어 있을 때 오류 발생
- #NAME? : 등록되지 않은 함수 사용에 오류 발생
- #NULL! : 잘못된 연산자 범위 사용이나 범위가 교차되지 않은 경우 오류 발생

다음은 IS 함수의 종류와 사용법이다.

#### 4.6.1 ISBLANK 함수

ISBLANK 함수는 셀 값이 비어있으면 TRUE 값이 있으면 FALSE이다. 다음 예를 보자.

##### 보기 4.6.1. [ISBLANK 예]

=ISBLANK(A1)

보기 4.6.1에서 A1 셀이 비어 있다면 결과는 .....TRUE이다.

#### 4.6.2 ISERR 함수

ISERR 함수는 셀 값이 #N/A를 제외한 #VALUE!, #REF!, #DIV/0!, #NUM!, #NAME?, #NULL! 오류값이면 TRUE 그렇지 않으면 FALSE이다.

#### 4.6.3 ISERROR 함수

ISERROR 함수는 셀 값이 #VALUE!, #REF!, #DIV/0!, #N/A, #NUM!, #NAME?, #NULL! 오류이면 TRUE 그렇지 않으면 FALSE이다.

#### 4.6.4 ISEVEN 함수

ISEVEN 함수는 셀 값이 짝수면 TRUE 홀수면 FALSE이다.

#### 4.6.5 ISLOGICAL 함수

ISLOGICAL 함수는 셀 값이 논리값이면 TRUE 논리값이 아니면 FALSE이다.

#### 4.6.6 ISNA 함수

ISNA 함수는 인수 값이 #N/A이면 TRUE 그렇지 않으면 FALSE이다.

#### 4.6.7 ISNONTEXT 함수

ISNONTEXT 함수는 셀 값이 텍스트가 아니면 TRUE 텍스트이면 FALSE이다.

#### 4.6.8 ISNUMBER 함수

ISNUMBER 함수는 셀 값이 숫자면 TRUE 그렇지 않으면 FALSE이다.

#### 4.6.9 ISODD 함수

ISODD 함수는 셀 값이 홀수이면 TRUE 짝수이면 FALSE이다.

#### 4.6.10 ISREF 함수

ISREF 함수는 셀 값이 셀 주소이면 TRUE 그렇지 않으면 FALSE이다.

#### 4.6.11 ISTEXT 함수

ISTEXT 함수는 셀 값이 텍스트이면 TRUE 값이 텍스트가 아니면 FALSE이다.

## 제 4.7 절 텍스트 관련 함수

텍스트 관련 함수는 텍스트 길이를 알아내거나 텍스트의 특정한 문자를 변경하거나 특정한 문자열을 얻을 때 사용한다.

### 4.7.1 CODE 함수

CODE 함수는 컴퓨터에 저장된 문자 코드 번호를 반환한다. 이 함수 사용법은

```
CODE(text)
```

이며, 결과는 인수 *text* 문자열의 첫 번째 코드번호이다. 이 함수의 사용 예로  
`=CODE("a")` ..... 97  
이다.

### 4.7.2 CHAR 함수

CHAR 함수는 코드 번호에 해당하는 ANSI 문자를 반환한다. 함수 사용법은

```
CHAR(number)
```

이며, 인수 *number*는 ANSI 코드번호이다. 이 함수의 사용 예로  
`=CHAR(97)` ..... a  
이다. CHAR 함수와 CODE 함수는 역함수 관계이다.

### 4.7.3 EXACT 함수

EXACT 함수는 두 텍스트 값이 동일한지 검사한다. 함수 사용법은

```
EXACT(text1, text2)
```

이며, 인수 *text1*는 첫 번째 문자열 *text2*는 두 번째 문자열이다. 이 함수 사용 예로 두 문자열이 같은지 함수를 실행한 결과

```
=EXACT("hello1", "hellol") ..... FALSE
```

이다. 두 문자열은 서로 다른 값이다.

#### 4.7.4 FIXED 함수

FIXED 함수는 숫자 표시 형식을 고정 소수점 텍스트로 지정한다. 함수 사용법은

```
FIXED(number, decimals, no_commas)
```

이며, 인수는

- ◉ *number*는 숫자로 변환할 값이다.
- ◉ *decimals*는 소수점 아래 유효 자리수로 음수를 입력하면 중수부분이 유효자리수가 된다.
- ◉ *no\_commas*는 숫자에 콤마를 사용할지 정한다. TRUE이면 숫자에 콤마를 사용하지 않는다.

이 함수 사용 예로

```
=FIXED(123.456, 2, TRUE) ..... 123.46
```

유효자리수가 소수점 둘째 자리고

```
=FIXED(123.456, -2, TRUE) ..... 100
```

유효자리수가 100 자리다.

#### 4.7.5 LEFT 함수

LEFT 함수는 텍스트 값 왼쪽부터 문자를 가져온다. 함수 사용법은

```
LEFT(text, num_chars)
```

이며, 인수는

- ◉ *text*는 텍스트이다.
- ◉ *num\_chars*는 왼쪽부터 가져올 문자의 개수이다.

함수 사용 예이다.

```
=LEFT("우리나라", 2) .....우리
```

#### 4.7.6 LEN 함수

LEN 함수는 텍스트 문자열의 문자 수를 계산한다. 함수 사용법은

```
LEN(text)
```

이며, 인수는

- ◉ *text*는 텍스트이다.

함수 사용 예이다.

```
=LEN("우리나라") ..... 4
```

#### 4.7.7 LOWER 함수

LOWER 함수는 텍스트를 소문자로 변환한다. 함수 사용법은

```
LOWER(text)
```

이며, 인수는

- ◎ *text*는 텍스트이다.

함수 사용 예이다.

```
=LOWER("ABCDE") ..... abcde
```

#### 4.7.8 MID 함수

MID 함수는 텍스트 문자열을 지정한 위치부터 지정한 개수의 문자를 가져온다.

함수 사용법은

```
MID(text, start_num, num_chars)
```

이며, 인수는

- ◎ *text*는 텍스트
- ◎ *start\_num*는 가져올 문자열 첫 번째 위치
- ◎ *num\_chars*는 가져올 문자열 개수

이다. 함수 사용 예이다.

```
=MID("우리나라", 3, 2) ..... 나라
```

#### 4.7.9 PROPER 함수

PROPER 이 함수는 텍스트의 첫 글자를 대문자로 바꾼다. 함수 사용법은

PROPER(*text*)

이며, 인수는

- ◉ *text*는 텍스트이다.

함수 사용 예이다.

=PROPER("abcde") ..... Abcde

#### 4.7.10 REPLACE 함수

REPLACE 함수는 문자열의 일부분을 바꾼다. 함수 사용법은

REPLACE(*old\_text*, *start\_num*, *num\_chars*, *new\_text*)

이며, 인수는

- ◉ *old\_text*는 바꿀 문자열
- ◉ *start\_num*는 바꿀 문자열에서 바꿀 문자열의 첫 번째 위치
- ◉ *num\_chars*는 바꿀 문자의 개수
- ◉ *new\_text*는 바꿀 문자열

이다. 함수 사용 예이다.

=REPLACE("우리나라", 3, 2, "가족") ..... 우리가족

### 4.7.11 REPT 함수

REPT 함수는 텍스트를 지정한 횟수만큼 반복한다. 함수 사용법은

```
REPT(text, times)
```

이며, 인수는

- ◎ *text*는 반복할 문자열
- ◎ *times*는 반복 횟수

이다. 함수 사용 예이다.

```
=REPT("우리", 5) .....우리우리우리우리우리
```

### 4.7.12 RIGHT 함수

RIGHT 함수는 텍스트 값에서 맨 오른쪽부터 문자를 가져온다. 함수 사용법은

```
RIGHT(text, num_chars)
```

이며, 인수는

- ◎ *text*는 텍스트이다.
- ◎ *num\_chars*는 왼쪽부터 가져올 문자의 개수이다.

함수 사용 예이다.

```
=RIGHT("우리나라", 2) .....나라
```

### 4.7.13 TRIM 함수

TRIM 함수는 텍스트에서 공백을 제거한다. 함수 사용법은

TRIM(*text*)

이며, 인수는

- ◉ *text*는 텍스트이다.

함수 사용 예이다.

=TRIM(" abcde ") ..... abcde

### 4.7.14 UPPER 함수

UPPER 함수는 문자열을 대문자로 바꾼다. 함수 사용법은

UPPER(*text*)

이며, 인수는

- ◉ *text*는 텍스트이다.

함수 사용 예이다.

=UPPER("abcde") ..... ABCDE

### 4.7.15 VALUE 함수

VALUE 함수는 텍스트 인수를 숫자로 바꾼다. 함수 사용법은

VALUE(*text*)

이며, 인수는

- ◎ *text*는 텍스트이다.

함수 사용 예이다.

```
=VALUE("123.456") ..... 123.456
```

## 제 4.8 절 행렬

행렬은 여러 셀과 연결하여 계산하므로 다른 함수 계산과는 그 과정이 다르다.  
행렬 연산은

1. 행렬 계산 결과가 출력될 셀 범위를 마우스로 드래그하여 설정한다.
2. 출력 영역에 행렬 함수를 입력하고 인수에 셀을 마우스로 선택한다. 이 때 주의사항은 마우스로 클릭하거나 방향 키를 누르면 출력결과 선택 영역이 해제되면서 행렬 연산을 할 수 없다.
3. CTRL + SHIFT + ENTER 키를 누른다.

### 4.8.1 전치행렬

행렬  $A$ 의 전치행렬(transpose matrix)은  $A^T$ 로 나타낸다.  $A$ 의 행이  $A^T$ 의 열이 되고  $A^T$ 의 열이  $A$ 의 행이 된다.

예를 들어 행렬  $A = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix}$  이면  $A^T = \begin{pmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{pmatrix}$  이다.

전치행렬을 구하는 함수는 TRANSPOSE이며 사용법은 다음과 같다.

```
TRANSPOSE(array)
```

이며, *array*는 행렬이다. 행렬  $A$ 의 전치행렬  $A^T$ 를 구해보자.

1. 행렬  $A$ 의 전치행렬  $A^T$ 의 결과가 출력될 셀을 마우스 드래그로 영역을 그림 4.1(a)처럼 한다.
2. 출력 영역을 선택하고 전치행렬 함수 **TRANSPOSE**(를 입력, 마우스로 행렬  $A$ 의 셀 영역을 선택하고 )를 입력한다(그림 4.1(a)). 주의할 점은 계산 과정에서 선택된 셀이 해제되지 않도록 한다.
3. **Ctrl + Shift + Enter** 키를 누르면 전치행렬  $A^T$ 의 결과가 그림 4.1(b)에 계산된다.



그림 4.8.1: 전치행렬 구하기

### 4.8.2 두 행렬의 곱

두 행렬의 곱은  $A_{m \times n} \times B_{n \times p}$ 이어야 한다. 즉 첫 번째 행렬 열 수  $n$ 과 두 번째 행렬 행 수  $n$ 이 같아야 한다. 두 행렬 곱 함수는 **MMULT**이며 사용법은 다음과 같다.

$\text{MMULT}(\text{array1}, \text{array2})$

이면,  $\text{array1}$ ,  $\text{array2}$ 는 행렬이다.

$$\text{행렬 } A = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix} \text{ 와 전치행렬 } A^T = \begin{pmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{pmatrix} \text{ 곱하자.}$$

1. 첫 번째는 행렬  $A$ 와 전치행렬  $A^T$ 를 곱한 결과가 출력될 셀을 마우스 드래그한다(그림 4.2(a)). 두 행렬 곱  $A_{2 \times 3}A_{3 \times 2}^T$ 의 계산 결과는  $2 \times 2$  행렬이다.
2. 계산 결과 셀 영역이 선택된 상태에서 두 행렬의 곱은 함수  $\text{MMULT}($ 를 키보드로 직접 입력, 마우스로 행렬  $A$  셀 영역과 전치행렬  $A^T$ 을 차례로 그림 4.2(a)와 같이 마우스 드래그로 인수를 채우고 )도 입력한다.
3. 끝으로 두 행렬 곱 실행은  $\text{Ctrl} + \text{Shift} + \text{Enter}$  키를 동시에 누른다. 그림 4.2(b)는 두 행렬 곱 계산 결과이다.

(a) 두 행렬 곱 함수 입력

(b) 두 행렬을 곱한 결과

그림 4.8.2: 두 행렬 곱 구하기

### 4.8.3 행렬식

행렬  $A_{n \times n}$ 의 행렬식(determinant)은  $\det(A)$  또는  $|A|$ 로 나타낸다. 행렬식은 다음과 같이 정의한다.

$$\det(A) = |A| = \sum_{i=1}^n a_{ij} c_{ij} = \sum_{j=1}^n a_{ij} c_{ij}, \quad i, j = 1, 2, \dots, n$$

여기서  $c_{ij}$ 는 공통인자(cofactor)이며 다음과 같이 정의한다.

$$c_{ij} = (-1)^{i+j} |A_{ij}|$$

$|A_{ij}|$ 는 행렬  $A$ 에서  $i$  번째 행과  $j$  번째 열을 제거한 행렬식이다. 공통인자  $c_{11}$ 은  $(-1)^{1+1} |A_{11}|$ 로 1행과 1열의 원소를 모두 제거하고 행렬식을 곱한 값이다. 행렬  $A_{3 \times 3}$ 의 원소가 다음과 같을 때

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}$$

$A$ 의 행렬식은 첫 행을 기준으로 했을 때

$$\begin{aligned} |A| &= \sum_{j=1}^3 = a_{1j} c_{1j} = a_{11} c_{11} + a_{21} c_{21} + a_{31} c_{31} \\ &= a_{11}(-1)^{1+1} |A_{11}| + a_{12}(-1)^{1+2} |A_{12}| + a_{13}(-1)^{1+3} |A_{13}| \\ &= a_{11} \begin{vmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{vmatrix} - a_{12} \begin{vmatrix} a_{21} & a_{23} \\ a_{31} & a_{33} \end{vmatrix} + a_{13} \begin{vmatrix} a_{21} & a_{22} \\ a_{31} & a_{32} \end{vmatrix} \\ &= a_{11}a_{22}a_{33} - a_{11}a_{23}a_{32} - a_{12}a_{21}a_{33} + a_{12}a_{23}a_{31} + a_{13}a_{21}a_{32} - a_{13}a_{22}a_{31} \end{aligned}$$

행렬식은 MDETERM 함수로 계산한다. 사용법은 다음과 같다.

MDETERM(array)

*array*는 행렬이다. 행렬  $\mathbf{A} = \begin{pmatrix} 1 & 0 & 1 \\ 2 & 1 & 2 \\ 0 & 4 & 6 \end{pmatrix}$ 에 대한 행렬식을 구해보자. 행렬식은  $1 \cdot 1 \cdot 6 - 1 \cdot 4 \cdot 2 - 0 \cdot 2 \cdot 6 + 0 \cdot 2 \cdot 0 + 1 \cdot 2 \cdot 4 - 1 \cdot 1 \cdot 0 = 6$ 이다. 엑셀에서 행렬식 계산은 그림 4.3(a)와 같이 =MDETERM(B2:D4)이다. 계산 결과는 직접 계산한 값과 같은 6이다.

#### 4.8.4 역행렬

역행렬(inverse matrix)은 행렬  $\mathbf{A}$ 와 행렬  $\mathbf{B}$ 를 곱하면 단위행렬(identity matrix)이 되는 행렬  $\mathbf{B}$ 이며  $\mathbf{A}^{-1}$ 로 나타낸다. 즉  $\mathbf{A}\mathbf{A}^{-1} = \mathbf{I}$  조건을 만족한다. 역행렬은 행렬식을 사용하여 구할 수 있으며 그 식은 다음과 같다.

$$\mathbf{A}^{-1} = \frac{1}{|\mathbf{A}|} (\text{adj } \mathbf{A})$$

$\text{adj } \mathbf{A}$ 는 공통인자 행렬의 전치행렬로 수반행렬(隨伴行列, adjoint matrix, adjugate matrix)이다. 즉  $\text{adj } \mathbf{A}$ 는  $C^T$ 이다. 만일 행렬이  $\mathbf{A}_{3 \times 3}$ 라면  $\text{adj } \mathbf{A}_{3 \times 3}$ 는 다음과 같이 공통인자로 구성된 행렬이다.

$$\text{adj } \mathbf{A} = \begin{pmatrix} c_{11} & c_{21} & c_{31} \\ c_{12} & c_{22} & c_{32} \\ c_{13} & c_{23} & c_{33} \end{pmatrix}$$

역행렬은 MINVERSE 함수로 계산한다. 사용법은 다음과 같다.

MINVERSE(*array*)

이며, *array*는 행렬이다. 행렬  $\mathbf{A} = \begin{pmatrix} 1 & 0 & 1 \\ 2 & 1 & 2 \\ 0 & 4 & 6 \end{pmatrix}$ 의 역행렬을 구해보자. 다음은 역행렬 계산과정이다.

$$\mathbf{C} = \left( \begin{array}{c|cc|c|cc|c|cc} (-1)^{1+1} & 1 & 2 & (-1)^{1+2} & 2 & 2 & (-1)^{1+3} & 2 & 1 \\ 4 & 6 & & 0 & 6 & & 0 & 4 & \\ \hline (-1)^{2+1} & 0 & 1 & (-1)^{2+2} & 1 & 1 & (-1)^{2+3} & 1 & 0 \\ 4 & 6 & & 0 & 6 & & 0 & 4 & \\ \hline (-1)^{3+1} & 0 & 1 & (-1)^{3+2} & 1 & 1 & (-1)^{3+3} & 1 & 0 \\ 1 & 2 & & 2 & 2 & & 2 & 1 & \end{array} \right)$$

$$\mathbf{C} = \begin{pmatrix} -2 & -12 & 8 \\ 4 & 6 & -4 \\ -1 & 0 & 1 \end{pmatrix}, \quad \mathbf{C}^T = \begin{pmatrix} -2 & 4 & -1 \\ -12 & 6 & 0 \\ 8 & -4 & 1 \end{pmatrix}$$

$$\mathbf{A}^{-1} = \frac{1}{|\mathbf{A}|} (\text{adj } \mathbf{A}) = \frac{1}{|\mathbf{A}|} \mathbf{C}^T = \frac{1}{6} \begin{pmatrix} -2 & 4 & -1 \\ -12 & 6 & 0 \\ 8 & -4 & 1 \end{pmatrix} = \begin{pmatrix} -\frac{2}{6} & \frac{4}{6} & -\frac{1}{6} \\ -\frac{12}{6} & \frac{6}{6} & \frac{0}{6} \\ \frac{8}{6} & -\frac{4}{6} & \frac{1}{6} \end{pmatrix}$$

다음은 엑셀에서 역행렬 계산과정이다.

1. 첫 단계는 그림 4.3(a)와 같이 행렬  $\mathbf{A}$ 의 결과가 출력될 셀을 마우스 드래그한다.
2. 계산 결과가 출력될 영역을 선택, 역행렬을 구하는 함수 `MINVERSE(B2:D4)`를 그림 4.3(a)와 같이 입력한다. 주의사항은 입력 중 키보드나 마우스 조작으로 선택영역이 해제되지 않도록 한다.
3. 계산은 `Ctrl + Shift + Enter`를 누른다. 그림 4.3(b)는 역행렬  $\mathbf{A}^{-1}$ 의 계산결과이다.

The screenshot shows two side-by-side Excel windows. The left window (a) shows the formula =MINVERSE(B2:D4) entered into cell B2, with a green border around the range B2:D4. The right window (b) shows the result of the formula, which is the inverse of the matrix, displayed in cells G2:H4.

	A	B	C	D	E	F	G	H	I
1									
2	1	0	1			=MINVERSE(B2:D4)			
3	2	1	2			MINVERSE(array)			
4	0	4	6						
5									
6									
7									
8									

(a) 역행렬 함수 입력

(b) 역행렬 결과

그림 4.8.3: 역행렬 구하기

## 제 4.9 절 수학·삼각 함수

수학 함수와 삼각 함수에 대하여 알아보자.

### 4.9.1 삼각 함수에서 유용한 함수

각은 호도(radian)와 각도(degree;°)가 있다. 호도는 RADIANS 함수, 각도는 DEGREES 함수로 계산한다.

**DEGREES(*angle1*) 또는 RADIANS(*angle2*)**

DEGREES 함수는 인수 *angle1*에 호도를 입력하면 각도를 계산하고, RADIANS 함수는 *angle2*에 각도를 입력하면 호도를 계산한다. 표 4.9.1을 보자. PI 함수는 인수값이 없으면 무리수  $\pi$  값을 계산한다.

수식	결과	설명
=DEGREES(PI()/2)	90	$\frac{\pi}{2}$ 의 각도
=RADIANS(180)	3.141593	180°의 호도

표 4.9.1: 도(angle)의 예

### 4.9.2 SIN 함수

SIN 함수는 삼각 함수에서 sin을 계산한다. 인수는 호도(radian)를 입력한다. 함수 사용 방법이다.

**SIN(number)**

이 함수는 인수 *number*에 호도를 입력한다. 만일 각도를 입력하려면 RADIANS 함수를 사용하거나  $\frac{PI()}{180} \times$  각도를 곱하여 호도로 변경하여 입력한다. 표 4.9.2의 예를 보자.

수식	결과	설명
=SIN(PI()/2)	1	$\sin(\frac{\pi}{2})$
=ASIN(1)	1.570796	1의 아크사인에 대한 라디안 = $\frac{\pi}{2}$
=SIN(30*PI()/180)	0.5	$\sin(30^\circ)$
=SIN(RADIAN(30))	0.5	$\sin(30^\circ)$
=ASIN(0.5)	0.523599	0.5의 아크사인에 대한 라디안 = $\frac{\pi}{6}$

표 4.9.2: SIN, ASIN의 예

### 4.9.3 ASIN 함수

ASIN 함수는 sin 함수의 역함수를 구한다. 함수 사용방법은 다음과 같다.

**ASIN(*number*)**

인수 *number*는  $\sin$  값 범위  $-1 \sim 1$ 를 입력한다. 구하려는 각이 각도(degree; $^\circ$ )면 DEGREES 함수를 사용하거나  $\frac{180}{\text{PI}()} \times$  계산결과를 곱한다. 예를 들어 다음과 같이 입력하면

=DEGREES(ASIN(0.5)) ..... 30

=180/PI()\*ASIN(0.5) ..... 30

이 출력된다.

#### 4.9.4 COS 함수

COS 함수는 삼각 함수에서  $\cos$ 을 계산한다. 인수는 호도(radian)를 입력한다. 함수 사용방법이다.

**COS(*number*)**

이 함수는 인수 *number*에 호도를 입력한다.

#### 4.9.5 ACOS 함수

ACOS 함수는  $\cos$  함수의 역함수를 구한다. 함수 사용방법은 다음과 같다.

**ACOS(*number*)**

인수 *number*는  $\cos$  값 범위  $-1 \sim 1$ 를 입력한다.

#### 4.9.6 TAN 함수

TAN 함수는 삼각 함수 tan를 계산한다. 인수는 호도(radian) 입력을 입력한다. 함수 사용방법은 다음과 같다.

**TAN(*number*)**

이 함수는 인수 *number*에 호도를 입력한다.

#### 4.9.7 ATAN 함수

ATAN 함수는 tan 함수의 역함수를 구한다. 함수의 결과 값은  $-\frac{\pi}{2}$ 에서  $\frac{\pi}{2}$ 이다. 함수 사용방법은 다음과 같다.

**ATAN(*number*)**

인수 *number*는 tan 값 범위  $-\infty \sim \infty$ 를 입력한다.

#### 4.9.8 COMBIN 함수

COMBIN 함수는 순열(combination)을 계산한다. 순열은

$${}_nC_r = \frac{n!}{(n-r)! \times r!} = \frac{n(n-1) \times \cdots \times (n-r+1)}{r!} \quad (4.1)$$

으로 나타낸다. 함수 사용방법은 다음과 같다.

**COMBIN(*number*, *number\_chosen*)**

이 함수는 인수 *number*에 식 4.1의 *n* *number\_chosen*에 식 4.1의 *r*을 입력한다. 다음은 COMBIN 함수를 사용한 예이다.

=COMBIN(5,2)은 .....  $\frac{5 \times 4}{2 \times 1} = 10$ 이다.

### 4.9.9 EXP 함수

EXP 함수는 무리수  $e$ 의 누승을 구한다. 함수 사용방법은 다음과 같다.

`EXP(number)`

이 함수는 인수 *number*에 누승을 입력한다. 다음은 EXP 함수 예제이다.

=EXP(2)은 .....  $e^2 = 7.3891$ 입니다.

### 4.9.10 GCD 함수

GCD 함수는 최대공약수를 구한다. 함수 사용방법은 다음과 같다.

`GCD(number1, number2, ...)`

인수 *number1, number2, ...*는 숫자를 입력한다. 인수 개수는 255개까지 입력할 수 있다. 다음은 GCD 함수를 사용한 예이다.

=GCD(24, 36)의 최대공약수는 ..... 12이다.

### 4.9.11 LCM 함수

LCM 함수는 최소공배수를 구한다. 함수 사용방법은 다음과 같다.

`LCM(number1, number2, ...)`

인수 *number1, number2, ...*는 최소공배수를 구하려는 숫자들을 입력하며 255 개까지 입력할 수 있다. 다음은 LCM 함수를 사용한 예이다.

=LCM(24, 36)의 최소공배수는 ..... 72이다.

### 4.9.12 INT 함수

INT 함수는 소수점 아래는 버리고, 가장 가까운 정수로 숫자를 내린다. 함수 사용방법은 다음과 같다.

`INT(number)`

이 함수는 인수 *number*에 소수를 입력한다. 다음은 INT 함수를 사용한 예이다.

=INT(2.4)는 ..... 2.4에서 내림값 2이고  
=INT(-2.4)는 ..... -2.4에서 내림값 -3이다.

### 4.9.13 로그 함수

- LN 함수는 밑(base)이 무리수  $e$ 인 자연로그값을 구하며 EXP 함수와 역함수 관계이다.
- LOG 함수는 입력한 밑과 진수에 대한 값을 구하고,
- LOG10 함수는 밑이 10인 상용로그 값을 구한다.

이 함수들의 사용방법은 다음과 같다.

`LN(number)`

`LOG(number, base)`

`LOG10(number)`

이 함수는 인수 *number*에 로그값을 구할 양의 실수를 입력하고, *base*에는 밑을 입력한다. 다음은 LN, LOG, LOG10 함수의 사용 예이다.

=LN(100)은 ..... 100에 대한 자연로그값 4.60517이다.  
=LOG(100,2)은 ..... 밑이 2인 100에 대한 로그값 6.64386이다.  
=LOG10(100)은 ..... 100에 대한 상용로그값 2이다.

#### 4.9.14 자릿수 조정 함수

- ◉ ROUND 함수는 반올림 기능을 제공한다.
- ◉ ROUNDDOWN 함수는 설정한 자리수에서 버림 기능을 제공하고
- ◉ ROUNDUP 함수는 설정한 자리수에서 올림 기능을 제공한다.

이 함수들의 사용방법은 다음과 같다.

```
ROUND(number, num_digits)  
ROUNDDOWN(number, num_digits)  
ROUNDUP(number, num_digits)
```

이 함수는 인수

- ◉ *number*에 자리수를 조정할 실수를 입력하고
- ◉ *num\_digits*에는 조정할 자리수를 입력하는데 양수는 소수점이하이고 음수는 실수부분이 자리수에 해당된다.

다음은 ROUND, ROUNDDOWN, ROUNDUP 함수를 사용한 예이다.

=ROUND(3.141592, 3)은 ..... 3.142

로 3.141592를 소수점 넷째 자리에서 반올림하여 유효숫자가 소수점 이하 세째 자리가 되고,

=ROUND(12345, -2)는 ..... 12300

로 12345를 십 자리에서 반올림하여 유효숫자가 백의 자리다.

=ROUNDDOWN(3.141592, 3)은 ..... 3.142

로 내림의 유효숫자가 소수점 이하 세째 자리이다.

=ROUNDDOWN(12345, -2)는 ..... 12300

으로 내림의 유효숫자가 백의 자리이다.

=ROUNDUP(3.141592, 3)은 ..... 3.142

로 올림의 유효숫자가 소수점 이하 세째 자리이다.

=ROUNDUP(12345, -2)는 ..... 12400

으로 올림의 유효숫자가 백의 자리이다.

#### 4.9.15 FACT 함수

FACT 함수는 계승(factorial)을 구한다.  $n$ 의 계승은

$$n! = n \times (n - 1) \times \cdots \times 2 \times 1 \quad (4.2)$$

로 구한다. 함수 사용방법은 다음과 같다.

**FACT(*number*)**

이 함수는 인수 *number*에 구하려는 계승에 대한 값을 입력한다. 다음은  $5!$ 을 구한 예이다.

=FACT(5)는 .....  $5 \times 4 \times 3 \times 2 \times 1 = 120$ 이다.

#### 4.9.16 SQRT 함수

SQRT 함수는 입력한 값에 대한 양의 제곱근을 구한다. 함수 사용방법은 다음과 같다.

**SQRT(*number*)**

이 함수는 인수 *number*에 실수를 입력한다. 다음은  $\sqrt{2}$ 를 구한 예이다.

=SQRT(2)는 ..... 2의 양의 제곱근 1.414이다.

### 4.9.17 합계 함수

합계를 구하는 함수는 자료의 형태에 따라 여러 가지로 제공되며 그 종류는 다음과 같다.

- ◉ SUMPRODUCT 함수는  $\sum_{i=1}^n X_i Y_i$ 로 배열에 대응하는 값과 서로 곱하고 합을 구한다.
- ◉ SUMSQ 함수는  $\sum_{i=1}^n X_i^2$ 로 제곱합을 구한다.
- ◉ SUMX2MY2 함수는  $\sum_{i=1}^n X_i^2 - \sum_{i=1}^n Y_i^2$ 로 두 배열 대응하는 값에 제곱을 합하고 두 합의 차이를 계산한다.
- ◉ SUMX2PY2 함수는  $\sum_{i=1}^n X_i^2 + \sum_{i=1}^n Y_i^2$ 로 두 배열의 대응값의 제곱을 구하고 합을 구한다.
- ◉ SUMXMY2 함수는  $\sum_{i=1}^n (X_i - Y_i)^2$ 로 두 배열에 대응값에 차이를 구하고 제곱합을 구한다.

다음은 함수들의 사용방법이다.

```
SUMPRODUCT(array1, array2, ...)
SUMSQ(number1, number2, ...)
SUMX2MY2(array_x, array_y)
SUMX2PY2(array_x, array_y)
SUMXMY2(array_x, array_y)
```

인수 *array1*, *array2*과 *array\_x*, *array\_y*는 배열이며 그 길이는 모두 같고 *number1*, *number2*는 실수를 입력한다. 표 4.9.3 예를 보자.

	A 열	B 열
1 행	1	2
2 행	2	4
3 행	3	6
4 행	4	8
=SUMPRODUCT(A1:A4,B1:B4)	계산 결과 $60 = 1 \times 2 + \dots + 4 \times 8$	
=SUMSQ(A1:A4)		계산 결과 $30 = 1^2 + \dots + 4^2$
=SUMX2MY2(A1:A4,B1:B4)		계산 결과 $-90 = 1^2 - 2^2 + \dots + 4^2 - 8^2$
=SUMX2PY2(A1:A4,B1:B4)		계산 결과 $150 = 1^2 + 2^2 + \dots + 4^2 + 8^2$
=SUMXMY2(A1:A4,B1:B4)		계산 결과 $30 = (1 - 2)^2 + \dots + (4 - 8)^2$

표 4.9.3: 합계 함수 예

## 제 4.10 절 통계 함수

통계 함수는 통계학에서 사용하는 함수의 범주이다.

### 4.10.1 대푯값에 대한 함수

대푯값은 관심있는 집단에서 자료를 수집하여 요약한 값이다. 통계에서 사용하는 대푯값은 평균(average), 중앙값(median), 최빈값(mode) 등이 있다. 다음은 사용법이다.

- AVERAGE 함수는 인수에 대한 산술평균을 구한다. 계산에 적합하지 않은 문자나 논리값은 개수에서 제외하고 계산한다.
- AVERAGEA 함수는 인수에 인수에 대한 평균을 구한다. AVERAGE와 다른 점은 평균 계산에서 인수에 결측치나 문자 논리값이 있더라도 전체 자료 수에 포함한다.
- GEOMEAN 함수는 기하평균(geometric mean)을 계산한다. 계산에 적합하

지 않은 문자나 논리값은 그 자료를 개수에서 제외하고 계산한다.

$$GM_{\bar{y}} = \sqrt[n]{y_1 y_2 \cdots y_n} \quad (4.3)$$

- ◉ HARMEAN 함수는 조화평균(harmonic mean)을 계산한다. 계산에 적합하지 않은 문자나 논리값은 그 자료를 개수에서 제외하고 계산한다.

$$\frac{1}{H_{\bar{y}}} = \frac{1}{n} \sum \frac{1}{Y_i} \quad (4.4)$$

- ◉ TRIMMEAN 함수는 주어진 자료에서 상위, 하위  $\alpha\%$ 를 제외하고 평균을 계산한다. 이 평균은 절삭평균(trimmed mean)이라고 부른다.
- ◉ MEDIAN 함수는 중앙값을 계산한다. 문자나 논리값이 포함되어 있으면 그 자료는 개수에서 제외하여 계산한다.
- ◉ QUARTILE 함수는 사분위수를 구한다.
- ◉ MODE 함수는 자료에서 빈도수가 가장 큰 값 즉 가장 많이 발생하는 값을 계산한다. 자료에 문자나 논리값이 포함되어 있다면 그 자료는 제외하고 계산한다.

다음은 함수 사용방법이다.

```
AVERAGE(number1, number2, ...)
AVERAGEA(value1, value2, ...)
GEOMEAN(number1, number, ...)
HARMEAN(number1, number2, ...)
TRIMMEAN(array, percent)
MEDIAN(number1, number2, ...)
QUARTILE(array, quart)
MODE(number1, number2, ...)
```

인수 *number1*, *number2*, ...는 숫자만 사용하면 문자가 사용하면 계산에서 제외하고 *value1*, *value2*, ...에는 배열이나 문자 등 어떤 자료형태도 사용할 수 있다. 인수는 최대 255개까지 입력이 가능하다.

#### TRIMMEAN 함수에서 인수

- ◉ *array*는 자료의 배열,
- ◉ *percent*는 절삭 평균 계산에서 제외할 자료 비율로 만일 자료가 10개 일 때 0.2는 상위 1개와 하위 1개를 제외한 8개에 대한 평균이다.

#### QUARTILE 함수에서 인수

- ◉ *array*는 자료의 배열,
- ◉ *quart*는
  - ✖ 0 : 최솟값
  - ✖ 1 : 제 1 사분위수로 상위 25% 값
  - ✖ 2 : 제 2 사분위수로 상위 50% 값으로 중앙값과 동일
  - ✖ 3 : 제 3 사분위수로 상위 75% 값
  - ✖ 4 : 최댓값

이다.

데이터 1, 2, 3, 4, 5, 6, 7, 8, 9, 100가 A1:A10에 있을 때 10% 절삭 평균은 상위 10%와 하위 10%를 제외한 것으로

=TRIMMEAN(A1:A10, 20) ..... 5.5

이다. 또한 이 자료에 대한 제 1 사분위수는

=QUARTILE(A1:A10, 1) ..... 3.25

가 된다. 함수 사용 예가 표 4.10.1에 더 있다.

	A 열	B 열
1 행	1	1
2 행	2	2
3 행	3	A
=AVERAGE(A1,A2,A3)	계산 결과 $2 = (1 + 2 + 3)/3$	
=AVERAGE(B1,B2,B3)	계산 결과 $1.5 = (1 + 2)/2$	
=AVERAGEA(A1,A2,A3)	계산 결과 $2 = (1 + 2 + 3)/3$	
=AVERAGEA(B1,B2,B3)	계산 결과 $1 = (1 + 2 + 0)/3$	
=GEOMEAN(A1,A2,A3)	계산 결과 $1.8171 = \sqrt[3]{1 \times 2 \times 3}$	
=HARMEAN(A1,A2,A3)	계산 결과 $1.6364 = \frac{1}{\left(\frac{1}{1} + \frac{1}{2} + \frac{1}{3}\right)}$	
=MEDIAN(A1,A2,A3)	계산 결과 2 = 두 번째 값	
=MODE(1,2,2)	계산 결과 2 = 2가 두개	

표 4.10.1: 대푯값 함수 예

### 4.10.2 자료의 흩어짐(dispersion)에 대한 함수

평균에서 자료가 얼마나 흩어져 있는지 알려주는 함수에 대하여 알아보자. 평균에 대한 흩어짐에 대한 척도는 분산, 표준편차 등이 있다. 사용법은 다음과 같다.

- ◉ VAR.P 함수는 모집단의 분산을 계산한다.
- ◉ VAR.S 함수는 표본집단의 분산을 계산한다.
- ◉ STDEV.P 함수는 모집단의 표준편차를 계산한다. 논리값이나 텍스트는 자료에서 제외하고 표준편차를 계산한다.
- ◉ STDEV.S 함수는 표본집단의 표준편차를 계산한다.
- ◉ STDEVPA 함수는 모집단의 표준편차를 계산한다. 논리값이나 텍스트도 자료의 수에 포함한다.

- STDEVA 함수는 표본집단의 표준편차를 계산한다. 인수에 논리값이나 텍스트도 자료의 수에 포함한다.

분산과 표준편차 함수들의 사용방법은 다음과 같다.

```
VAR.P(number1, number2, ...)
VAR.S(number1, number2, ...)
STDEV.P(number1, number2, ...)
STDEV.S(number1, number2, ...)
STDEVPA(number1, number2, ...)
STDEVA(number1, number2, ...)
```

인수 *number1, number2, ...*는 숫자를 포함한 이름, 배열, 참조영역을 사용한다. 빈 셀, 논리값, 텍스트, 오류 등은 계산에서 무시된다. 그러나 인수목록에 직접 입력하는 논리값, 텍스트로 나타낸 숫자 등은 계산에 포함된다. 인수는 254개까지 입력이 가능하다. 표 4.10.2의 예를 보자.

### 4.10.3 기술 통계량(descriptive statistics)

기술 통계량은 자료로부터 계산된 통계량으로 4.10.1에서 소개한 대푯값과 4.10.2에서 소개한 분산, 표준편차 등과 최솟값, 최댓값, 범위 등이 있다. 기술 통계량에 대하여 알아보자.

- MAX 함수는 인수에 입력된 자료중 최댓값(maximum)을 구하며 계산에 적합하지 않은 문자나 논리값이 포함되어 있으면 계산에서 제외한다.
- MAXA 함수는 인수에 입력된 자료중 최댓값(maximum)을 구하며 계산에 적합하지 않은 논리값이 포함되어 있을 때 그 값이 TRUE면 1 FALSE나 텍스트이거나 값이 없는 빈 셀이면 0이 반환된다.
- MIN 함수는 인수에 입력된 자료중 최솟값(minimum)을 구하며 계산에 적합하지 않은 문자나 논리값이 포함되어 있으면 계산에서 제외한다.

	A 열
1 행	1
2 행	2
3 행	3
4 행	A
=VAR.P(A1, A2, A3, A4)	계산 결과 1.6667
=VAR.S(A1, A2, A3, A4)	계산 결과 1
=STDEV.P(A1, A2, A3, A4)	계산 결과 0.8165
=STDEV.S(A1, A2, A3, A4)	계산 결과 1
=STDEVAP(A1, A2, A3, A4)	계산 결과 1.1180
=STDEVA(A1, A2, A3, A4)	계산 결과 1.2910

표 4.10.2: 흘어짐에 대한 척도 함수 예

- ⑥ MINA 함수는 인수에 입력된 자료중 최솟값(minimum)을 구하여 계산에 포함하지 않은 논리값이 포함되어 있을 때 그 값이 TRUE면 1 FALSE나 텍스트이거나 값이 없는 빈 셀이면 0이 반환된다.

최대값, 최고값 함수 사용방법이다.

```
MAX(number1, number2, ...)
MAXA(value1, value2, ...)
MIN(number1, number2, ...)
MINA(value1, value2, ...)
```

인수 *number1, number2, ...*는 숫자만 사용하며 문자가 사용하면 계산에서 제외한다. *value1, value2, ...*는 배열이나 문자 등 어떤 자료형태도 사용할 수 있다. 표 4.10.3 예를 보자.

	A 열	B 열
1 행	1	1
2 행	2	2
3 행	3	A
=MAX(A1,A2,A3)	계산 결과 3 = 최댓값	
=MAXA(1,2,TRUE)	계산 결과 1 = TRUE	
=MIN(A1,A2,A3)	계산 결과 1 = 1 + 2 + 3	
=MINA(1,2,"문자")	계산 결과 0 = 텍스트	

표 4.10.3: 대푯값 함수 예

#### 4.10.4 두 변수의 관련성을 나타내는 함수

통계학에서 두 변수의 관련성에 대한 지수로 공분산(covariance) 상관계수(correlation coefficient)가 있다. COVARIANCE.P 함수는 두 변수  $X, Y$ 에 대한 모집단의 공분산이다. 모집단의 공분산은

$$cov(X, Y) = \frac{\sum_{i=1}^n (X_i - \mu_X)(Y_i - \mu_Y)}{n} \quad (4.5)$$

로 구하며  $cov(X, Y)$ 는 두 변수의 모집단에 대한 공분산이다. COVARIANCE.S 함수는 두 변수  $X, Y$ 에 대한 표본집단의 공분산이다. 표본집단의 공분산은

$$cov(X, Y) = \frac{\sum_{i=1}^n (X_i - \bar{X})(Y_i - \bar{Y})}{n - 1} \quad (4.6)$$

로 구하며  $cov(X, Y)$ 는 두 변수의 표본집단에 대한 공분산이다. CORREL 함수는 두 변수  $X, Y$ 에 대한 모집단 상관계수(population correlation coefficient)를 구한다. 두 변수  $X, Y$ 의 모집단 상관계수는

$$corr(X, Y) = \frac{cov(X, Y)}{\sigma_X \times \sigma_Y}$$

$$\begin{aligned}
 & \frac{\sum_{i=1}^n (X_i - \mu_X)(Y_i - \mu_Y)}{\sqrt{\frac{\sum_{i=1}^n (X_i - \mu_X)^2}{n} \frac{\sum_{i=1}^n (Y_i - \mu_Y)^2}{n}}} \\
 & = \frac{\sum_{i=1}^n (X_i - \mu_X)(Y_i - \mu_Y)}{\sqrt{\sum_{i=1}^n (X_i - \mu_X)^2 \sum_{i=1}^n (Y_i - \mu_Y)^2}}
 \end{aligned}$$

로 구한다.  $\text{corr}(X, Y)$ 는 두 변수의 상관계수,  $\text{cov}(X, Y)$ 는 두 변수의 공분산,  $\sigma_X$ 는 변수  $X$ 의 표준편차,  $\sigma_Y$ 는 변수  $Y$ 의 표준편차이다. 상관계수는 모집단과 표본집단 모두 계산과정이 동일하며 CORREL 함수로 모두 구할 수 있다. 함수 사용방법은 다음과 같다.

```

COVARIANCE.P(array1, array2)
COVARIANCE.S(array1, array2)
CORREL(array1, array2)

```

이 함수는 인수  $array1, array2$ 에 상관계수를 구할 두 변수의 배열을 입력한다. 표 4.10.4의 예를 보자.

### 4.10.5 개수 함수

COUNT 계열의 함수는 인수 값이나 셀 값이 어떤 특성을 만족하는 것들의 개수 계산에 사용한다. 다음은 개수를 구하는 함수의 종류이다.

- ◉ COUNT 함수는 인수 목록에서 숫자로 되어있는 셀의 개수를 계산한다.
- ◉ COUNTA 함수는 인수 목록에서 비어있지 않은 셀의 개수를 계산한다.

	A 열	B 열
1 행	1	1.2
2 행	2	2.5
3 행	3	3.1
4 행	4	4.9
=COVARIANCE.P(A1:A4,B1:B4)	계산 결과 1.4625	
=COVARIANCE.S(A1:A4,B1:B4)	계산 결과 1.9500	
=CORREL(A1:A4,B1:B4)	계산 결과 0.98271	

표 4.10.4: 공분산과 상관계수 예

- ◉ COUNTBLANK 함수는 인수 목록에서 비어있는 셀의 개수를 계산한다.
- ◉ COUNTIF 함수는 지정한 범위 내에서 조건을 만족하는 셀의 개수를 계산 한다.

개수 함수들의 사용방법은 다음과 같다.

```
COUNT(value1, value2, ...)
COUNTA(value1, value2, ...)
COUNTBLANK(range)
COUNTIF(range, criteria)
```

인수

- ◉ *value1, value2, ...*는 셀을 입력하고
- ◉ *range*는 셀 범위를 입력하며
- ◉ *criteria*에 조건을 입력한다.

표 4.10.5의 예를 보자.

	A 열	B 열
1 행	1	2
2 행	2	4
3 행	3	6
4 행	A	8
=COUNT(A1,A2,A3,A4)		3 숫자 셀
=COUNTA(A1,A2,A3,A4)		4 비어 있지 않은 셀
=COUNTBLANK(A1:A4)		0 빈 셀의 수
=COUNTIF(B1,B2,B3,B4, ">=4")		3 4 보다 큰 셀의 수

표 4.10.5: 개수 함수 예

#### 4.10.6 분포함수

이 절에서는 통계학에서 널리 알려지고 자주 사용하는 확률분포함수(probability distribution function)에 대하여 알아보자. 확률분포함수는 자료의 형태에 따라 이산형(discrete)일 때 확률질량함수(probability mass function)라 하고 연속형(continuous)일 때 확률밀도함수(probability density function)가 있다.

확률분포를 알아보기 전에 먼저 감마함수(gamma function,  $\Gamma(x)$ )를 알아보자. 이 함수는 확률분포함수는 아니지만 확률분포함수 계산에 포함되어 있어 먼저 알아본다. 감마함수는

$$\Gamma(x) = \int_0^{\infty} t^{x-1} e^{-t} dt \quad (4.7)$$

이며  $x$ 가 양수면

$$\Gamma(x) = (x - 1)! \quad (4.8)$$

이 된다. 엑셀에서 제공하는 감마함수는 GAMMA와 GAMMALN이 있으며 사용법은 다음과 같다.

GAMMA( $x$ ), GAMMALN( $x$ )

GAMMA 함수는  $x$  값에 대하여  $\Gamma(x)$ 의 값을 구한다. GAMMALN 함수는  $\Gamma(x)$ 의 자연로그 값  $\ln(\Gamma(x))$ 를 계산한다. 그렇기 때문에 이 함수로 감마함수를 구하려면

$$e^{\ln \Gamma(x)}$$

이미 엑셀에서 EXP(GAMMALN(x))를 입력한다. 예를 들어  $\Gamma(4)$ 를 계산하려면  
 GAMMA(4) ..... 3! = 6  
 EXP(GAMMALN(4)) ..... 3! = 6  
 이다.

먼저 이산형 분포함수의 확률질량함수에 대하여 알아보자.

- ◉ BINOM.DIST 함수는 이항분포(binomial distribution) 확률값  $P[X = x]$  또는 누적 확률값  $P[X \leq x]$ 를 구한다. 이항분포의 확률질량함수는 다음과 같다.

$$f(x) = \begin{cases} \binom{n}{x} p^x (1-p)^{n-x} & \text{if } x = 0, 1, \dots, n \\ 0 & \text{o.w} \end{cases} \quad (4.9)$$

다음은 함수 사용법이다.

BINOM.DIST(*number\_n*, *trials*, *probability\_s*, *cumulative*)

입력값은 다음과 같다.

- ✖ *number\_s* : 성공한 회수( $x$ )
- ✖ *trials* : 독립 시행 회수( $n$ )

- ✖ *probability\_s* : 각 시행에서 성공할 확률( $p$ )
- ✖ *cumulative* : TRUE는 누적확률  $P[X \leq x]$ 이고 FALSE는 확률  $P[X = x]$

함수 사용 예이다.

=BINOM.DIST(2, 4, 0.5, TRUE) ..... 0.6875

위의 예는 시행회수가 4, 성공회수가 2, 성공확률이 0.5일때 누적확률값이 0.6875이다.

=BINOM.DIST(2, 4, 0.5, FALSE) ..... 0.375

위의 예는 시행회수가 4, 성공회수가 2, 성공확률이 0.5일때 확률값이 0.375이다.

- ◉ BINOM.INV 함수는 이항분포의 누적확률  $P[X \leq x_\alpha] = \alpha$ 에서  $x_\alpha$ 를 구한다. 계산 결과 정수가 아니면 소수부분은 무시한다.

**BINOM.INV(*trials*, *probability\_s*, *alpha*)**

입력값은 다음과 같다.

- ✖ *trials* : 독립 시행 회수( $n$ )
- ✖ *probability\_s* : 각 시행에서 성공할 확률( $p$ )
- ✖ *alpha* :  $P[X \leq x_\alpha] = \alpha$ 에서  $\alpha$

함수 사용 예이다.

=BINOM.INV(10, 0.5, 0.5) ..... 5

시행횟수가 10, 성공확률이 0.5, 누적확률이 0.5일 때 성공횟수는 5번이다.

- ◉ NEGBINOM.DIST 함수는 음이항분포(negative binomial distribution)의 확률값  $P[X = x]$  또는 누적 확률값  $P[X \leq x]$ 를 구한다. 음이항분포의

확률질량함수는 다음과 같다.

$$f(x) = \begin{cases} \binom{x-1}{r-1} p^r (1-p)^{x-r} & \text{if } x = r, r+1, \dots \\ 0 & \text{o.w} \end{cases} \quad (4.10)$$

함수 사용법은 다음과 같다.

`NEGBINO.DIST(number_f, number_s, probability_s)`

입력값은 다음과 같다.

- ✖ `number_f` : 실패한 회수( $r$ )
- ✖ `number_s` : 성공 회수의 임계값( $x - r$ )
- ✖ `probability_s` : 각 시행에서 성공할 확률( $p$ )

함수 사용 예이다.

=NEGBIN.DIST(3, 2, 0.5) ..... 0.125

위의 예는 실패한 회수가 3, 성공 회수의 임계값 2, 성공확률이 0.5일때 확률값이 0.125이다.

- ◉ `POISSON.DIST` 함수는 포아송분포(poission distribution)의 확률값  $P[X = x]$  또는 누적 확률값  $P[X \leq x]$ 를 구한다. 포아송분포의 확률질량함수는 다음과 같다.

$$f(x) = \begin{cases} \frac{e^{-m} m^x}{x!} & \text{if } x = 0, 1, \dots \\ 0 & \text{o.w} \end{cases} \quad (4.11)$$

함수 사용법은 다음과 같다.

POISSON.DIST(*x, mean, cumulative*)

입력값은 다음과 같다.

- ✖ *x* : 사건수(*x*)
- ✖ *mean* : 평균(*m*)
- ✖ *cumulative* : TRUE는 누적확률  $P[X \leq x]$ 이고 FALSE는 확률  $P[X = x]$

함수 사용 예이다.

=POISSON.DIST(2, 4, TRUE) ..... 0.2381

위의 예는 사건수가 4, 평균이 2일때 누적확률값이 0.2381이다.

=POISSON.DIST(2, 4, FALSE) ..... 0.1466

위의 예는 사건수가 4, 평균이 2일때 확률값이 0.1466이다.

연속형 분포함수의 확률밀도함수에 대하여 알아보자.

- ◉ BETA.DIST 함수는 베타분포(beta distribution)의 누적확률밀도함수 값  $P[X \leq x]$ 를 구한다. 다음은 베타분포의 확률밀도함수이다.

$$f(x) = \begin{cases} \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} x^{\alpha-1} (1-x)^{\beta-1} & \text{if } 0 < x < 1 \\ 0 & \text{o.w} \end{cases} \quad (4.12)$$

함수 사용법은

BETA.DIST(*x, alpha, beta, A, B*)

이며, 각 입력값은

✖  $x$  : 누적확률을 계산할 값으로  $A$ 와  $B$ 사이의 값

✖  $\alpha$  : 분포에서 모수 값  $\alpha$

✖  $\beta$  : 분포에서 모수 값  $\beta$

✖  $A$  :  $x$ 가 취할 하한값으로 생략하면 0

✖  $B$  :  $x$ 가 취할 상한값으로 생략하면 1

이다. 다음은 함수 사용 예이다.

=BETA.DIST(3,4,10,1,5) ..... 0.9539

- ◉ BETA.INV 함수는 베타분포(beta distribution) 누적확률밀도함수의 역 함수를 계산하며  $p = P[X \leq x_p]$ 의  $x_p$ 이다. 함수 사용법은

**BETA.INV(*probability*,  $\alpha$ ,  $\beta$ ,  $A$ ,  $B$ )**

이며, 각 입력값은

✖  $probability$  : 누적확률로 0과 1사이의 값

✖  $\alpha$  : 분포에서 모수 값  $\alpha(\alpha)$

✖  $\beta$  : 분포에서 모수 값  $\beta(\beta)$

✖  $A$  :  $x$ 가 취할 하한값으로 생략하면 0

✖  $B$  :  $x$ 가 취할 상한값으로 생략하면 1

이다. 다음은 이 함수의 사용 예와 결과이다.

=BETA.INV(0.9539,4,10,1,5) ..... 3.0002

이 함수의 결과값과 BETA.DIST의 예제 결과를 보면 역함수 관계에 있음을 확인할 수 있다.

- CHISQ.DIST 함수는 카이제곱분포(chi distribution;  $\chi^2$ )의 누적확률밀도 함수의 값  $P[X \leq x]$ 를 구한다. 카이제곱분포의 확률밀도함수는 다음과 같다.

$$f(x) = \begin{cases} \frac{1}{2^{\nu/2}\Gamma(\nu/2)}x^{\frac{\nu-2}{2}}e^{-\frac{x}{2}} & \text{if } x > 0 \\ 0 & \text{o.w} \end{cases} \quad (4.13)$$

이 분포의 사용법은

`CHISQ.DIST(x, df)`

이며, 각 입력값은

- $x$  : 누적확률을 구할 값으로 0보다 큰 값
- $df$  : 분포에서 자유도(degree of freedom)( $\nu$ )

이다. 다음은 이 함수의 사용 예와 결과이다.

=CHISQ.DIST(2,5) ..... 0.8492

- CHISQ.INV 함수는 카이제곱분포(chi distribution)의 누적확률밀도함수 역함수 값  $\alpha = P[X \leq x_\alpha]$ 에서  $x_\alpha$ 를 구한다. 사용법은

`CHISQ.INV(probability, df)`

이며, 각 입력값은

- $probability$  : 누적확률로 0과 1사이의 값
- $df$  : 분포에서 자유도(degree of freedom)( $\nu$ )

이다. 다음은 이 함수의 사용 예와 결과이다.

=CHISQ.INV(0.8492,5) ..... 1.9996

이 함수의 결과값과 CHISQ.DIST의 예제 결과를 보면 역함수 관계에 있음을 확인할 수 있다.

- ◉ EXPON.DIST 함수는 지수분포(exponential distribution)의 누적확률밀도함수의 값  $P[X \leq x]$ 를 구한다. 지수분포의 확률밀도함수는 다음과 같다.

$$f(x) = \begin{cases} \lambda e^{-\lambda} & \text{if } x > 0 \\ 0 & \text{o.w} \end{cases} \quad (4.14)$$

이 분포의 사용법은

**EXPON.DIST( $x, \lambda, cumulative$ )**

이며, 각 입력값은

✖  $x$  : 누적확률을 구할 값으로 0보다 큰 값

✖  $\lambda$  : 지수분포에서 모수값( $\lambda$ )

✖  $cumulative$  : TRUE는 누적확률  $P[X \leq x]$ 이고 FALSE는 확률  $P[X = x]$

이다. 다음은 이 함수의 사용 예와 결과이다.

=EXPON.DIST(1,2,TRUE) ..... 0.8647

위의 예는 누적확률에 대한 것이고

=EXPON.DIST(1,2,FALSE) ..... 0.2707

위의 예는 확률에 대한 것이다.

- ◉ F.DIST 함수는 F-분포(F-distribution)의 누적확률밀도함수의 값  $P[X \leq x]$ 를 구한다. F-분포의 확률밀도함수는 다음과 같다.

$$F = \frac{\chi_{\nu_1}^2 / \nu_1}{\chi_{\nu_2}^2 / \nu_2} \quad (4.15)$$

이 분포의 사용법은

F.DIST(*x, df1, df2*)

이며, 각 입력값은

- ✖ *x* : 누적확률을 구할 값으로 0보다 큰 값
- ✖ *df1* : 분포에서 첫 번째 자유도(degree of freedom)( $\nu_1$ )
- ✖ *df2* : 분포에서 두 번째 자유도(degree of freedom)( $\nu_2$ )

이다. 다음은 이 함수의 사용 예와 결과이다.

=F.DIST(2, 4, 3) ..... 0.2978

- ◉ F.INV 함수는 F – 분포(F distribution)의 누적확률밀도함수 역함수 값  $\alpha = P[X \leq x_\alpha]$ 의  $x_\alpha$ 를 구한다. 이 분포의 사용법은

F.INV(*probability, df1, df2*)

이며, 각 입력값은

- ✖ *probability* : 누적확률로 0과 1사이의 값
- ✖ *df1* : 분포에서 첫 번째 자유도(degree of freedom)( $\nu_1$ )
- ✖ *df2* : 분포에서 두 번째 자유도(degree of freedom)( $\nu_2$ )

이다. 다음은 이 함수의 사용 예와 결과이다.

=F.INV(0.2978, 4, 3) ..... 2.0000

이 함수의 결과값과 F.DIST의 예제 결과를 보면 역함수 관계에 있음을 확인할 수 있다.

- GAMMA.DIST 함수는 감마분포(gamma distribution)의 누적확률밀도함수의 값  $P[X \leq x]$ 를 구한다. 감마분포의 확률밀도함수는 다음과 같다.

$$f(x) = \begin{cases} \frac{1}{\beta^\alpha \Gamma(\alpha)} x^{\alpha-1} e^{-\frac{x}{\beta}} & \text{if } x > 0 \\ 0 & \text{o.w} \end{cases} \quad (4.16)$$

이 분포의 사용법은

**GAMMA.DIST(*x, α, β, cumulative*)**

이며, 각 입력값은

- ✖  $x$  : 확률을 구할 값으로 0보다 큰 값
- ✖  $\alpha$  : 분포에서 모양 모수  $\alpha$ (shape parameter)
- ✖  $\beta$  : 분포에서 크기 모수  $\beta$ (scale parameter)
- ✖  $cumulative$  : TRUE는 누적확률  $P[X \leq x]$ 이고 FALSE는 확률  $P[X = x]$

이다. 다음은 이 함수의 사용 예와 결과이다.

=GAMMA.DIST(7,2,5,TRUE) ..... 0.4082

위의 예는 누적확률에 대한 것이고

=GAMMA.DIST(7,2,5,FALSE) ..... 0.0690

위의 예는 확률에 대한 것이다.

- GAMMA.INV 함수는 감마분포(gamma distribution)의 누적확률밀도함수 역함수 값  $\alpha = P[X \leq x_\alpha]$ 의  $x_\alpha$ 를 구한다. 이 분포의 사용법은

**GAMMA.INV(*probability, α, β*)**

이며, 각 입력값은

- ✖ probability : 누적확률로 0과 1사이의 값
- ✖  $\alpha$  : 분포에서 모수  $\alpha(\alpha)$
- ✖  $\beta$  : 분포에서 모수  $\beta(\beta)$

이다. 다음은 이 함수의 사용 예와 결과이다.

=GAMMA.INV(0.4082,2,5) ..... 7.0005

이 함수의 결과값과 GAMMA.DIST의 예제 결과를 보면 역함수 관계에 있음을 확인할 수 있다.

- ◉ LOGNORM.DIST 함수는 로그정규분포(log-normal distribution)의 누적확률밀도함수의 값  $P[X \leq x]$ 를 구한다. 로그정규분포의 확률밀도함수는 다음과 같다.

$$f(x) = \begin{cases} \frac{1}{x\sigma\sqrt{2\pi}} e^{-\frac{(lnx-\mu)^2}{2\sigma^2}} & \text{if } x > 0 \\ 0 & \text{o.w} \end{cases} \quad (4.17)$$

이 분포의 사용법은

`LOGNORM.DIST(x, mean, sd)`

이며, 각 입력값은

- ✖  $x$  : 확률을 구할 값으로 0보다 큰 값
- ✖  $mean$  : 로그정규분포에서 평균( $\mu$ )
- ✖  $sd$  : 로그정규분포에서 표준편차 ( $\sigma$ )

이다. 다음은 이 함수의 사용 예와 결과이다.

=LOGNORM.DIST(10,6,4) ..... 0.1777

위의 예는 누적확률이다.

- LOGNORM.INV 함수는 로그정규분포(log-normal distribution)의 누적확률밀도함수 역함수 값  $\alpha = P[X \leq x_\alpha]$ 의  $x_\alpha$ 를 구한다. 이 분포의 사용법은

`LOGNORM.INV(probability, mean, sd)`

이며, 각 입력값은

- ❖ *probability* : 누적확률로 0과 1사이의 값
- ❖ *mean* : 로그정규분포에서 평균( $\mu$ )
- ❖ *sd* : 로그정규분포에서 표준편차 ( $\sigma$ )

이다. 다음은 이 함수의 사용 예와 결과이다.

=LOGNORM.INV(0.1777,6,4) ..... 10.0075

이 함수의 결과값과 LOGNORM.DIST의 예제 결과를 보면 역함수 관계에 있음을 확인할 수 있다.

- NORM.DIST 함수는 정규분포(normal distribution)의 누적확률밀도함수의 값  $P[X \leq x]$ 를 구한다. 정규분포의 확률밀도함수는 다음과 같다.

$$f(x) = \begin{cases} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}} & \text{if } -\infty < x < \infty \\ 0 & \text{o.w} \end{cases} \quad (4.18)$$

이 분포의 사용법은

`NORM.DIST(x, mean, sd, cumulative)`

이며, 각 입력값은

- ✖  $x$  : 누적확률을 구할 값으로  $-\infty \sim \infty$
- ✖  $mean$  : 정규분포에서 평균( $\mu$ )
- ✖  $sd$  : 정규분포에서 표준편차 ( $\sigma$ )
- ✖  $cumulative$  : TRUE는 누적확률  $P[X \leq x]$ 이고 FALSE는 확률  $P[X = x]$

이다. 다음은 함수 사용 예이다.

=NORM.DIST(1.96, 0, 1, TRUE) ..... 0.9750

위의 예는 누적확률에 대한 것이고

=NORM.DIST(1.96, 0, 1, FALSE) ..... 0.0584

위의 예는 확률에 대한 것이다. 참고로 평균이 0이고 분산이 1인 표준정규분포(standard normal distribution)의 확률밀도함수 값과 누적확률밀도함수 값은 NORM.S.DIST 함수를 사용한다.

- ◉ NORM.INV 함수는 표준정규분포의 누적확률밀도함수 역함수 값  $\alpha = P[X \leq x_\alpha]$ 의  $x_\alpha$ 를 구한다. 이 분포의 사용법은

NORM.INV(*probability*, *mean*, *sd*)

이며, 각 입력값은

- ✖  $probability$  : 누적확률로 0과 1사이의 값
- ✖  $mean$  : 정규분포에서 평균( $\mu$ )
- ✖  $sd$  : 정규분포에서 표준편차 ( $\sigma$ )

이다. 다음은 이 함수의 사용 예와 결과이다.

=NORM.INV(0.975, 0, 1) ..... 1.9599

이 함수의 결과값과 NORM.DIST의 예제 결과를 보면 역함수 관계에 있음을 확인할 수 있다. 참고로 표준정규분포의 누적확률밀도함수 역함수 값  $\alpha = P[X \leq x_\alpha]$ 의  $x_\alpha$ 는 NORM.S.INV로 구한다.

- T.DIST 함수는 t - 분포(t - distribution)의 누적확률밀도함수의 값  $P[X \leq x]$ 를 구한다. t-분포의 확률밀도함수는 다음과 같다.

$$f(x) = \begin{cases} \frac{\Gamma(\frac{\nu+1}{2})}{\sqrt{\pi\nu}\Gamma(\frac{\nu}{2})} \cdot \left(1 + \frac{x^2}{\nu}\right)^{-\frac{\nu+1}{2}} & \text{if } -\infty < x < \infty \\ 0 & \text{o.w} \end{cases} \quad (4.19)$$

이 분포의 사용법은

T.DIST(*x, df, tail*)

이며, 각 입력값은

- ✖ *x* : 누적확률을 구할 값으로  $-\infty \sim \infty$
- ✖ *df* : t-분포에서 자유도( $\nu$ )
- ✖ *tail* : t-분포에서 2는 양측꼬리확률  $P[X \geq x] \times 2$ 과 1은 단측꼬리 확률  $P[X \geq x]$

이다. 다음은 이 함수의 사용 예와 결과이다.

=T.DIST(1.96, 60, 2) ..... 0.0546

위의 예는 양측꼬리확률에 대한 것이고

=T.DIST(1.96, 60, 1) ..... 0.0273

위의 예는 단측꼬리확률에 대한 것이다.

- T.INV 함수는 t - 분포(t - distribution)의 누적확률밀도함수 역함수 값  $\alpha = P[X \leq x_\alpha]$ 의  $x_\alpha$ 를 구한다. 이 분포의 사용법은

T.INV(*probability, df*)

이며, 각 입력값은

- ✖ probability : 누적확률로 0과 1사이의 값
- ✖ df : 분포에서 자유도(degree of freedom)( $\nu$ )

이다. 다음은 이 함수의 사용 예와 결과이다.

=T.INV(0.0546, 60) ..... 1.9604

이 함수의 결과값과 T.DIST의 예제 결과를 보면 역함수 관계에 있음을 확인할 수 있다.

- ◉ WEIBULL.DIST 함수는 와이블분포(Weibull distribution)의 누적확률밀도함수의 값  $P[X \leq x]$ 를 구한다. 와이블분포의 확률밀도함수는 다음과 같다.

$$f(x) = \begin{cases} \frac{k}{\lambda} \left(\frac{x}{\lambda}\right)^{k-1} e^{-(x/\lambda)^k} & \text{if } x \geq 0 \\ 0 & \text{o.w.} \end{cases} \quad (4.20)$$

이 분포의 사용법은

**WEIBULL.DIST( $x, \alpha, \beta, cumulative$ )**

이며, 각 입력값은

- ✖  $x$  : 누적확률을 구할 값으로 0보다 큰 값
- ✖  $\alpha$  : 분포에서 모수  $\alpha$
- ✖  $\beta$  : 분포에서 모수  $\beta$
- ✖ cumulative : TRUE는 누적확률  $P[X \leq x]$ 이고 FALSE는 확률  $P[X = x]$

이다. 다음은 이 함수의 사용 예와 결과이다.

=WEIBULL.DIST(3, 2, 4, TRUE) ..... 0.4302

위의 예는 누적확률에 대한 것이고

=WEIBULL.DIST(3, 2, 4, FALSE) ..... 0.2137

위의 예는 확률에 대한 것이다.

#### 4.10.7 통계적 추론 함수

통계학에서 추론(inference)은 추정(estimation)과 검정(testing)이다. 추정은 모집단의 특성 값을 한 값 예측하면 점추정(point estimation), 구간을 예측하면 구간추정(interval estimation)이다. 검정은 연구자가 세운 가설이 옳고 그른지 판단한다. 우선 통계적 가설 검정에서 많이 사용하는 것을 알아보자.

##### ◉ CHISQ.TEST 함수는

- ✖ 적합도 검정(goodness of fit test)
- ✖ 동일성 검정(homogeneity test)
- ✖ 독립성 검정(independence test)

등에 사용한다. 적합도 검정인 경우 귀무가설이

$$H_0 = \text{goodness of fit of probability model}$$

이고, 동일성 검정인 경우 귀무가설이

$$H_0 : p = p_1 = p_2 = \dots$$

이며 독립성 검정인 경우

$$H_0 : p_{i,j} = p_i \times p_j \quad \text{for all } i, j.$$

이다. 이 검정방법의 사용방법은

`CHISQ.TEST(actual_range, expected_range)`

이며, 각 입력값은

- ✖ *actual\_range* : 범주형 자료에서 관측값의 범위

관측값(기대값)	너무 많다	적당하다	너무 적다	합계
남자	378(392.8)	237(222)	26(26.2)	641
여자	388(373.2)	196(211)	25(24.8)	609
합계	766	433	51	1250

표 4.10.6: 관측값과 기대값

☞ *expected\_range* : 범주형 자료에서 평균값의 범위. 이 값은 자료에서  $\frac{n_i \cdot n_j}{n_{..}}$ 로 구한다. 표 4.10.6에서 각 값이  $n_i = 766, 433, 51$ 이고  $n_j = 641, 609$ 이며  $n_{..} = 1250$ 일 때 남자이고 너무 많다의 평균은  $\frac{766 \times 641}{1250} = 392.8$ 이다. 나머지 값도 동일한 방법으로 구하면 표 4.10.6의 괄호 한의 값이 계산된다.

이미 결과는 유의확률(p-value)이다. 검정하는 자료가 표 4.10.6와 같다면

```
=CHISQ.TEST({378, 237, 26; 388, 196, 25}, {392.8, 222.04, 26.15; 373.2, 210.96, 24.85}) ..... 0.2004
```

위의 예는 모든 값을 직접 입력하였고 행 구분은 세미콜론(;)이다. 만일 관측값 범위가 B1:D2이고 기대값 범위가 B5:D6이면

```
=CHISQ.TEST(B1:D2, B5:D6) ..... 0.2004  
로 입력하여 결과를 확인한다. 결과는 유의확률이다.
```

- ◉ CONFIDENCE.NORM 함수는 모 평균  $\mu$ 에 대한  $100 \times (1 - \alpha)\%$  신뢰구간에서 정규분포의 오차한계(margin of error)이다. 또한 CONFIDENCE.T 함수는 모 평균  $\mu$ 에 대한  $100 \times (1 - \alpha\%)$  신뢰구간에서 t - 분포의 오차한계(margin of error)를 계산한다. 모 평균  $\mu$ 에 대한  $100 \times (1 - \alpha)\%$  신뢰구간은 정규분포를 따르는 경우와 t - 분포를 따르는 경우로

$$\text{정규분포} : \bar{X} \pm z_{\alpha/2} \frac{\sigma}{\sqrt{n}} \quad (4.21)$$

$$\text{t - 분포} : \bar{X} \pm t_{\alpha/2}(n - 1) \frac{\sigma}{\sqrt{n}} \quad (4.22)$$

이며 식에서 각 값은

- ✖  $\bar{X}$  : 표본 평균
  - ✖  $z_{\alpha/2}$  : 정규분포에서  $P[Z \geq z_{\alpha/2}] = \alpha/2$ 의 값
  - ✖  $t_{\alpha/2}(n - 1)$  : t - 분포에서  $P[T \geq t_{\alpha/2}] = \alpha/2$ 의 값
  - ✖  $\sigma$  : 표본의 표준편차
  - ✖  $n$  : 표본수

이다. 여기서 오차한계는

$$\text{정규분포인 경우 : } Z_{\alpha/2} \frac{\sigma}{\sqrt{n}} \quad (4.23)$$

$$t\text{-분포인 경우} : t_{\alpha/2}(n-1) \frac{\sigma}{\sqrt{n}} \quad (4.24)$$

이다. 이 오차한계의 사용방법은

**CONFIDENCE.NORM( $\alpha$ ,  $sd$ ,  $size$ )**  
**CONFIDENCE.T( $\alpha$ ,  $sd$ ,  $size$ )**

이며, 각 입력값은

- ✖  $\alpha$  : 신뢰구간의 신뢰수준으로  $100 \times (1 - \alpha)\%$ 에서  $\alpha$
  - ✖  $sd$  : 표본 표준편차
  - ✖  $size$  : 표본 크기

이다. 만일 자료가 1, 2, 3, 4, 5 일 때 모 평균  $\mu$ 에 대한 95% 신뢰구간을 구해보자. 여기서

표준편차는 STDDEV.S(1, 2, 3, 4, 5) ..... 1.5811  
 $\alpha$ 는 ..... 0.05  
 $n$ 은 표본수로 ..... 5

이다. 이 값으로 정규분포를 따르는 자료에 대한 오차한계는  
=CONFIDENCE.NORM(0.05, 1.5811, 5) ..... 1.3859  
이다. 따라서 신뢰구간은  $3 \pm 1.3859$ 으로  $(1.6141, 4.3859)$ 이다. 또한 t -  
분포를 따르는 자료에 대한 오차한계는  
=CONFIDENCE.T(0.05, 1.5811, 5) ..... 1.9632  
이다. 따라서 신뢰구간은  $3 \pm 1.9632$ 으로  $(1.0368, 4.9632)$ 이다.

- F.TEST 함수는 두 집단의 분산이 같은지 대한 등분산성 검정으로 귀무 가설은

$$H_0 : \frac{\sigma_1^2}{\sigma_2^2} = 1$$

## 함수 사용방법은

F.TEST(array1, array2)

이며, 각 입력값은

- ✖ *array1* : 첫 번째 변수에 입력된 데이터로 배열이나 범위를 입력하며 공백은 무시
  - ✖ *array2* : 두 번째 변수에 입력된 데이터로 배열이나 범위를 입력하며 공백은 무시

이다. 만일 첫 번째 변수가 A1에서 A5까지 1, 2, 3, 4, 5이고, 두 번째 변수가 B1에서 B5까지 2, 4, 6, 8, 10일 때 두 변수의 등분산(equal variance)에 대한 검정에서 유의확률은

=F.TEST(A1:A5, B1:B5) ..... 0.208  
로 구한다.

- T.TEST 함수는 두 집단의 평균이 같은지 다른지에 대한 검정으로 함수 사용방법은

```
T.TEST(array1, array2, tails, type)
```

이며, 각 입력값은

- ✖ *array1* : 첫 번째 변수에 입력된 데이터로 배열이나 범위를 입력하며 공백은 무시
- ✖ *array2* : 두 번째 변수에 입력된 데이터로 배열이나 범위를 입력하며 공백은 무시
- ✖ *tails* : 1은 단측검정이고 2는 양측검정
- ✖ *type* : 1은 두 변수값이 쌍으로 된 경우, 2는 두 집단에 대한 검정으로 분산이 같은 경우, 3은 두 집단에 대한 검정으로 분산이 다른 경우

이다. 만일 첫 번째 변수가 A1에서 A5까지 1, 2, 3, 4, 5이고, 두 번째 변수가 B1에서 B5까지 2, 4, 6, 8, 10일 때 두 변수의 분산은 같고 서로 다른 두 집단인 경우 두 변수의 평균이 같은지에 대한 검정에서 유의확률은

=T.TEST(A1:A5, B1:B5, 2, 2) ..... 0.0943  
로 구한다.

- ◉ Z.TEST 함수는 귀무가설  $H_0 : \mu = \mu_0$ 에 대한 검정에서 단측 검정의 유의확률을 계산해 준다. 이 검정방법의 사용방법은

```
Z.TEST(array, x, σ)
```

이며, 각 입력값은

- ✖ *array* : 변수에 속한 데이터의 배열이나 범위를 입력하며 공백은 무시
- ✖ *x* : 귀무가설  $H_0 : \mu = x$
- ✖ *σ* : 표본의 표준편차  $\sigma (\sigma)$

이다. 만일 첫 변수가 A1에서 A5까지 1, 2, 3, 4, 5일 때 모 평균  $\mu$ 에 대한 검정에서 귀무가설  $H_0 : \mu = 2$ 이고 모집단의 표준편차가  $\sigma = 1.5811$ 일 때 유의확률은

=Z.TEST(A1:A5, 2, 1.5811) ..... 0.0786  
로 구한다.

## 제 4.11 절 찾기/참조 함수

찾기/참조함수는 벡터나 배열에서 값을 찾는다. 이 함수들은 데이터 셋에서 특정한 값을 찾아주기에 자료 검색에 편리하다.

### 4.11.1 순위 함수

RANK.AVG 함수와 RANK.EQ 함수는 어떤 데이터가 그 데이터가 속 집단에서 순위를 구해준다. 두 함수의 차이점은 동점 순위가 있는 경우 RANK.AVG 함수는 동점 순위의 평균을 순위로 제공한다. 3위와 4위가 동점이라면 둘 다 3.5위가 된다. RANK.EQ 함수는 동점순위의 먼저 순위를 동점의 순위로 한다. 예를 들어 3위 4위가 동점이라면 두 자료의 순위는 모두 3위이다. 함수 사용법은

```
RANK.AVG(number, ref, order)
RANK.EQ(number, ref, order)
```

이며, 각 입력값은

- ◉ *number* : 순위를 구하려는 수

● *ref* : 순위를 구할 데이터들의 목록이나 배열

● *order* : 순위를 결정하는 것으로

    ✖ 0이나 생략하면 내림차순

    ✖ 1은 오름차순

이다. 여기서

    ✖ 오름차순은 숫자인 경우 데이터가 증가할 때 순위도 증가하는 형태로 예를 들어 100m 육상에서 순위를 나타낼 때 기록이 제일 빠른 선수가 1등, 그 다음 기록이 빠른 선수가 2등으로 나타내는 것을 말한다.

    ✖ 내림차순은 숫자인 경우 데이터가 증가할 때 순위가 감소하는 형태로 예를 들어 성적인 경우 점수가 제일 높은 사람이 1등, 그 다음으로 성적이 높은 사람이 2등으로 나타내는 것을 말한다.

    ✖ 숫자인 경우 오름차순은 1, 2, 3 문자인 경우 오름차순은 a, b, c 형태로 정렬된다.

만일 자료가 셀 A1부터 아래로 2, 3, 3, 1, 7있을 때

=RANK.AVG(A2, A1:A5, 1)는 ..... 3.5이다.  
즉 3의 값은 목록내에서 오름차순으로 순위가 3,4이다. 두 순위의 평균인 3.5 가 이 자료의 순위이다.

또한 =RANK.EQ(A2, A1:A5, 1)는 ..... 3이다.  
즉 3의 값은 목록내에서 오름차순으로 순위가 3,4이다. 두 순위에서 빠른 순위 3이 데이터 목록 내에서 순위이다.

#### 4.11.2 찾기 함수

찾기 함수는 찾으려는 값이 데이터 셋의 첫 째 열이나 행에 있을 때 그 값을 가져온다. 행이나 열에 있는 데이터가 벡터이면 가져올 데이터 번호를 입력한다.

- ◉ VLOOKUP 함수 : 행에서 찾는 함수

- ◉ HLOOKUP 함수 : 열에서 찾는 함수

를 사용한다. 함수 사용법은

```
VLOOKUP(lookup_value, table_array, col_index_num, range_lookup)
HLOOKUP(lookup_value, table_array, row_index_num, range_lookup)
```

이며, 각 입력값은

- ◉ *lookup\_value* : 첫 번째 열이나 행에서 찾는 값으로 첫 행이나 열은 반드시 오름차순으로 정렬되어 있어야 올바른 값을 구할 수 있다.
- ◉ *table\_array* : 2개 이상의 데이터 열이나 행
- ◉ *col\_index\_num* : 첫 행에서 데이터가 일치할 때 가져올 열의 번호
- ◉ *row\_index\_num* : 첫 열에서 데이터가 일치할 때 가져올 행의 번호
- ◉ *range\_lookup* : 정확한 값이나 근사값 중에서 찾는 값 결정으로
  - ✖ TRUE나 생략하면 근사값을 찾는다. 근사값은 행이나 열에서 작은 값을 중 최댓값을 찾는다. 예를 들어 찾는 값이 6이고 첫 행이 1, 3, 5, 7, 9이라면 6보다 작은 값들은 1, 3, 5이고 이 중 최댓값은 5 가 된다.<sup>2</sup>
  - ✖ FALSE는 정확하게 일치하는 값을 찾는다.

만일 데이터가 표 4.11.1일 때 다음과 같이 함수를 사용하면

=VLOOKUP(85, A1:B5, 3, TRUE) ..... 우  
이다. 첫 번째 인수 85는 찾을 값, 두 번째 인수 A1:B5는 찾을 범위로

---

<sup>2</sup>첫 행이나 열은 반드시 오름차순으로 정렬되어야 한다. 그렇지 않으면 근사값을 찾을 때 올바른 결과가 계산되지 않는다.

실제 찾는 범위는 A1:A5에서 85를 찾는다. 세 번째 인수 3은 값을 찾았을 때 가져올 열 번호이다. 네 번째 인수 TRUE는 찾을 값이 근사값이다. 85를 0, 60, 70, 80, 90에서 찾는데 작은 값 중에서 최대값을 찾는다. 85 보다 작은 값은 0, 60, 70, 80이고 이 중에서 최대값은 80이므로 이 값을 85의 근사값으로 찾는다.

	A 열	B 열	C 열
1 행	0	가	매우 노력 바람
2 행	60	양	노력 바람
3 행	70	미	보통
4 행	80	우	잘함
5 행	90	수	매우 잘함

표 4.11.1: VLOOKUP 함수 사용 예

### 4.11.3 MATCH, INDEX, CHOOSE 함수

MATCH 함수는 찾는 값이 찾을 셀 범위에서 위치를 알려준다. MATCH 함수는 INDEX, CHOOSE 함수와 함께 사용하면 유용하다. 함수 사용법은

`MATCH(lookup_value, lookup_array, match_type)`

이며, 각 입력값은

- ◉ *lookup\_value* : 찾을 값으로 숫자, 문자열, 논리값이다.
- ◉ *lookup\_array* : 찾을 값의 셀 범위이다.
- ◉ *match\_type* : 찾는 값 검색 옵션이다. -1, 0, 1 중 하나를 설정한다.

- ✖ -1은 찾는 값이 셀 범위 값에서 크거나 같은 값 중 최솟값이다. 셀 범위는 내림차순으로 정렬되어 올바른 결과를 얻는다.
- ✖ 0 같은 첫 번째 값을 찾는다.
- ✖ 1은 찾는 값이 셀 범위 값에서 작거나 같은 값 중 최대값이다. 셀 범위는 오름차순으로 정렬되어 올바른 결과를 얻는다.

만일 셀 범위가 방풍자켓, 등산배낭, 텐트, 등산화, 바지에서 등산화가 몇 번째인지 다음과 같이 실행하면

=MATCH("등산화", C1:C5, 0) .....4

를 얻는다. INDEX 함수는 1차원 또는 2차원 셀 범위에서 행 번호와 열 번호가 교차되는 값을 가져온다. 1 차원이면 해당없는 인수는 생략한다. 함수 사용법은

**INDEX(array, row\_num, col\_num)**

이며, 각 입력값은

- ◉ *array* : 셀 범위
- ◉ *row\_num* : 행 번호
- ◉ *col\_num* : 열 번호

이다. 이 함수는 같은 이름이 두 개이므로 인수 선택 창에서 *array*, *row\_num*, *col\_num*를 선택한다. 표 4.11.1의 셀 범위에서 다음과 같이 실행하면

=INDEX(A1:C5, 4,2) ..... 우

얻는다. CHOOSE 함수는 인수 목록에서 하나를 고른다. 이 함수 사용법은

**CHOOSE(index, value<sub>1</sub>, value<sub>2</sub>, ...)**

이며, 각 입력값은

- ◎ *index* : 인수 목록에서 찾을 번호로 *value*에서 첨자
- ◎ *value<sub>1</sub>*, *value<sub>2</sub>*, ... : 목록으로 1에서 254개까지 사용이 가능

하다. 다음과 같이 실행하면

=CHOOSE(3,"빨","주","노","초","파","남","보") .....노  
를 얻는다.

## 제 4.12 절 재무 함수

재무 함수는 재화에 대한 대출과 투자에 영향을 주는 함수이다. 이 절에서 소개하는 재무 함수는 미래가치를 계산하는 FV 함수와 미래 가치에 대한 현재 납입 금액을 계산하는 PMT 함수이다.

### 4.12.1 FV 함수

FV 함수는 현재 금액에 대한 미래 가치를 계산한다. 이 함수 사용법은

$$\text{FV}(\text{Rate}, \text{Nper}, \text{Pmt}, \text{Pv}, \text{Type})$$

이며, 각 입력값은

- ◎ *Rate* : 이자율
- ◎ *Nper* : 기간
- ◎ *Pmt* : 고정 지급액
- ◎ *Pv* : 현재 가치
- ◎ *Type* : 지급 시기

✖ 1 : 투자 주기 초

✖ 0 : 투자 주기 말

만일 연이율이 2.76%, 납입기간이 2년, 월 납입금액이 860,000원이면 2년 후 받는 금액은  $=FV(2.76\%/12, 2*12, -860000, 0, 0)$ 로 입력하면 .. 21,195,248 이다.

### 4.12.2 PMT 함수

PMT 함수는 전체 대출 상환금에 대하여 일정 기간에 은행에 빚을 갚는다. 이 함수 사용법은

 $\text{PMT}(Rate, Nper, Pmt, Pv, Fv, Type)$ 

이며, 각 입력값은

◉ Rate : 이자율

◉ Nper : 기간

◉ Pmt : 고정 지급액

◉ Pv : 현재 가치

◉ Fv : 미래 가치

◉ Type : 지급 시기

✖ 1 : 투자 주기 초

✖ 0 : 투자 주기 말

만일 연이율이 2.76%, 납입기간이 2년, 대출 총 금액이 20,000,000원이면 2년 동안 매 달 은행에 갚는 금액은 =PMT(2.76%/12, 2\*12, 20000000, 0, 0)로 입력하면 ..... -857,503이다.

## 제 4.13 절 사용자가 만든 함수

사용자가 함수를 제작하여 엑셀에서 사용하려면 비주얼 베이직(VBA)으로 프로그밍을 해야 한다. 비주얼 베이직은 오피스를 설치하면 그 안에 포함되기 때문에 엑셀을 사용할 수 있다면 즉시 사용할 수 있다. 비주얼 베이직에 대한 자세한 사용법은 이후에 자세히 다룰 것이다. 사용자가 만든 함수를 제작하는 과정을 알아보자.

1. 비주얼 베이직 에디터에 비주얼 베이직 프로그램으로 함수를 작성한다.  
그림 4.13.1의 사용자가 만든 함수는 숫자를 입력하였을 때 소수인지

```

Microsoft Visual Basic for Applications - prime.xlam - [Module1 (코드)]
파일(F) 편집(E) 보기(U) 삽입(I) 형식(Q) 디버그(D) 실행(R) 도구(I) 추가 기능(A) 창(W) 도움말(H)
프로젝트: VBAProject (prime.xlam)
Module1 모듈
(일반) Prime
Function Prime(num) As String
    Dim n As Integer
    If Not IsNumeric(num) Then
        Prime = "문자는 입력되지 않습니다."
        Exit Function
    End If
    n = num
    If n <= num Then
        Prime = "올바른 숫자를 입력하세요."
    ElseIf n < 1 Then
        Prime = "양의 정수만 입력이 가능합니다."
    ElseIf n = 1 Then
        Prime = "1은 소수도 합성수도 아닙니다."
    ElseIf n > 1 Then
        For i = 2 To n - 1 Step 1
            If (n Mod i = 0) Then
                Prime = n & "(는)" & i & "로 나누어 소수가 아닙니다."
                Exit Function
            End If
        Next i
        Prime = n & "(는) 소수입니다."
    End If
End Function

```

그림 4.13.1: 사용자가 만든 함수 1

합성수인지 판별하는 것으로 비주얼 베이직으로 작성되었으며 내용은 다음과 같다.

```
1 Function Prime(num) As String
2     Dim n As Integer
3     If Not IsNumeric(num) Then
4         Prime = "문자는 입력되지 않습니다."
5         Exit Function
6     End If
7     n = num
8     If n <> num Then
9         Prime = "올바른 숫자를 입력하세요."
10    ElseIf n < 1 Then
11        Prime = "양의 정수만 입력이 가능합니다."
12    ElseIf (n = 1) Then
13        Prime = "1은 소수도 합성수도 아닙니다."
14    ElseIf (n > 1) Then
15        For i = 2 To n - 1 Step 1
16            If (n Mod i = 0) Then
17                Prime = n & "은(는) " & i & "로 나누어 소수가 아닙니다."
18                Exit Function
19            End If
20        Next i
21        Prime = n & "은(는) 소수입니다."
22    End If
23 End Function
```

사용자 함수 만들기는

① 개발도구→Visual Basic 메뉴

를 클릭하면 Microsoft Visual Basic 프로그램 창이 그림 4.13.1과 같이 나타나고 여기에 코드를 작성한다. 개발도구 메뉴는 기본값 메뉴에 보이지 않으므로 직접 추가해야 한다. 이 메뉴 추가는 파일 → 옵션 → 리본 사용자 지정 → 추가 기능 메뉴를 체크한다. 추가한 개발 도구 메뉴는 엑셀 메뉴에서 즉시 확인할 수 있다.

- 처음에는 프로그램 작성이 쉽지 않으므로 이미 작성된 프로그램을 메모

장에서 열어 복사한 후 VBA 편집기에 붙여넣기한다.

3. 완성된 VBA 프로그램은 저장할 때 파일 형식을 Excel 추가 기능으로 선택하면 저장 파일의 위치가 Addins 폴더가 되며 여기에 저장한다. 이 파일의 확장자는 .xlsa이며 저장된 파일 명은 Userf.xlsa라고 하자.

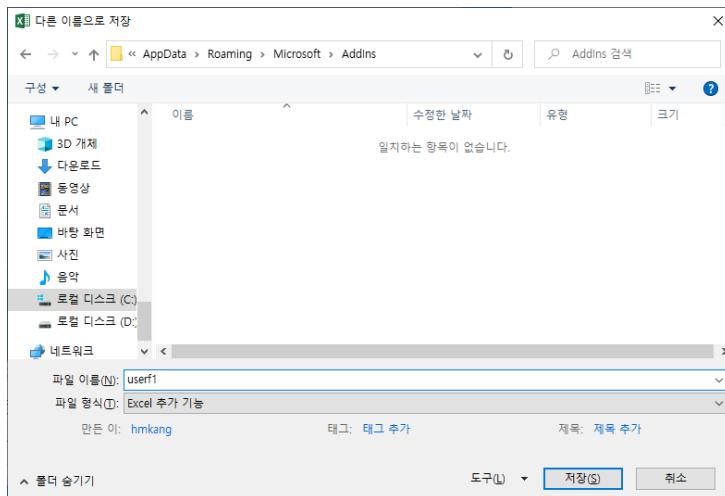
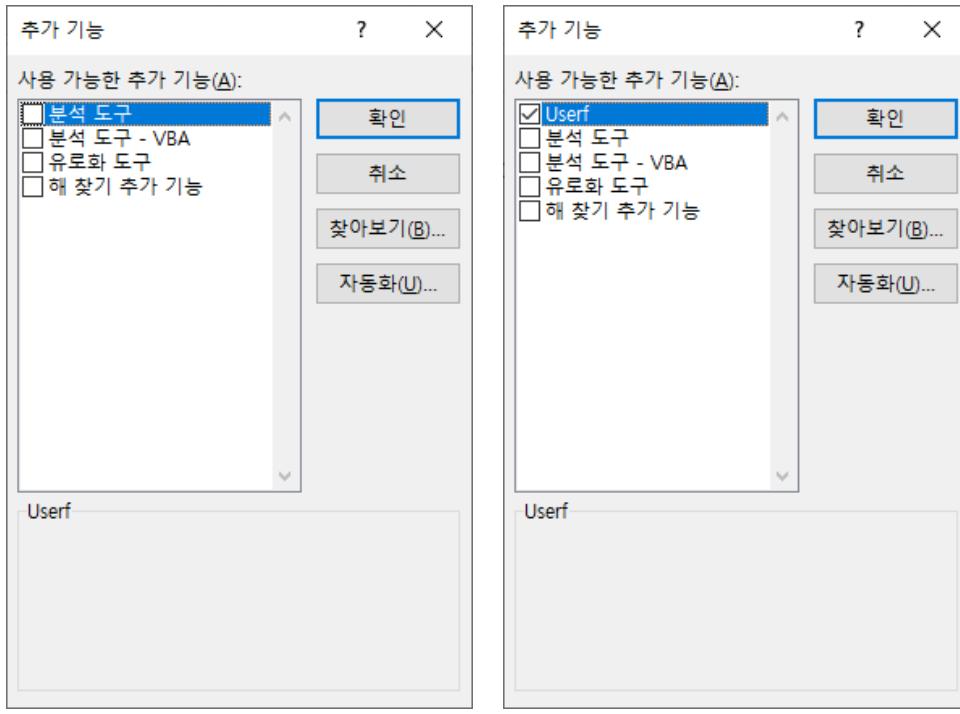


그림 4.13.2: 사용자가 만든 함수 2

4. 사용자가 만든 함수는 파일 → 옵션 → 추가 기능 → Excel 추가 기능에서 이동 버튼 클릭하여 추가기능 창을 연다. 추가하려는 함수를 선택하면 그 내용이 엑셀에 추가된다. 그림 4.3(a)에서 찾아보기 버튼을 누르고 찾아보기 창에서 Userf.xlsa 파일을 추가하면 Userf 함수가 추가된 것을 4.3(b)에서 확인할 수 있다.
5. 추가한 사용자가 만든 함수는 엑셀에서 사용할 수 있다. 사용방법은 기존에 있는 함수 사용방법과 동일하다.
  - ◉ 셀에 함수를 직접 입력하거나
  - ◉ 그림 4.13.4에 추가된 Prime 함수가 보인다. 이 함수는 함수마법사 → 범주 선택 사용자 정의 → 함수 선택 Prime 함수를 호출한다.



(a) 추가 기능에 사용자 함수 추가 전

(b) 추가 기능에 사용자 함수 추가 후

그림 4.13.3: 사용자가 만든 함수 3

6. 그림 4.5(a)는 솟수 판별할 숫자를 입력하고 실행하면 결과가 그림 4.5(b)에 보인다.

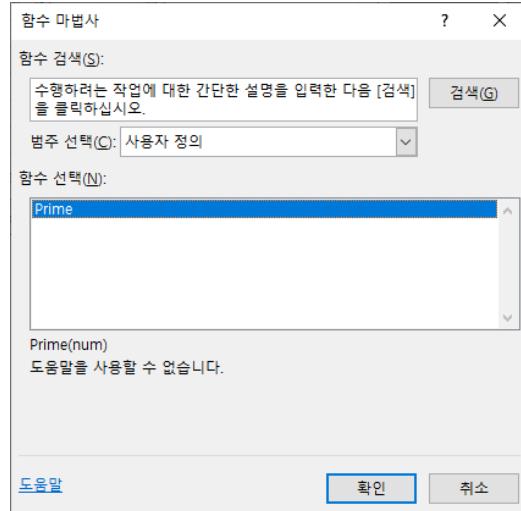


그림 4.13.4: 사용자가 만든 함수 4

(a) 인수에 값 입력하기

(b) 사용자가 만든 함수 실행 결과

## 제 5 장

### 차트 그리기

차트(그래프)는 데이터를 눈으로 쉽게 이해시킬 수 있기에 자료 요약 방법으로 널리 사용하고 있다. 엑셀에서 제공하는 그래프는 한 변수와 여러 변수에 대하여 그레프를 만든다. 차트의 제작은 삽입 리본메뉴→차트에서 만들수 있으며 그 종류는

- ◉ 세로 막대형
- ◉ 꺽은 선형
- ◉ 원형
- ◉ 가로 막대형
- ◉ 영역형
- ◉ 분산형
- ◉ 주식형
- ◉ 표면형
- ◉ 방사형

- 트리맵
- 선버스트
- 히스토그램
- 상자 수염 그림
- 폭포
- 콤보

등이 있으며 그래프의 디스플레이 형태에 따라

- 1차원 그래프
- 2차원 그래프
- 3차원 그래프

등으로 표현할 수 있다.

## 제 5.1 절 차트

엑셀에서 차트는 만들기가 매우 간단하다.

- dataset은 데이터를 모아놓은 것으로 현재 셀을 데이터 셋(dataset)에 놓는다.
1. 차트에 사용할 데이터 선택은 현재 셀을 데이터 셋(dataset)에 놓는다.
  2. 삽입 → 차트에서 차트 종류를 선택한다.
  3. 세부적인 차트 수정은 차트를 클릭하여 진행한다.

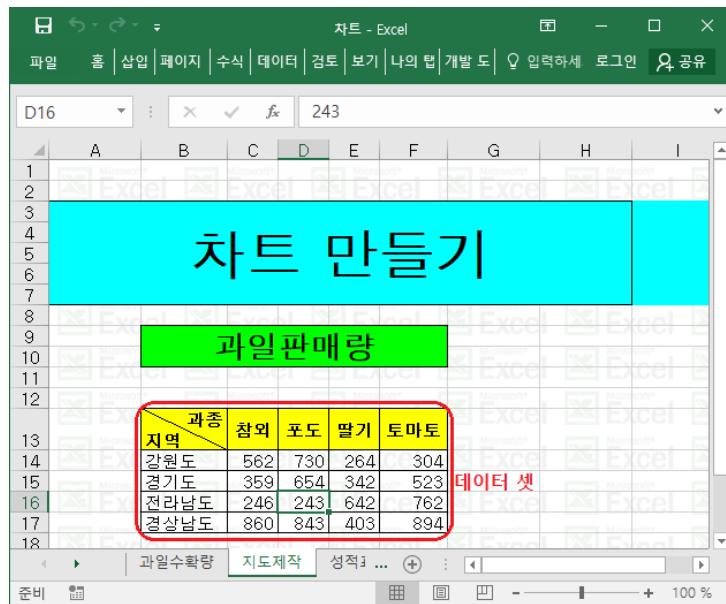


그림 5.2.1: 차트 시작전 현재셀 위치

## 제 5.2 절 원 그래프

원 그래프는 데이터가 속하는 범주의 개수를 총 개수로 나누어 호(arc)로 나타낸 것이다. 원 그래프를 그리는 과정에 대하여 알아보자.

1. 현재셀을 데이터 셋에 옮긴다(그림 5.2.1).
2. 삽입 → 차트 → 원형 또는 도넛형 차트 삽입 메뉴를 선택하거나
3. 삽입 → 차트 → 추천 차트 → 모든 차트 → 원형 메뉴를 선택(그림 5.2.2)
4. 기본으로 그린 원 그래프이다.(그림 5.2.2). 몇 가지 설정하면 원 그래프에 많은 정보의 표현이 가능하다.
  - ◉ 차트 요소 : 차트에 차트 제목, 데이터 레이블, 범례를 선택하거나 제거할 수 있다(그림 5.3(a)).

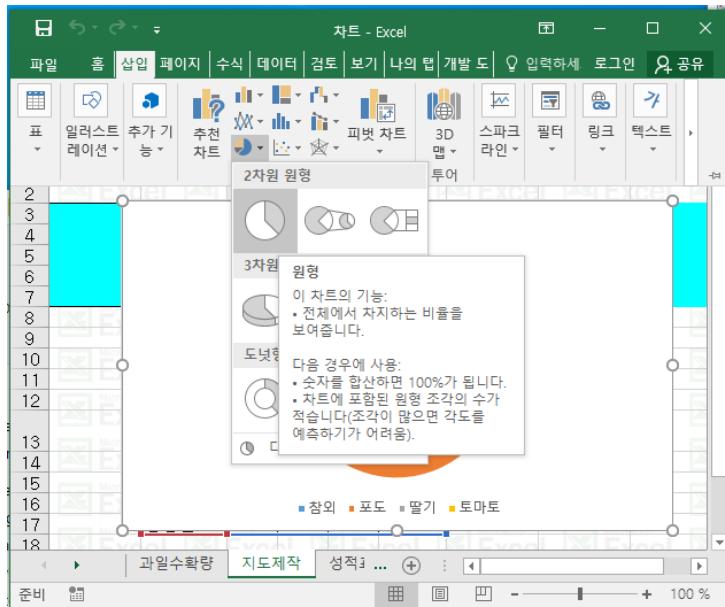


그림 5.2.2: 원 그래프 선택

- ◉ 차트 스타일 : 차트 스타일을 선택한다(그림 5.3(b)).
- ◉ 차트 필터 : 차트에 표현할 값과 이름을 선택한다(그림 5.3(c)).

#### 5. 차트 이동 : 차트 위치를 설정하는 것이다(그림 5.2.4).

- ◉ 새로운 차트로 : 차트 이름을 입력하여 차트가 새로운 시트에 그려 지도록 설정한다.
- ◉ 워크시트에 삽입 : 자료가 포함된 워크시트나 기존에 존재하는 워크시트에 차트를 삽입한다(그림 5.2.4).

## 제 5.3 절 막대 그래프

막대 그래프는 그릴 범주에 해당하는 값의 도수(frequency)나 해당 도수를 총 도수로 나눈 상대도수(relative frequency)를 막대로 그린다.

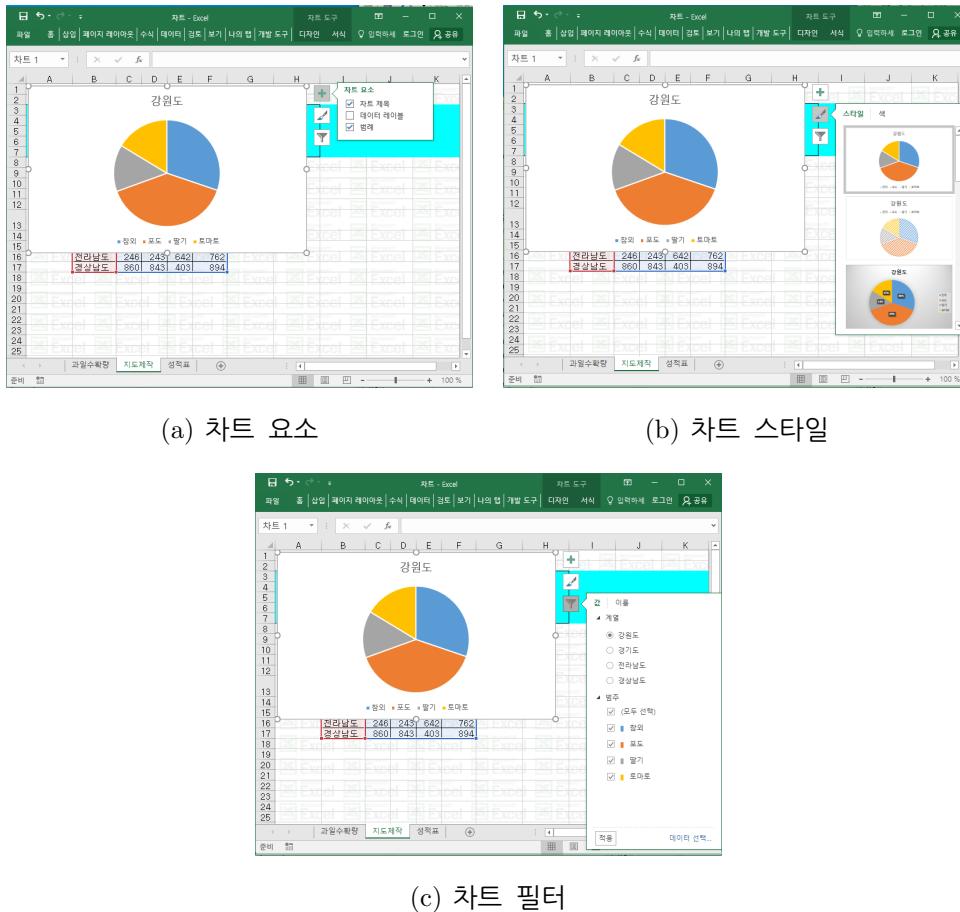


그림 5.2.3: 차트 서식

1. 현재셀을 데이터 셋에 옮긴다(그림 5.2.1).
2. 삽입 → 차트 → 세로 또는 가로 막대형 차트 삽입 메뉴를 선택하거나
3. 삽입 → 차트 → 추천 차트 → 모든 차트 → 세로 막대형 → 묶은 세로 막대형 메뉴를 선택한다(그림 5.3.1).
4. 기본으로 그린 막대 그래프이다.(그림 5.3.1). 몇 가지 설정하면 막대 그

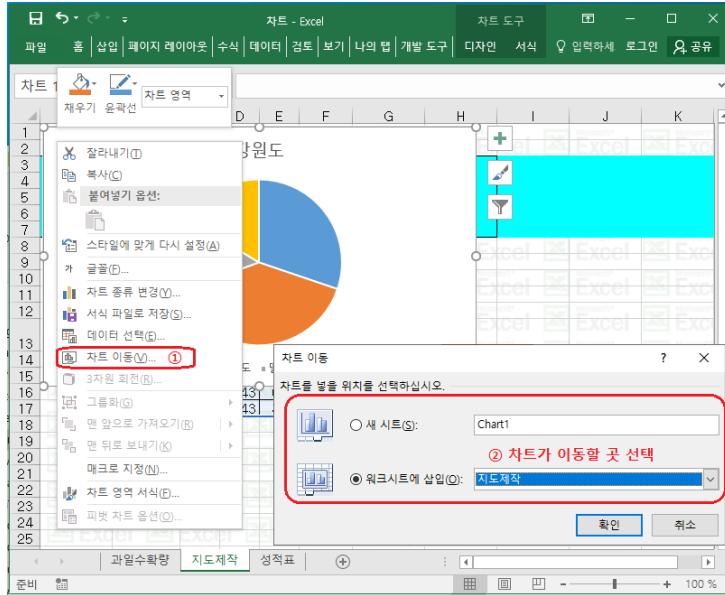


그림 5.2.4: 차트 이동

래프에 많은 정보의 표현이 가능하다.

- ◉ 차트 요소 : 차트에 차트 제목, 데이터 레이블, 범례를 선택하거나 제거할 수 있다(그림 5.2(a)).
- ◉ 차트 스타일 : 차트 스타일을 선택한다(그림 5.2(b)).
- ◉ 차트 필터 : 차트에 표현할 값과 이름을 선택한다(그림 5.2(c)).

#### 5. 데이터 선택 : 차트에서 사용할 데이터를 선택한다(그림 5.3.3).

- ◉ 차트 데이터 범위 : 차트 데이터 범위를 직접 변경한다.
- ◉ 범례 항목(계열) : 범례를 추가, 편집, 제거한다.
- ◉ 가로(항목) 축 레이블 : 가로 축 레이블을 설정한다. 기본값은 셀에서 가져오고 변경은 편집 버튼을 누르고 한다(그림 5.3.3).

#### 6. 막대에 그림을 추가한다.

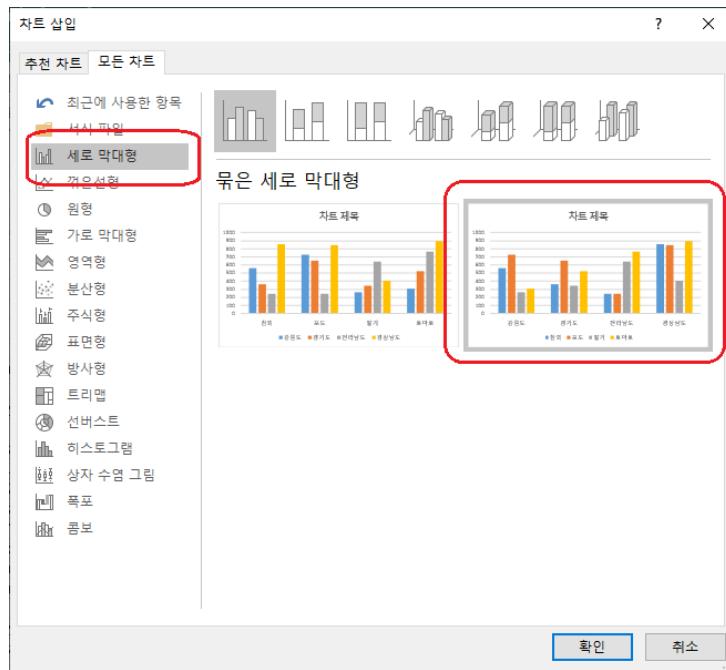


그림 5.3.1: 막대 그래프 종류 선택

- 막대에 파일 그림 추가는 막대에서 마우스 오른쪽 버튼을 누르고 데이터 계열 서식 메뉴를 선택한다(5.4(a)).
- 오른쪽에 생긴 데이터 계열 서식에서 계열 옵션 → 채우기 → 그림 또는 질감 채우기를 선택한다. 그리고 다음에서 그림 삽입 → 파일 버튼을 눌러 파일그림을 추가한다(그림 5.4(b)).
- 데이터 계열 서식에서 아래로 내려오면 다음 배열에 맞게 쓰기의 Units/Picture에 적당한 값 200을 입력한다(그림 5.4(c)).

## 제 5.4 절 산점도와 추세선

2차 평면에서 두 개 축이 교차하는 점을 산점이라 하고, 추세선은 산점들과 오차가 최소가 되는 직선이나 곡선이다. 이 절은 차트에 산점을 추가하고 추

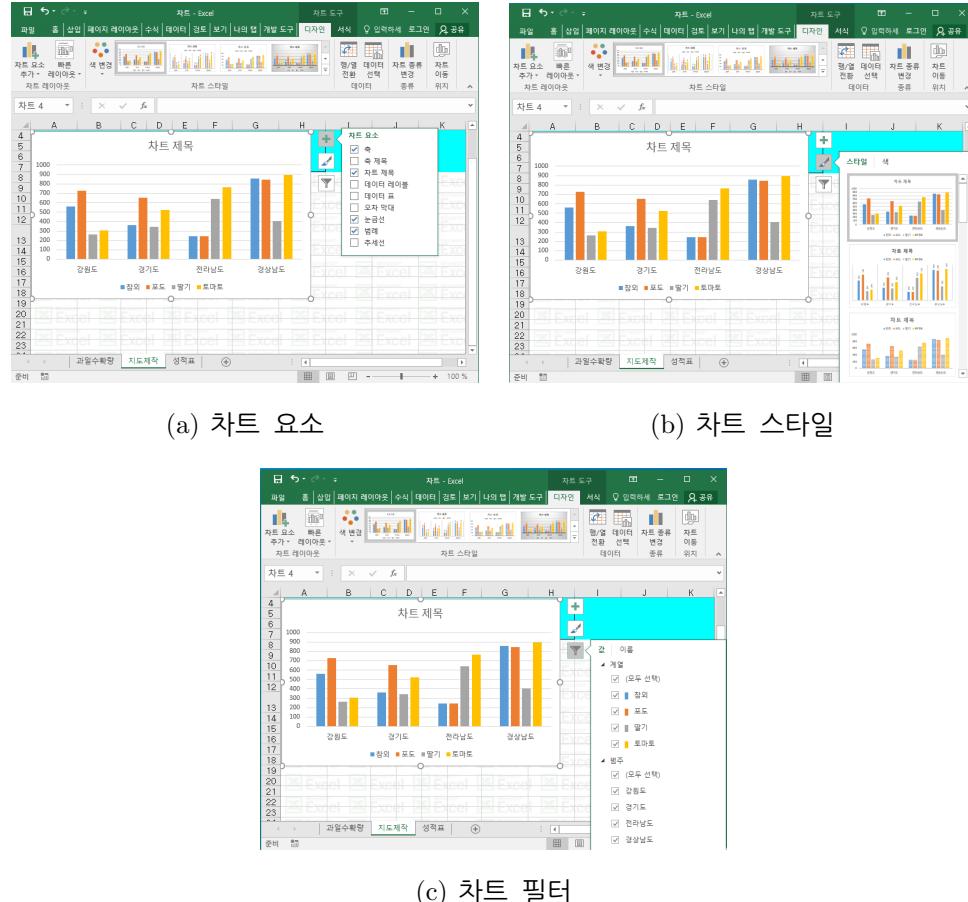


그림 5.3.2: 차트 서식

세션을 그려본다. 다음은 산점도와 직선 그래프 그리고 두 변수의 상관계수를 구한다.

- 시트에 있는 변수가 차트 작성에 모두 사용하면 현재셀이 데이터 셋에 있으면 되지만 일부 변수만 사용하면 해당 변수를 블록으로 지정한다. 차트에서 키와 몸무게 변수만 사용하기에 그림 5.1(a)에서는 두 변수에 블록을 설정한다 .

- 삽입 → 추천 차트 → 분산형 메뉴를 선택한다(그림 5.1(b)). 대략적인

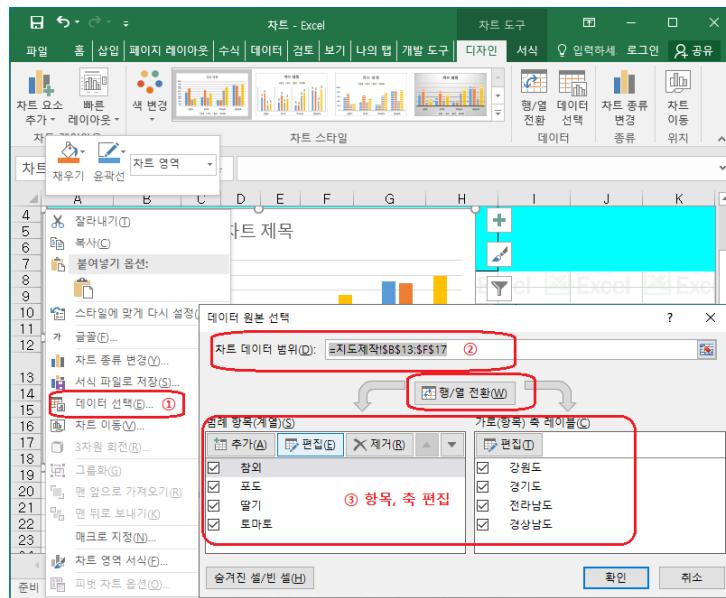


그림 5.3.3: 데이터 선택

그림이 완성된다. 그림 수정은 세부 옵션을 선택하여 진행한다.

3. 추세선은 산점에서 마우스 오른쪽 클릭하고 추세선 추가 메뉴를 선택한다(그림 5.1(c)).
4. 다음은 차트를 수정하는 세부 옵션이다.
  - (a) 추세선 서식 → 추세선 옵션에서 R-제곱 값을 차트에 표시
  - (b) 추세선 서식 → 추세선 옵션 → 선에서 실선을 선택, 너비는 4pt, 대시 종류는 실선,
  - (c) 데이터 요소 서식은 차트에서 산점을 클릭하면 나타난다. 산점 크기는 데이터 계열 서식 → 표식 옵션 → 기본제공 크기 → 3으로 설정하면 기본 설정한 값 5보다 작게 된다.
  - (d) 차트에서 몸무게 범위는 없는 범위까지 포함되어 있어 수정이 필요하다. 몸무게 범위 조정은 세로축에서 마우스 오른쪽 클릭하여

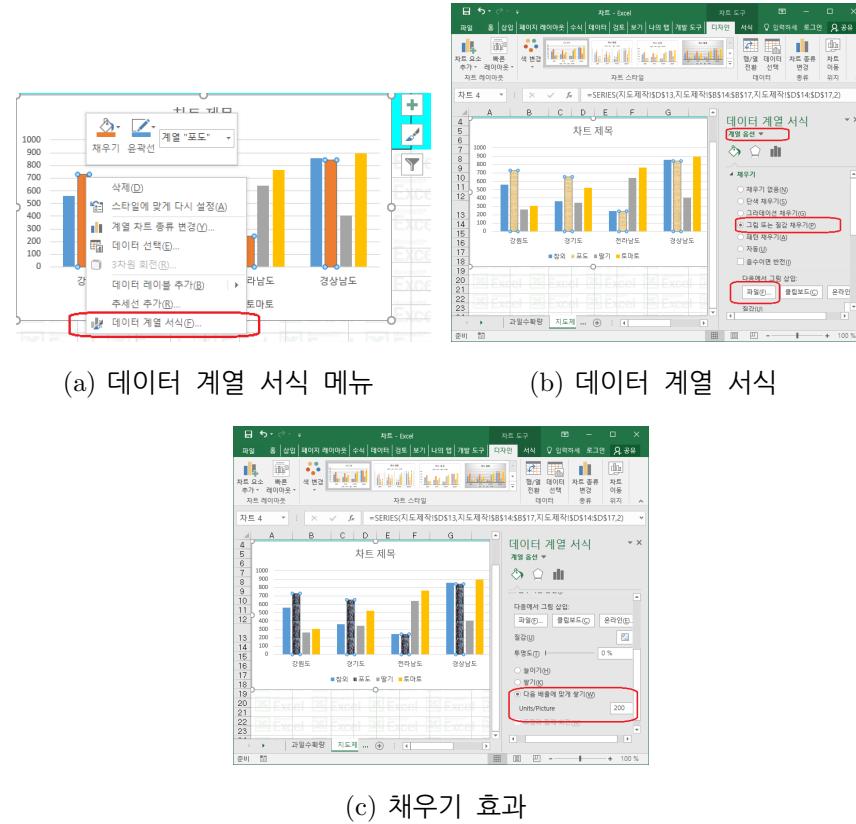


그림 5.3.4: 막대에 그림 넣기

축 서식을 선택한다. 축 서식에서 축 옵션 → 경계 최소값에 30을 입력한다.

5. 그림 5.4.2는 산점도와 단순회귀직선 그리고 상관계수를 계산한 차트이다.  $R^2 = 0.4191$ 이므로 상관계수  $R = 0.6474$ 이다.



그림 5.4.1: 산점도와 직선 그리고 상관수

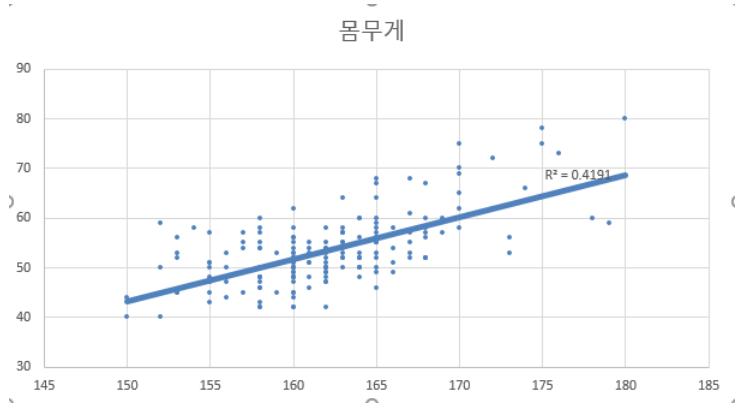


그림 5.4.2: 완성한 선점도와 회귀직선

## 제 5.5 절 곡선 그래프와 산점도

곡선 그래프는 2 차원 좌표 평면에서 점과 점을 서로 연결하여 그리는 그림이다. 만일 선과 선을 연결하지 않고 좌표만 그리면 산점도(scatter plot)가 된다. 다음은 곡선 그래프를 그리는 과정이다.

1. 현재셀을 데이터 셋에 옮긴다(그림 5.5.1).
2. 삽입 → 차트 → 분산형(X, Y) 또는 거품형 차트 삽입 메뉴를 선택하거나
3. 삽입 → 차트 → 추천 차트 → 꺥은 선형 메뉴를 선택한다(그림 5.5.2). 사실 그래프 목표는 첫 번째 분산형을 그리는 것이다. 그러나 분산형을 선택하면 자동으로 곡선 그래프와 산점도가 그려지므로 중간 과정을 알 수 없다. 그래서 자세한 설명이 필요한 꺥은 선형을 선택하였다.
4. 기본으로 그린 곡선과 산점도이다(그림 5.5.2). 몇 가지 설정하면 원하는 산점과 가로축 레이블의 변경이 가능하다.
  - ◉ 차트 요소 : 차트에 차트 제목, 데이터 레이블, 범례를 선택하거나 제거할 수 있다(그림 5.3(a)).
  - ◉ 차트 스타일 : 차트 스타일을 선택한다(그림 5.3(b)).
  - ◉ 차트 필터 : 차트에 표현할 값과 이름을 선택한다(그림 5.3(c)).
5. 데이터 선택 : 차트에서 사용할 데이터를 선택 및 변경한다(그림 5.5.4).
  - ◉ 차트 데이터 범위 : 차트 데이터 범위를 직접 변경한다. 그림 5.4(a)) 는 범례 항목(계열)에서 자료를 수정하면 그 내용이 즉시 적용된다.
  - ◉ 범례 항목(계열) : 범례를 추가, 편집, 제거한다. 변수 X를 제거한다.

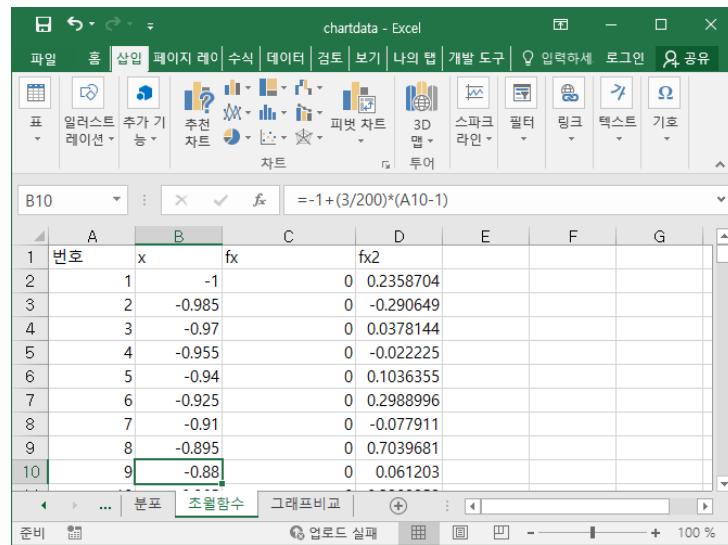


그림 5.5.1: 차트 현재셀 위치

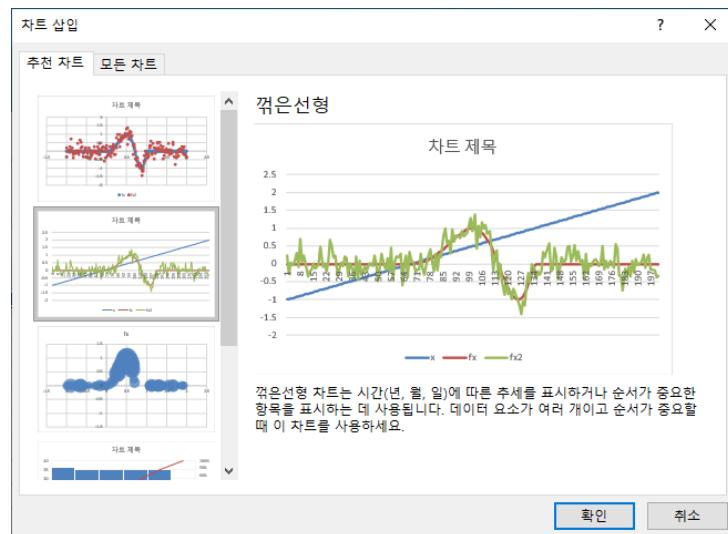


그림 5.5.2: 차트 삽입

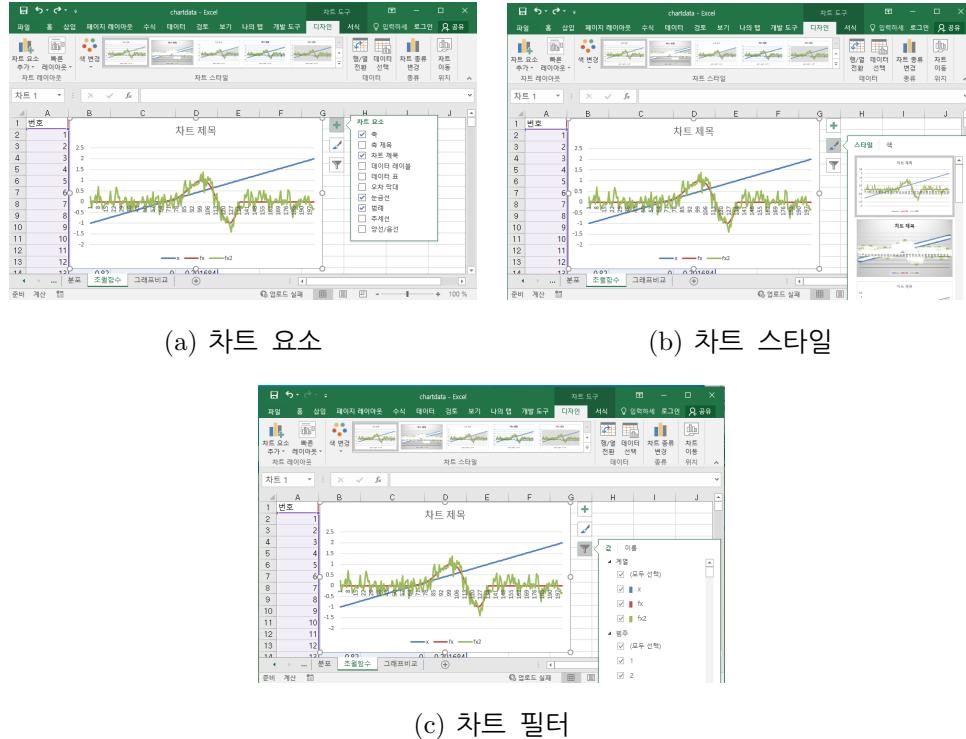


그림 5.5.3: 차트 서식

- **가로(항목) 축 레이블** : 가로 축 레이블을 설정한다. 기본값은 셀에서 가져오고 변경은 편집 버튼을 누르고 한다. 가로(항목) 축 레이블은 변수 X로 변경한다(그림 5.4(b)).

6. 데이터 계열 서식 : 곡선과 산점은 그림 5.5(a)에서 데이터 계열 서식 메뉴를 실행한다. 오른쪽 계열 옵션 → 채우기 및 선에서

- **선은** 그릴 선의 설정사항으로 없음을 선택하여 산점만 나오게 한다,
- **표식은** 산점의 모양과 크기 설정으로 표식 옵션 → 기본 제공 → 원모양을 선택한다(그림 5.5(b)).

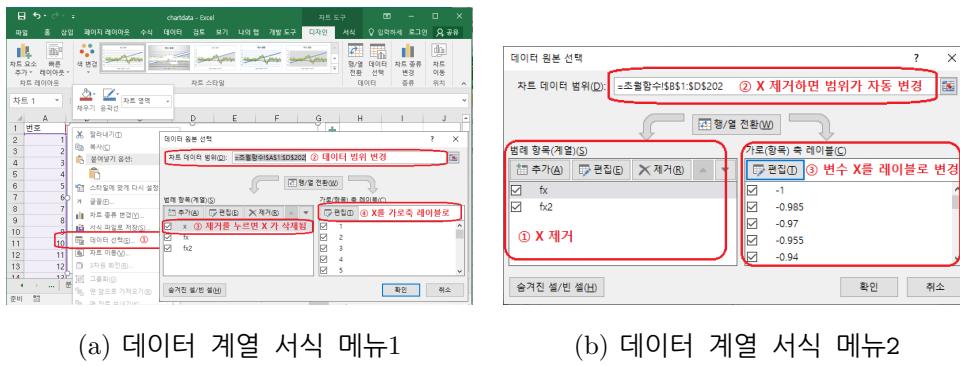


그림 5.5.4: 데이터 계열 서식

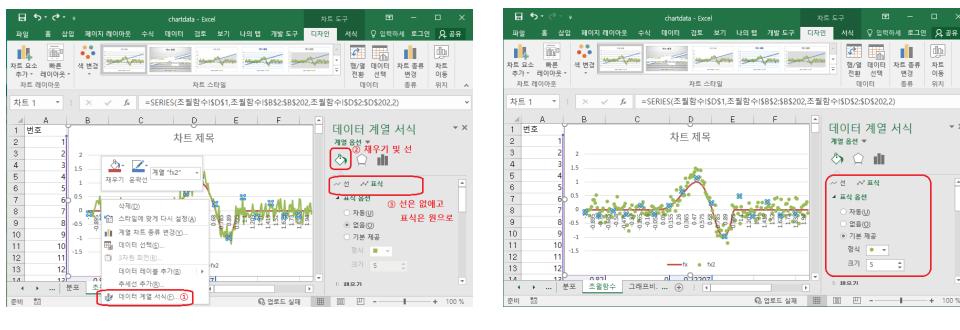


그림 5.5.5: 곡선 그리기

## 제 5.6 절 도수분포표

도수분포표(度數分布表, frequency table)는 관측(觀測)값을 몇 개의 범주로 나눈 다음 그 범주에 속하는 관측값의 개수(도수, frequency)와 그 도수를 전체 관측값의 개수로 나눈 값(상대(相對) 도수, relative frequency)에 대한 표이다. 범주형 자료는 해당 범주에 대한 것에 대한 빈도를 계산하고 연속형자료는 각 범주를 범위로 지정한다.

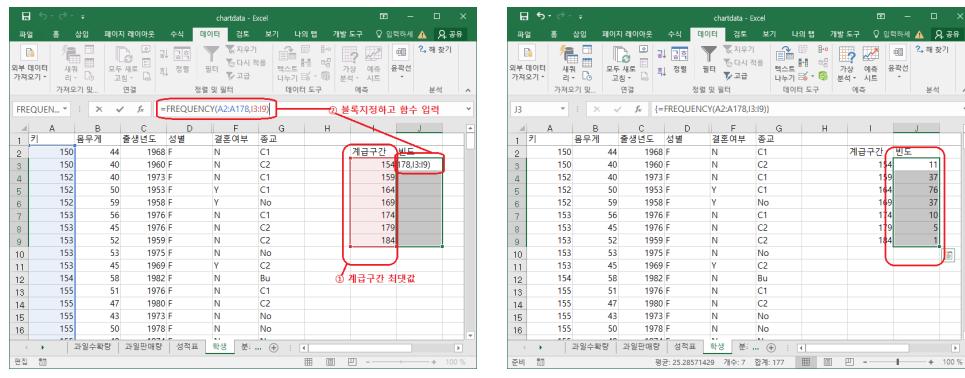
엑셀에서 도수분포표는 FREQUENCY 함수를 사용한다. 함수 사용법은

FREQUENCY(*data\_array*, *bins\_array*)

이며, 각 입력값은

- ◉ *data\_array* : 도수분포표를 구할 데이터 범위이다.
- ◉ *bins\_array* : 계급구간 정보가 있는 배열이다. 계급구간 정보는 입력값보다 작거나 같은 값의 갯수를 구한다.

이 함수는 배열을 계산하기 때문에 배열 수식을 입력해야 한다. 즉 계산 방법은 행렬 계산 방법과 동일하다(4.8 절, 107 페이지). 다음은 실습 파일로 도수분



(a) 블록지정 및 함수입력

(b) 도수분포표 결과

그림 5.6.1: 도수분포표

포표를 계산하는 과정이다.

1. 계급구간에서 최대값을 입력한다(그림 5.1(a)).
2. 빈도를 구할 셀 영역의 블록 선택하고 함수 마법사에서 FREQUENCY 함수를 선택한다.
3. *data\_array*는 빈도를 구할 데이터, *bins\_array*는 계급구간을 입력한다.

4. Ctrl + Shift + Enter 키를 누르면 도수분포표 결과가 그림 5.1(b)에 계산된다.

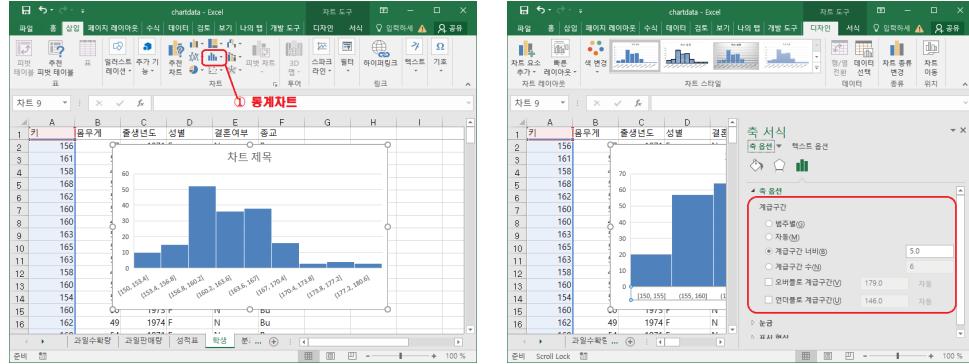
## 제 5.7 절 히스토그램

히스토그램(histogram)은 연속형 자료에서 둑수분포표를 기초로 각 계급구간(bin)에 대한 둑수나 상대둘수를 막대형태로 그린 그림이다. 히스토그램은 다음과 같은 특징이 있다.

- ◉ 계급구간(bin) : 히스토그램에서 변수를 구간으로 제시한 것으로 다른 계급구간과 교차되는 부분은 없다.
- ◉ 계급구간 높이는  $\frac{\text{상대도수}}{\text{계급구간 폭(width, 幅)}}$  또는  $\frac{\text{도수}}{\text{계급구간 폭(width, 幅)}}$ 이고
- ◉ 계급구간 높이가  $\frac{\text{상대도수}}{\text{계급구간 폭(width, 幅)}}$ 이면 모든 면적은 1
- ◉ 각 계급구간은 길이가 같지 않아도 됨
- ◉ 연속형 자료에서 용이(容易)하게 사용됨
- ◉ 가장 적합한 히스토그램은 사용자가 범주의 개수를 변경하면서 여러 히스토그램을 작성하여 자료의 윤곽(輪廓)을 파악하는데 용이한 히스토그램을 선택
- ◉ 계급구간의 수와 구간이 같더라도 시작점이 다르면 그래프의 전체적인 형태가 변하는 단점(短點)이 있음

다음은 히스토그램을 그리는 과정이다.

1. 현재셀을 데이터 셋에 옮긴다. 히스토그램은 한 변수를 그리므로 데이터 셋에 여러 변수가 있으면 해당 변수를 블록지정하는 것이 더 편리하다.
2. 삽입 → 차트 → 통계 차트 삽입 → 히스토그램 메뉴를 선택하거나



(a) 히스토그램 그리기

(b) 계급구간 설정

그림 5.7.1: 히스토그램

3. 삽입 → 차트 → 추천 차트 → 히스토그램 메뉴를 선택한다(그림 5.1(a)).

4. 축 서식 → 축 옵션에서 계급구간을 설정한다. 계급구간은

- ◉ 범주별 : 범주형 자료인 경우 계급구간을 범주로 표현할 때
- ◉ 자동 : 계급구간을 자동으로 설정할 때
- ◉ 계급구간 너비 : 계급구간의 너비를 설정한다. 그림 5.1(b)에서 계급구간 너비는 5로 하였다.
- ◉ 계급구간 수 : 계급구간의 개수를 설정

파레토 그림(pareto chart)은 히스토그램을 뜻수 순으로 그리고 같은 그림에 격은선 그래프를 추가한다. 히스토그램에서 차트 변경 메뉴를 선택하여 파레토 그림으로 변경할 수 있다. 그림 5.7.2는 그림 5.1(b) 히스토그램을 파레토 그림으로 변경하였다. 방법은 그림 5.1(b)에서 차트 종류 변경 메뉴를 선택하여 파레토를 선택한다.

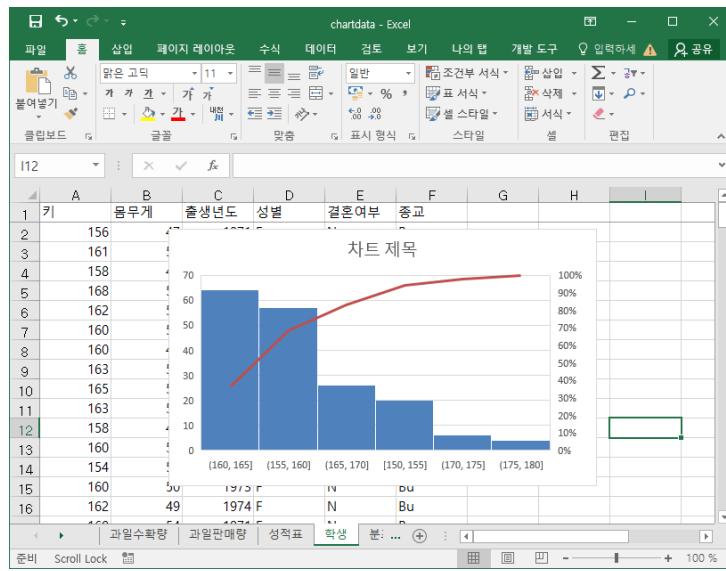


그림 5.7.2: 파레토그램

## 제 5.8 절 상자 수염 그림

상자 수염 그림(box – whisker plot)은 최솟값(最小)값, 제 1 사분위수(Q1), 제 2 사분위수(Q2), 제 3 사분위수(Q3), 최댓값 등 5 개의 통계량을 이용하여 상자와 수염으로 작성한 그림으로 상자 그림(box plot)이라고도 부른다.

사분위수(四分位數, quartile)는 자료를 오름차순으로 정렬하고 4등분 한 값으로

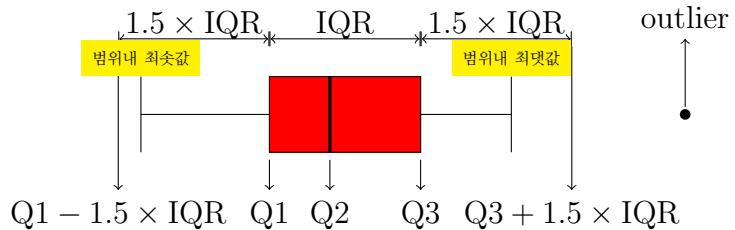
- ◉ 제 1 사분위수(Q1)은 크기가 25%에 위치한 값이고,
- ◉ 제 2 사분위수(Q2)는 크기가 50%에 위치한 값으로 중앙값이라도 부르며,
- ◉ 제 3 사분위수(Q3)은 크기가 75%에 위치한 값이다.

다음은 상자 수염 그림의 특징이다.

- ◉ 상자 그림(box plot)이라고도 부름

- ◉ 자료의 범위(範圍), 사분위수(四分位數)의 위치 파악에 용이
- ◉ 이상점<sup>1</sup>(異常點, outlier) 판별(判別)에 사용
- ◉ 두 개 이상의 변수에도 상자그림 작성(作成) 가능
- ◉ Q2 : 중앙(中央)값
- ◉ IQR(Interquartile Range) : 사분위수 범위(Q3–Q1)

다음 그림은 상자 수염 그림을 그린 예이다.



상자 수염 그림에서 이상점 판별은 다음과 같다.

1.  $Q1 - 1.5 \times IQR$ 보다 작은 값이나  $Q3 + 1.5 \times IQR$  보다 큰 값은 이상점
2. 특히  $Q1 - 3 \times IQR$ 보다 작은 값이나  $Q3 + 3 \times IQR$  보다 큰 값은 지극히 한 쪽에 몰려있다는 의미로 극단값(extreme value)라고 부른다.

다음은 상자 수염 그림을 그리는 과정이다.

1. 현재셀을 데이터 셋에 옮긴다. 상자 수염 그림은 여러 변수를 그림에 표현할 수 있다. 각 변수의 자료 수가 같지 않아도 그림 그리기가 가능하다. 각 변수는 열로 구분한다. 예를 들어 성별로 정렬하고 남자키를 복사하여 한 열에 옮기고 여자키를 복사하여 다른 열에 옮긴 후 그러면 두 변수의 상자 수염 그림이 만들어진다.

---

<sup>1</sup> 실험 또는 관찰을 통하여 데이터를 수집하였을 때, 데이터의 전반적인 흐름에서 벗어나는 데이터이다.

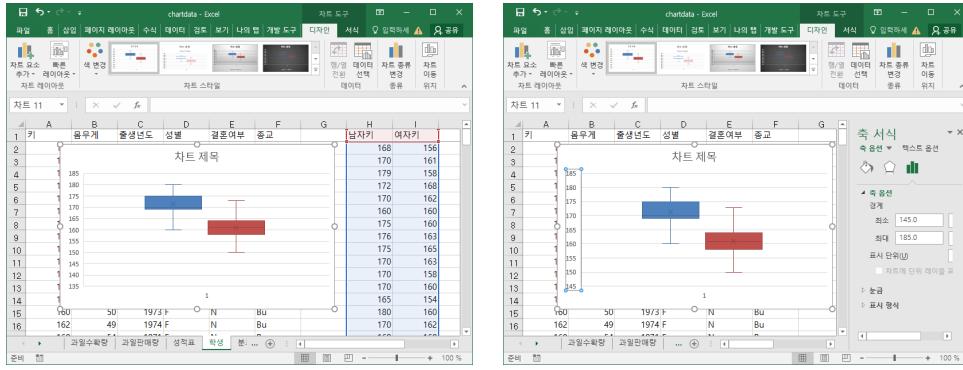


그림 5.8.1: 상자 수염 그림

2. 삽입 → 차트 → 통계 차트 삽입 → 상자 수염 그림 메뉴를 선택하거나
3. 삽입 → 차트 → 추천 차트 → 상자 수염 그림 메뉴를 선택한다(그림 5.1(a)).
4. 세로축을 선택하고 축 서식 → 축 옵션에서 최소에 145, 최대에 185(그림 5.1(a)).
5. 이상점은 데이터 계열서식 → 계열 옵션에서 외부 요소 표시를 선택하면 나타난다.

## 제 5.9 절 이중 축 혼합 차트

단위가 다른 두 종류의 값에 대한 차트를 그려보자. 그림 5.9.1은 단위가 다른 두 변수의 관측값을 한 차트에 그렸다. 다음은 두 개 축에 혼합 차트를 그리는 과정이다.

1. 현재셀을 데이터 셋에 옮긴다(그림 5.9.2).

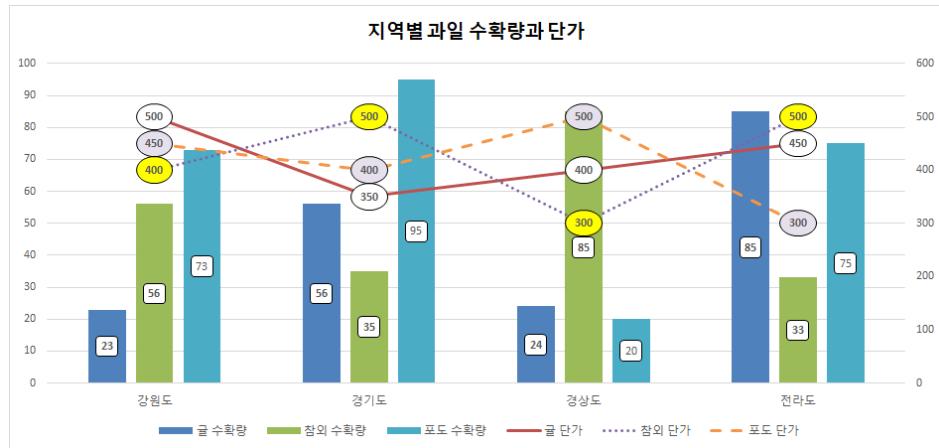


그림 5.9.1: 혼합 차트 완성

2. 삽입 → 차트 → 세로 또는 가로 막대형 차트 삽입 메뉴를 선택하거나
3. 삽입 → 차트 → 추천 차트 → 모든 차트 → 묶은 세로 막대형 메뉴를 선택한다(그림 5.9.3).
4. 기본으로 그린 막대그래프이다(그림 5.9.3). 왼쪽 세로축은 기본축으로 수확량, 오른쪽 세로축은 보조축으로 단가로 사용한다. 보조축에는 꺽은 선형 차트를 그린다.
5. 그림 5.4(a)에서 참외 단가 막대에 마우스 옮기고 오른쪽 버튼을 클릭하여 계열 차트 종류 변경 메뉴를 선택한다.
6. 그림 5.4(b)에서 차트 종류는 콤보, 데이터 계열에 대한 차트 종류와 축을 선택합니다:에서 모든 과일 단가의 차트 종류는 꺽은 선형, 보조 축을 선택한다. 사용자 지정 조합에서 보조축이 꺽은 선형으로 변경된 것을 확인한다.
7. 그림 5.5(a)에서 참외 단가 계열에 마우스를 옮기고 데이터 레이블 추가 → 데이터 레이블 추가를 선택한다.

8. 그림 5.5(b)에서 추가한 데이터 레이블에 마우스를 옮기고 오른쪽 버튼을 클릭하여 데이터 레이블 서식 메뉴를 선택한다.
9. 그림 5.9.6의 설정은 채우기 및 선에서 채우기는 단색 채우기, 색은 노란색, 테두리는 실선이고 색은 검정색이다. 레이블 옵션에서 레이블 위치는 오른쪽에서 가운데로 변경한다.
10. 차트 크기를 크게하고 차트 제목을 지역별 과일 수확량과 단가로 변경 한다.

	과일	귤		참외		포도	
		수확량	단가	수확량	단가	수확량	단가
4	시도						
5	강원도	23	500	56	400	73	450
6	경기도	56	350	35	500	95	400
7	경상도	14	400	85	300	20	500
8	전라도	85	450	33	500	75	300
9							
10							

그림 5.9.2: 차트 현재셀 위치

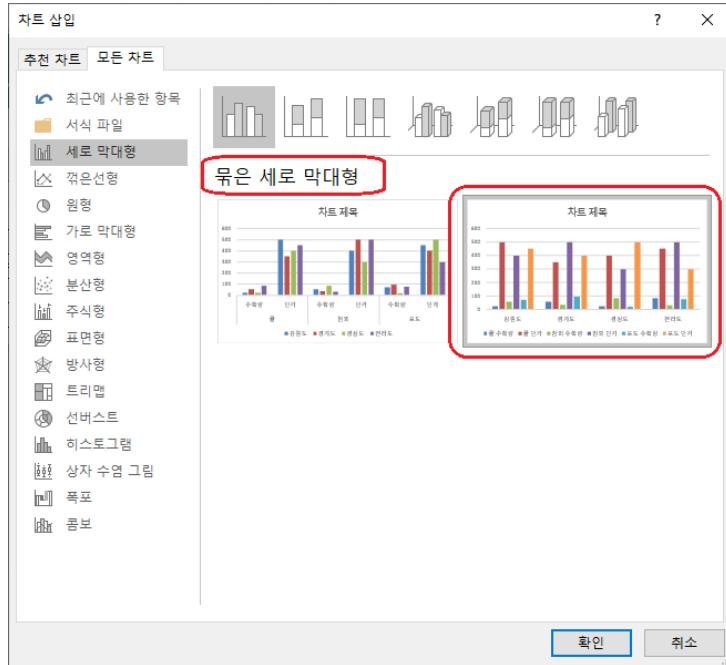
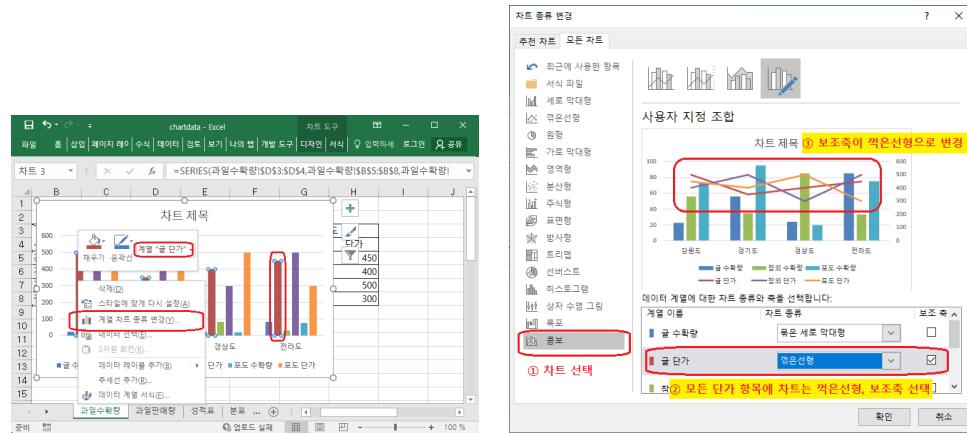


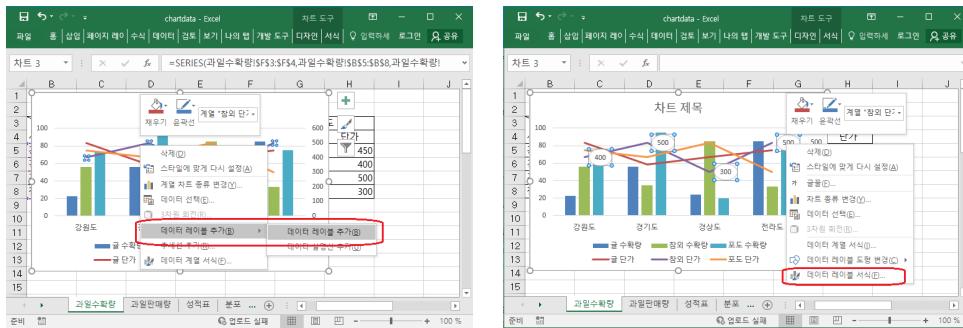
그림 5.9.3: 차트 삽입



(a) 계열 차트 종류 변경 메뉴

(b) 계열 차트 종류 변경 설정

그림 5.9.4: 계열 차트 종류 변경



(a) 데이터 레이블 추가 메뉴

(b) 데이터 레이블 서식 메뉴

그림 5.9.5: 데이터 레이블 추가

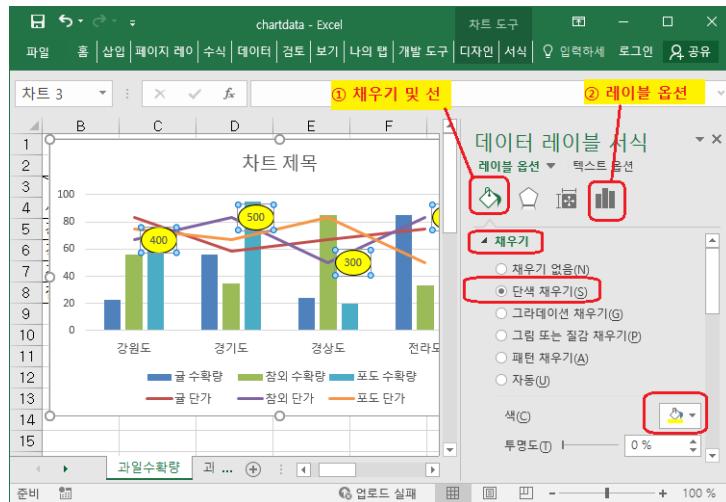


그림 5.9.6: 데이터 레이블 서식 설정

## 제 6 장

### 데이터베이스

데이터의 양이 적은 경우에는 간단하게 에디터나 기타 스프레드시트 응용프로그램을 통하여 그 내용을 확인할 수 있다. 그러나 데이터 양이기가(giga byte) 단위 이상인 경우에는 응용프로그램을 사용하여 데이터의 내용을 확인할 수 없고 데이터베이스를 이용해야 한다.

데이터베이스는 자료의 검색과 갱신을 목적으로 효율성을 증대시키기 위하여 관리되는 데이터의 집합이다. 다음은 데이터베이스에서 사용하는 용어를 소개한다.

- 레코드(record)는 각 행에 저장할 수 있는 데이터의 케이스에 대한 정보이고
- 필드(field)는 각 열에 변수값으로 저장할 수 있는 것을 의미한다.

이 장에서는 데이터베이스와 관련된 메뉴들과 함수들에 대하여 알아보자.

#### 제 6.1 절 정렬

정렬은 데이터를 오름차순이나 내림차순 순으로 배열하는 것이다. 엑셀에서 정렬은 한 변수 이상의 데이터가 포함되어 있어도 가능하다. 한 열이나 행을 기

준으로 정렬하는 간단한 방법은 정렬하려는 필드에서 한 값을 선택하여 현재 셀을 위치한 후

- ◉ 데이터를 오름차순 버튼 을 클릭하여 정렬하거나
- ◉ 데이터를 내림차순 버튼 을 클릭하여 정렬할 수 있다.

약간 복잡한 정렬은 다음과 같은 과정으로 진행한다.

1. 우선 정렬을 실행하기 전에 현재 셀이 데이터가 있는 있는 곳에 있어야 한다.

	A	B	C	D	E	F
1	지역	종류	수확량	단가		
2	강원도	귤	23	5		
3	강원도	포도	73	5		
4	경기도	귤	56	2		
5	경기도	포도	95	1		
6	경상도	귤	24	4		
7	경상도	포도	20	3		
8	전라도	귤	85	1		
9	전라도	포도	75	4		
10						

그림 6.1.1: 현재 셀의 위치

2. 정렬은 데이터→정렬 메뉴 를 클릭한다. 그러면 그림 6.1.2의 창이 나타나고 여기에
  - ◉ 내 데이터에 머리글 표시는 데이터에서 첫 행을 머리글로 사용할 것인지 선택

- ◉ 열→정렬기준은 정렬할 필드를 선택
- ◉ 정렬기준은 값, 셀 색, 글꼴 샘플, 셀 아이콘에서 정렬할 기준을 선택
- ◉ 정렬은 오름차순, 내림차순, 사용자 지정 목록에서 정렬할 종류를 선택
- ◉ 기준추가는 정렬조건을 위에서 설정한 것과 같이 더 추가할 수 있음
- ◉ 옵션은 정렬 방향을 위쪽에서 아래쪽으로 선택하면 행 방향으로 정렬되고 왼쪽에서 오른쪽으로 선택하면 열 방향으로 정렬된다.

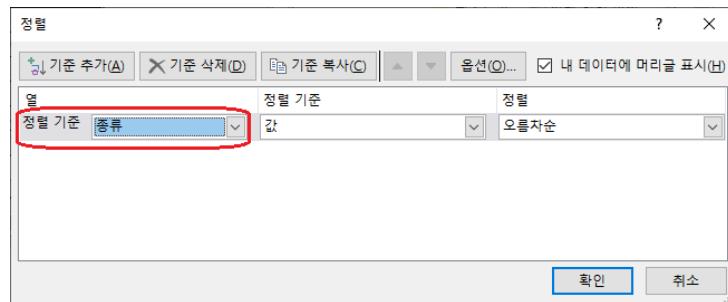


그림 6.1.2: 정렬 조건 선택

3. 그림 6.1.3에서 데이터가 첫 열을 기준으로 정렬된 것을 확인할 수 있음

## 제 6.2 절 필터

필터(filter)는 사용자가 원하는 자료를 설정한 조건에 따라 추출하는 것으로 그 종류는

- ◉ 자동 필터
- ◉ 고급 필터

가 있다.

	A	B	C	D	E	F
1	지역	종류	수확량	단가		
2	강원도	귤	23	5		
3	경기도	귤	56	2		
4	경상도	귤	24	4		
5	전라도	귤	85	1		
6	강원도	포도	73	5		
7	경기도	포도	95	1		
8	경상도	포도	20	3		
9	전라도	포도	75	4		
10						

그림 6.1.3: 정렬 결과 확인

먼저 자동필터에 대하여 알아보자. 그림 6.1.3에서 데이터→필터( )를 실행하면 그림 6.2.1의 화살표 버튼( )이 나타난다. 사용자가 뽑으려는 데이터는 이 화살표 버튼을 누르고 데이터의 특정한 값을 선택하거나 특정한 조건을 설정할 수 있다(그림 6.2.1). 자동 필터의 설정 해제는 필터 설정과 동일하게 데이터→필터 메뉴를 선택하거나 메뉴를 선택하면 설정이 해제된다.

고급 필터는 실행전에 데이터의 필터링 조건을 먼저 설정해야 한다. 필터링 조건은 우선 필드명을 한 행에 입력하고 필드의 조건을 필드명 아래에 설정한다.

조건을 행에 설정하면 모두 일치하는 조건만 필터링 되고 추가적인 조건 설정은 바로 다음 행에 한다. 이렇게 추가한 조건은 이미 설정된 조건과 상관없이 추가된 조건에 맞는 자료를 추출하며 동일한 레코드는 한번만 추출한다.

필터 명령은 그림 6.2.2에 필터링 할 데이터(왼쪽)와 필터링 할 조건(오른쪽)을 가지고 실행한다. 고급필터는 데이터→고급 메뉴를 실행하면 그림 6.2.3

지역	종류	수확량	단가
강원도	귤	23	5
경기도	귤	56	2
경상도	귤	24	4
전라도	귤	85	1
강원도	포도	73	5

그림 6.2.1: 필터 결과

지역	종류	수확량	단가
강원도	귤	23	5
경기도	귤	56	2
경상도	귤	24	4
전라도	귤	85	1
강원도	포도	73	5

고급 필터 조건

강원도	귤	>=75
-----	---	------

그림 6.2.2: 고급필터를 위한 조건 설정

창이 나타나며

- 필터링한 결과는 현재 위치에 필터, 다른 장소에 복사 중 하나를 선택

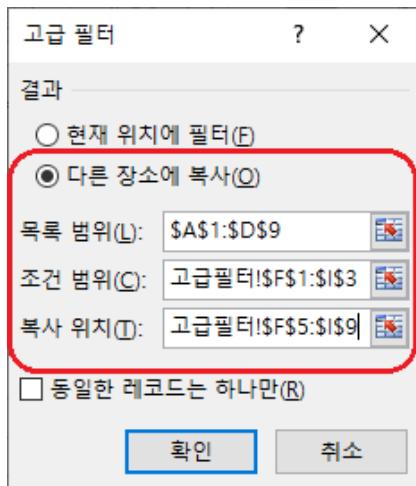


그림 6.2.3: 고급필터 설정

하여 저장할 수 있다.

- ✖ 현재 위치에 필터는 원 자료가 있는 곳에 필터링 결과를 보여 주며 원래 데이터가 사라진다.
- ✖ 다른 장소에 복사는 필터링 결과가 별도로 지정한 위치에 보여진다.

- ◉ 목록 범위는 원 데이터가 있는 곳을 입력한다.
- ◉ 조건 범위는 필터링한 조건이 있는 곳을 입력한다.
- ◉ 복사 위치는 필터링한 결과를 다른 장소에 복사하기를 원하는 경우에 그 위치를 설정한다.
- ◉ 동일한 레코드는 하나만은 필터링한 결과 해당 레코드가 두 개 이상인 경우 하나만 표시하도록 설정한다.

고급 필터 과정이 모두 올바르게 실행되면 그림 6.2.4와 같은 결과가 보인다.

A	B	C	D	E	F	G	H	I
1	지역	종류	수확량	단가				
2	강원도	귤	23	5	강원도			
3	경기도	귤	56	2		귤	$\geq 75$	
4	경상도	귤	24	4				
5	전라도	귤	85	1				
6	강원도	포도	73	5				
7	경기도	포도	95	1				
8	경상도	포도	20	3				
9	전라도	포도	75	4				
10								

고급필터 적용 결과

그림 6.2.4: 고급필터 설정 결과

### 제 6.3 절 유효성 검사

유효성 검사는 셀에 값을 입력할 때

- ◉ 특정한 데이터의 형태만 입력하도록 하거나
- ◉ 데이터의 입력범위를 통제하는 기능을 제공하고

부가적으로 데이터를 셀에 입력할 때

- ◉ 입력범위에 대한 메시지를 알려주거나
- ◉ 입력범위를 벗어나면 경고 메시지가 나오게 설정할 수 있다.

유효성 검사는 적용할 셀을 선택하고 데이터→데이터 유효성 검사 메뉴를 누르면 그림 6.3.1과 같은 데이터 유효성 창이 나타난다.

- ◉ 설정 : 유효성 조건에 제한 대상 설정으로
  - ✖ 정수 : 정수만 입력할 수 있고 조건은 최소값과 최대값을 설정
  - ✖ 소수점 : 실수 값 입력에 최소값과 최대값을 설정

- ✖ 목록 : 목록 중 하나를 선택하여 해당 리스트를 직접 입력할 수도 있고, 셀에서 범위를 선택할 수도 있음
  - ✖ 날짜 : 날짜 입력 범위 지정
  - ✖ 시간 : 시간 입력 범위 지정
  - ✖ 텍스트 길이 : 문자만 입력 가능하도록 숫자 길이의 범위를 지정
  - ✖ 사용자 정의 : 사용자가 정의한 것만 입력되도록 수식에 조건을 지정
- ◉ 설명 메시지 : 사용자가 셀을 선택하면 나타나는 메시지로 제목과 설명 메시지를 입력한다. 이 설명 메시지는 셀을 선택하면 설명 메시지 표시를 체크 표시를 선택해야 해당 셀을 선택하였을 때, 제목과 설명 메시지 내용이 나타난다.
  - ◉ 오류 메시지 : 유효하지 않은 데이터를 입력하면 나타낼 오류 메시지는 유효성 검사에서 설정된 값이 아닌 값이 입력되면 오류 메시지가 나타나며 이 경우
    - ✖ 중지 : 데이터 입력 오류가 발생하였을 때 데이터의 입력을 중지
    - ✖ 경고 : 데이터 입력 오류가 발생하였을 때 경고창이 나타나며 입력을 허용, 취소, 수정할 것인지 선택할 수 있음
    - ✖ 정보 : 데이터 입력 오류가 발생하였을 때 정보창이 나타나며 입력을 허용, 취소할 것인지 선택할 수 있음이것들 중 하나를 선택할 수 있고 그 때 오류 메시지에 대한 설명을 설정할 수 있다.
  - ◉ IME 모드 : 유효성 검사가 적용된 셀을 선택한 경우 입력할 수 있는 문자를 설정한 것으로
    - ✖ 현재 상태 유지 : 현재 컴퓨터에 설정된 문자 입력 상태 유지

✖ 영문 전자 : 영문 전자 입력 상태

✖ 영문 : 영문 입력 상태

✖ 한글 전자 : 한글 전자 입력 상태

✖ 한글 : 한글 입력 상태

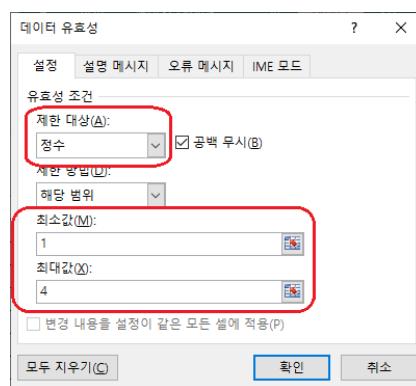


그림 6.3.1: 데이터 유효성 검사

## 제 6.4 절      부분합

SUM 함수는 주어진 변수에 대한 합을 구해주는 반면 SUBTOTAL 함수는 주어진 변수를 다른 부그룹(subgroup)로 나눈 후 부그룹별 합을 구한다.

그림 6.4.1 자료로 부분합을 구해 보자. 먼저 부분합은 현재 데이터 셋에 현재의 셀이 있어야 하고 이어지는 선행조건은 부분합을 구하려는 필드가 오름차순 또는 내림차순으로 정렬되어 있어야 한다. 부분합 실행은 데이터→윤곽선→부분합 메뉴를 클릭한다. 실행 결과는 그림 6.4.2와 같은 부분합 설정 창이 나타난다. 이 창에

- 그룹화할 항목 : 부분합을 구할 때 부그룹을 설정

	A	B	C	D	E	F	G	H	I
4									
5		부서명	성명	급호	성별	직위	급여	상여금	
6	관리팀	조영필	2106	남	과장	1,600,000원	560,000원		
7	관리팀	김영길	2210	남	과장	1,670,000원	584,500원		
8	관리팀	이경자	3213	여	사원	870,000원	304,500원		
9	기획팀	나훈아	2207	남	과장	1,500,000원	525,000원		
10	기획팀	이정길	3112	남	대리	1,200,000원	420,000원		
11	기획팀	정운아	3111	여	대리	1,100,000원	385,000원		
12	총무팀	최진실	3114	여	대리	1,000,000원	350,000원		
13	총무팀	이정재	3214	남	사원	840,000원	294,000원		
14	총무팀	최불암	3215	남	사원	850,000원	297,500원		
15									

The screenshot shows an Excel spreadsheet titled "database - Excel". The data is organized into a table with columns: 부서명 (Department), 성명 (Name), 급호 (Extension), 성별 (Gender), 직위 (Position), 급여 (Salary), and 상여금 (Overtime). Row 10 is highlighted with a green background, and cell D10 (containing the value 3112) is selected. A red circle highlights cell D10. The ribbon at the top has tabs for 고급필터 (Advanced Filter), 부분합 (Subtotal), 해찾기1 (Find & Replace), 해찾기2 (Find & Replace), and other options.

그림 6.4.1: 부분합 계산 전 현재 셀 위치

- ◉ 사용할 함수 : 부분합을 구할 함수를 설정하는 것으로 합계, 개수, 평균, 최대값, 최소값 등이 있음
- ◉ 부분합 계산 항목 : 데이터 셋에 저장된 필드에서 선택하여 여러 가지를 선택할 수 있음
- ◉ 새로운 값으로 대치 : 사용할 함수에서 여러 가지 함수를 반복적으로 사용하였을 때, 먼저 계산한 것은 후에 계산한 값에 지워진다. 따라서 구한 함수값이 모두 나타나게 하려면 이 메뉴를 선택하지 않으면 된다.
- ◉ 데이터 아래에 요약 표시 : 데이터의 요약을 표시할 것인지 아닌지에 대한 설정
- ◉ 모두 제거 : 부분합으로 구한 모든 것을 삭제

한다. 그림 6.4.3에 부분합을 구한 결과가 보인다.

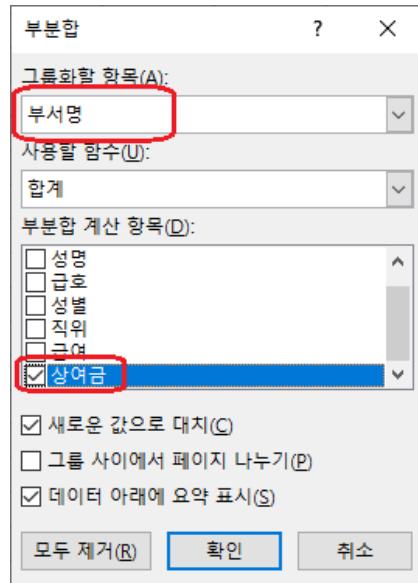


그림 6.4.2: 부분합 설정

	A	B	C	D	E	F	G	H
1	부서명	성명	급호	성별	직위	급여	상여금	
6	관리팀	조영필	2106	남	과장	1,600,000원	560,000원	
7	관리팀	김영길	2210	남	과장	1,670,000원	584,500원	
8	관리팀	이경자	3213	여	사원	870,000원	304,500원	
9	<b>관리팀 요약</b>							1,449,000원
10	기획팀	나훈아	2207	남	과장	1,500,000원	525,000원	
11	기획팀	이정길	3112	남	대리	1,200,000원	420,000원	
12	기획팀	정우아	3111	여	대리	1,100,000원	385,000원	
13	<b>기획팀 요약</b>							1,330,000원
14	총무팀	최진실	3114	여	대리	1,000,000원	350,000원	
15	총무팀	이정재	3214	남	사원	840,000원	294,000원	
16	총무팀	최불암	3215	남	사원	850,000원	297,500원	
17	<b>총무팀 요약</b>							941,500원
18	<b>총합계</b>							3,720,500원

그림 6.4.3: 부분합 계산 결과

## 제 6.5 절 목표값 찾기

목표값 찾기는 어떤 함수의 결과 값을 알고 있는 경우 입력 값을 구할 때 사용한다.

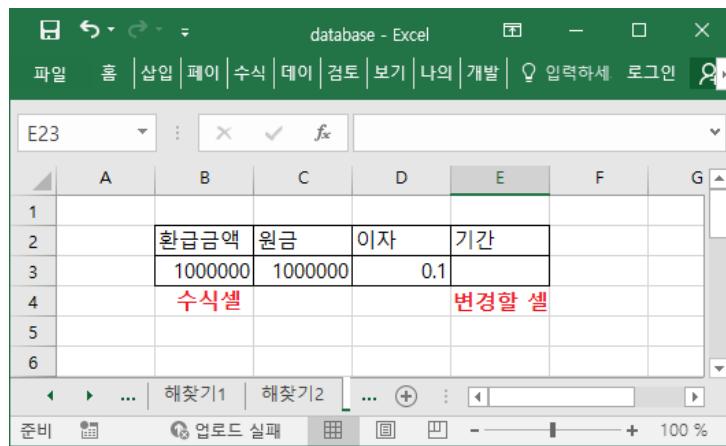


그림 6.5.1: 복리 계산

그림 6.5.1에 정기예금의 원금, 이자, 기간으로 계산된 만기환금액이 있다. 총 납입금액이 1,000,000원, 연이율이 10%일 때 만기 환급금액이 원금의 두 배인 2,000,000원이 되려면 납입기간은 얼마인지 계산해보자. 이 때 납입기간은 목표값 찾기를 사용하면 편리하게 계산할 수 있다. 복리 계산 식은

$$P = a(1 + r)^n$$

이며,  $P$ 는 만기 환급금액이고  $a$ 는 원금,  $r$ 은 이자율,  $n$ 은 기간이다.

목표값 찾기는 데이터→데이터 도구→가상 분석→목표값 찾기 메뉴를 클릭하면 그림 6.5.2의 목표값 찾기 창에 나타난다. 이 창에

- ① 수식 셀 : 수식이 있는 셀로 환급액 셀 선택
- ② 찾는 값 : 목표값으로 원금의 두 배인 2,000,000
- ③ 값을 바꿀 셀 : 환급액과 수식이 연결된 것으로 기간

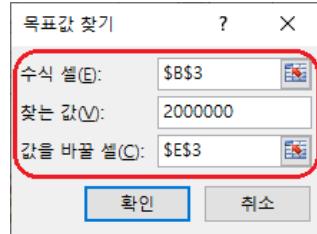


그림 6.5.2: 복리계산 설정

설정값을 입력한다. 만일 값을 찾을 경우에만 목표값이 출력되며 그렇지 않은 경우 적합한 값을 찾을 수 없다는 메시지가 출력된다. 그림 6.5.3에 목표값을 찾은 상태와 결과가 셀에 보이며

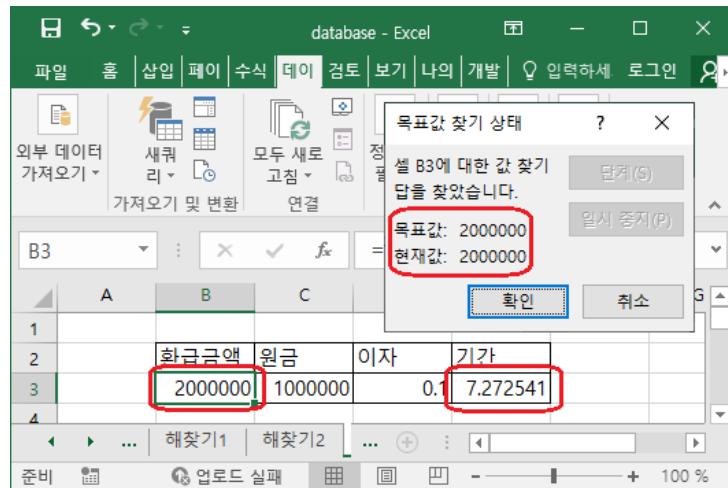


그림 6.5.3: 복리 계산 결과

목표값 찾기는 뉴턴 – 래슨(Newton-Raphson) 방법으로 해를 찾는다. 이 방법은 초기값에서 출발하여 참값과 근사값의 차이가  $10^{-3}$  보다 작은 경우의 값을 해로 찾아 준다. 참값과 근사값의 오차 설정값은 Excel 옵션 → 수식 → 계산 옵션 → 변화 한도값에서 확인할 수 있으며, 변경이 가능하다. 만일 참값과 근사값의 차이가 설정된 값보다 계속 큰 값이 나오면 해를 찾지 못하는

단점이 있으며, 해가 여러 개인 경우에는 시작값에서 수렴하는 하나의 값만 찾아준다.

또 다른 예로 두 함수  $f_1(x) = x^4$ ,  $f_2(x) = \sin(x)$ 의 해를 구해 보자. 그림

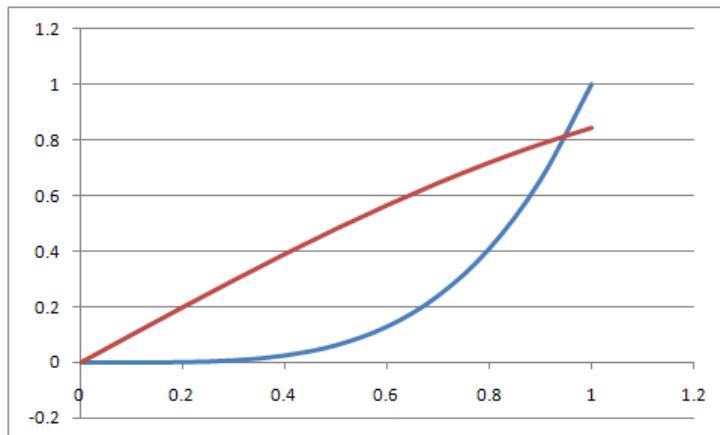


그림 6.5.4: 두 함수의 그래프

6.5.4에서 두 함수의 해가 두 개인 것을 확인할 수 있다. 목표값 찾기는 하나의 해만 찾아주기 때문에 이 두 함수의 해를 구하려면 목표값 찾기를 두 번 실행해야 한다.

- ◉ 그림 6.5.4에서 두 함수의 해는 0과 1 근처에 있다. 하나의 해는 0.1을 초기값으로 하고
- ◉ 또 다른 해는 초기값을 1로 시작하면

두 함수의 해를 구할 수 있다. 그림 6.5.5는 두 함수의 해를  $f_1(x) - f_2(x) = 0$ 인 경우에  $x$ 의 값을 계산하였다.

- ◉ 첫 번째 해는 두 함수의 차이가  $f_1(x) - f_2(x) = 4.8 \times 10^{-5}$ 일 때  $-4.8 \times 10^{-5}$ 이고
- ◉ 두 번째 해는 두 함수의 차이가  $f_1(x) - f_2(x) = 0.00082$ 일 때 0.949905이다.

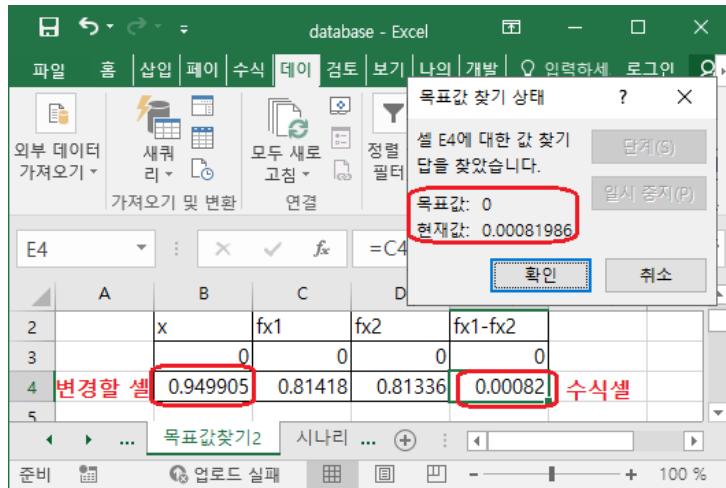


그림 6.5.5: 두 함수의 해를 구한 결과

## 제 6.6 절 해 찾기

해 찾기는 여러 개의 해를 한 번에 구할 때 사용하는 분석 도구이다.

우선 해 찾기는 기본 설정 메뉴에 포함되지 않아서 메뉴를 실행할 수 없다. 해 찾기 메뉴 추가는 파일 → 옵션 → 추가기능 → 관리 메뉴에서 해찾기 추가 기능을 추가한다.

예를 들어 어느 학교에서 이번 달 성적을 지난 달 성적보다 평균 5점이 향상되도록 하려면 각 학생들이 얻어야 되는 점수는 각각 얼마인지 생각해 보자. 그림 6.1(a)에 지난 달 평균 성적이 79.52점 일 때, 이번 달 전체 평균 성적이 85 점이 되려면 각 학생이 얻어야 되는 점수는 어떻게 되는지 계산해 보자. 그림 6.1(a)에 전체 평균은 각 셀을 반드시 참조하여 구한 값이어야 한다. 해 찾기 메뉴를 실행하면 그림 6.1(b)가 나타난다. 여기에

- ① 목표 셀에는 전체 평균이 있는 셀을 참조한다. 대상에는 목표값 85를 입력한다.
- ② 해의 조건에는

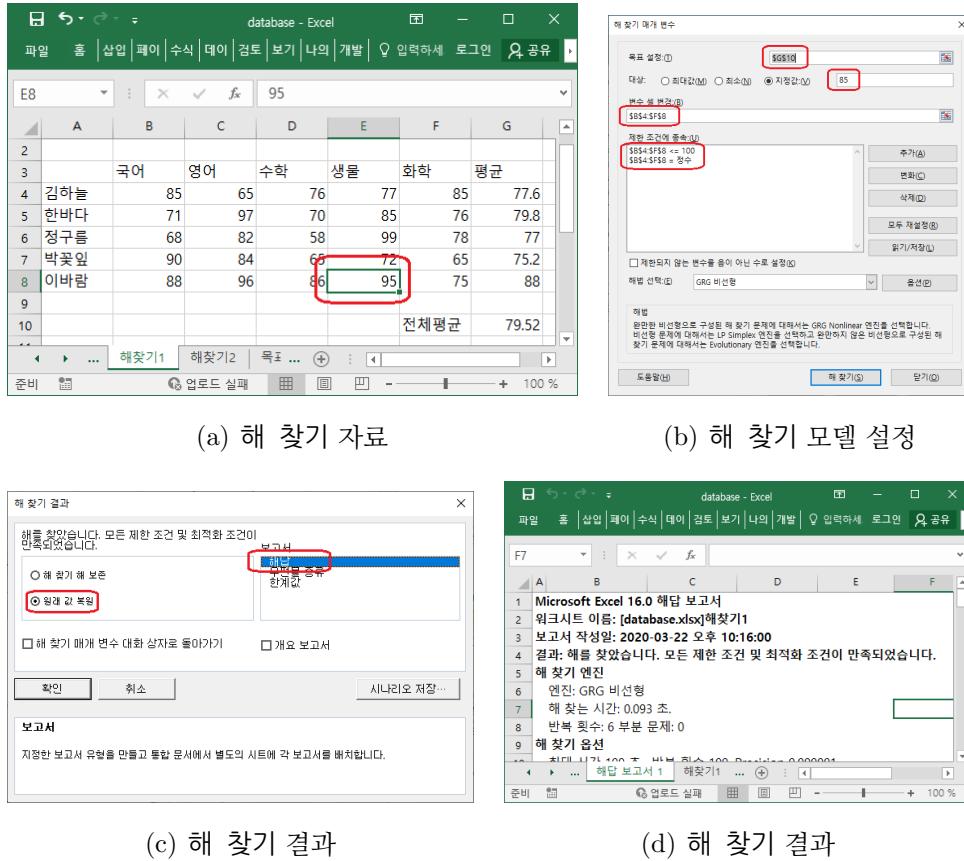


그림 6.6.1: 해 찾기

- ✖ **최대값** : 최대값을 찾는 경우
- ✖ **최소값** : 최소값을 찾는 경우
- ✖ **지정값** : 사용자가 지정한 값을 찾는 경우

를 선택하여 입력한다.

- ◉ **값을 바꿀 셀에는 목표 셀을 구하려면 변경할 셀을 입력하는 것으로 학생들의 지난 달 점수를 선택한다.** 이 값은 반드시 목표 셀과 참조 관계로 계산되어 있어야 한다.

- 제한 조건에는 값을 바꿀 셀의 조건으로 학생 성적은 100을 초과할 수 없기 때문에 100점 이하로 설정하였고, 성적이 실수인 경우는 학생이 취득할 수 없는 점수이므로 정수라는 조건을 설정하였다.
- 설정 종료 후 그림 6.1(b)에서 실행 버튼을 클릭한다. 그 결과 그림 6.1(c)에 해 찾기 결과가 나타난다.

✖ 해 찾기 해 보존은 원자료를 구한 해로 바꾸는 것이며

✖ 원래 값 복원은 초기값을 유지하고 구한 결과만 잠시 보며 확인 버튼을 누르면 구한 결과가 사라진다. 이 때 보고서 → 해답 → 확인 버튼을 누르면 해 찾기 결과가 다른 시트에 나타난다.

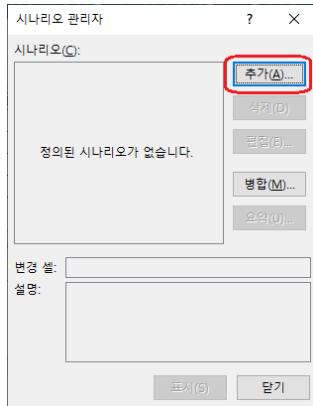
그림 6.1(d)에 해 찾기 결과가 해답보고서 1 시트에 보인다.

	A	B	C	D	E	F	G
1	지점명	할인율	단가	판매수량	할인단가	할인가격	
2	강동		0.3	3000	10	2100	21000
3	강서		0.3	2800	15	1960	29400
4	강남		0.3	3200	10	2240	22400
5	강북		0.3	3500	17	2450	41650
6							

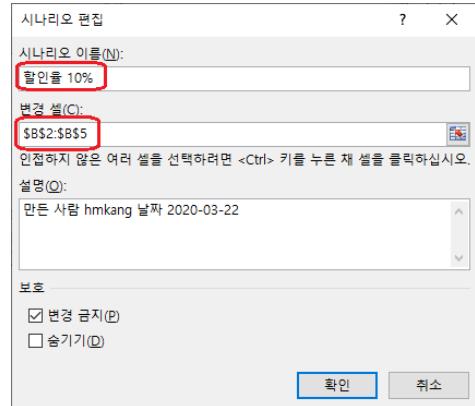
그림 6.6.2: 시나리오 자료

## 제 6.7 절 시나리오 작성

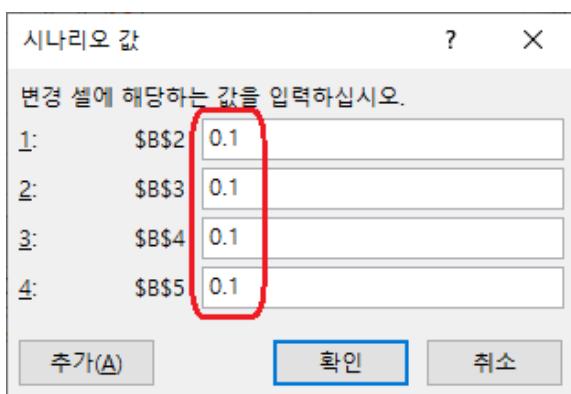
시나리오는 입력된 값과 그 값을 변경하였을 때의 결과를 비교할 때 사용할 수 있다.



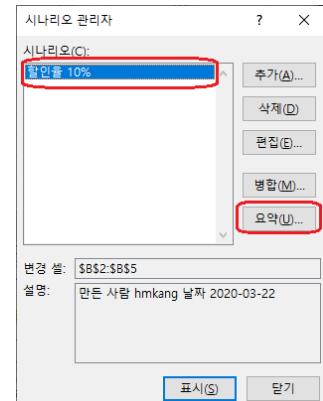
(a) 시나리오 관리자



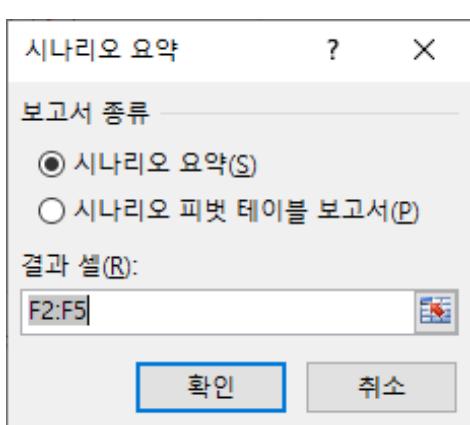
(b) 시나리오 추가



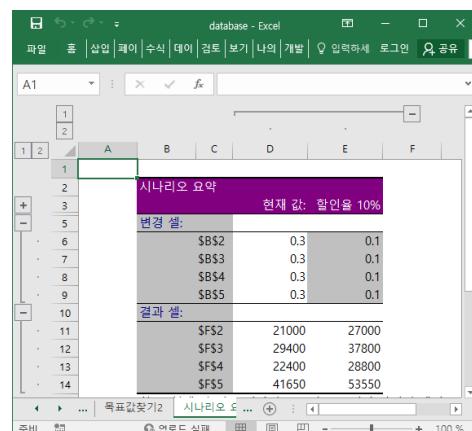
(c) 시나리오 값



(d) 시나리오 요약



(e) 시나리오 요약



(f) 시나리오 요약 시트

예를 들어 어느 회사의 각 지역지점에서 판매하는 물건 가격을 할인하려고 한다. 이 때 할인율이 10%, 20%, 30% 일 때 할인가격이 얼마인지 알아보려고 할 때 시나리오를 사용하면 각 할인율에 대한 할인가격을 알아볼 수 있다. 그림 6.6.2 자료는 어느 회사의 할인율이 30%일 때 판매가격이다. 할인율이 10%와 20%일 경우에는 물건의 판매가격이 어떻게 되는지 알아보자. 시나리오는 데이터→데이터 도구→가상 분석→시나리오 관리자 메뉴를 클릭으로 실행한다. 그림 6.3(a)가 시나리오를 실행하기 위한 시나리오 관리자 창이다. 이 창에서 추가 버튼을 클릭하면 그림 6.3(b)의 시나리오 추가 창이 나타난다. 이 창에

- 시나리오 이름 : 시나리오 이름 입력. 할인율 10%를 입력
- 변경 셀 : 수식에 영향을 줄 입력 값으로 할인율 0.1를 입력
- 설명 : 시나리오에 대한 설명으로 컴퓨터에 등록된 만든 사람과 날짜가 기본값으로 설정 됨

모든 값을 입력하고 확인 버튼을 클릭한다. 그 결과 그림 6.3(c)의 시나리오 값 창이 나타나고 변경될 셀에 값을 입력한다. 그림 6.3(c)는 모든 변경셀의 할인율을 0.1로 입력하였다. 그림 6.3(d)는 시나리오 관리자 창에 추가된 시나리오를 확인할 수 있다. 또 다른 시나리오 추가는 그림 6.3(d)에서 추가 버튼을 클릭하고 그림 6.3(b)부터 그림 6.3(c)까지 과정을 반복한다. 시나리오 설정이 끝났으면 요약 버튼을 누른다. 그림 6.3(e)는 시나리오 요약 창으로 보고서 종류는

- 시나리오 요약
- 시나리오 피벗 테이블 보고서

중에서 출력할 것을 선택한다. 그림 6.3(f)에 시나리오 요약 결과를 확인할 수 있다.

## 제 II 편

# Visial Basic for Applications



## 제 7 장

### 매크로(Macro)

일상 생활에서 우리는 같은 내용을 많이 반복한다. 예를 들어 매월 지급하는 월급, 매월 은행에 지급하는 적금, 퇴직자들이 지급받는 연금은 달마다 계산식을 반복계산한다. 또한 엑셀에서 작업한 여러 개 표를 하나씩 복사하여 워드프로세서에 붙여넣기도 반복작업이다. 매크로(Macro)는 작업할 내용을 미리 만들었다가 한 번에 실행하는 도구이다. 이후에도 사용한 매크로는 반복 사용이 가능하다. 좋은 점이 있으면 나쁜 점도 있다. 매크로는 비주얼베이직 프로그래밍 언어로 만든다. 비주얼베이직 사용법을 알아야 매크로 사용이 가능하다.

매크로는 마이크로소프트사에서 만든 비주얼 베이직(Visual Basic) 프로그래밍을 작성하여 MS Office 포함된 제품 엑셀, 워드, 파워포인트, 엑스스와 같은 응용프로그램(Application)과 연결한다. 이런 MS Office 기능을 VBA로 부른다. VBA는 Visual Basic for Applications로 엑셀, 워드, 파워포인트, 엑스스가 Applications이다. 비주얼 베이직은 객체지향언어이다. 객체지향언어(Object Oriented Programming)는 만들어진 객체(object)를 사용하는 4 세대 프로그래밍 언어이다. 객체는 이미 만들어진 프로그램이다. 3세대 프로그래밍 언어까지는 필요한 내용은 사용자가 대부분 만들지만 객체지향언어는 필요한 개체를 이미 만들어진 프로그램에서 찾아 사용한다.

## 제 7.1 절 매크로 기록과 실행

매크로는 매크로 기록으로 만들고 이미 만들어진 매크로는 매크로 실행으로 실행한다.

### 7.1.1 매크로 기록

매크로 기록은 엑셀에서 사용자가 여러 명령을 실행할 때 각 단계 정보를 저장한다. 이후 저장된 매크로는 사용자가 필요할 때 사용한다.

매크로 기록은

- ① 개발 도구→코드→매크로 기록 () 메뉴를 클릭
- ② 보기→매크로→매크로 기록 메뉴를 클릭
- ③ 개발 도구→코드→Visual Basic 메뉴를 클릭하여 코드를 직접입력

한다.

개발도구 메뉴는 기본값이 메뉴바에 없으니 추가시킨다. 메뉴 추가는 파일 → 옵션 → 리본 사용자 지정 → 추가 기능 메뉴를 선택한다(그림 7.1.1). 추가한 개발 도구 메뉴는 엑셀 메뉴바에서 즉시 확인된다.

작성한 매크로는 Excel 매크로 사용 통합 문서로 저장한다(그림 7.1.2). Excel 매크로 사용 통합 문서 파일 확장자는 .xlsm이다.

매크로 관련 메뉴는 개발도구→코드에 있다(그림 7.1.3). 매크로 기록은

1. 그림 7.1.3에서 매크로 기록을 누르면 그림 7.1.4 창이 나온다.
2. 그림 7.1.4에 매크로 이름을 입력한다. 매크로 이름에서 첫 글자는 반드시 문자이고 이후 나머지는 문자, 숫자, 밑줄(\_)을 사용할 수 있지만 공백은 사용할 수 없다. 매크로 이름은 Visual Basic 식별자 규칙과 동일하다 (11.3절 참조). 매크로 이름이 바로 식별자이다.
3. 매크로는 바로가기 키로 실행할 수 있다. 바로가기 키 설정은 그림 7.1.4에 문자를 입력한다. 바로가기 키는

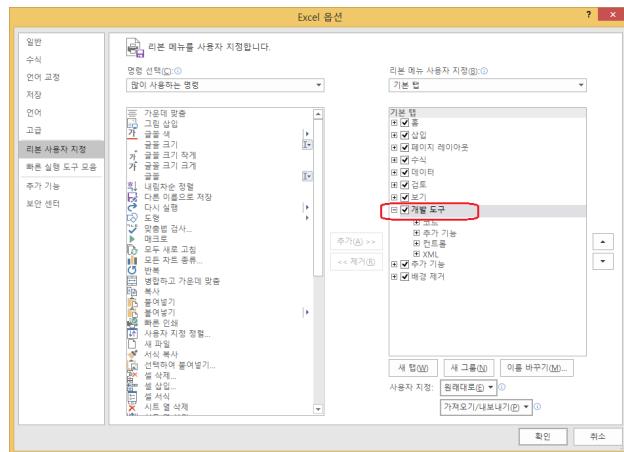


그림 7.1.1: 개발도구 메뉴 추가

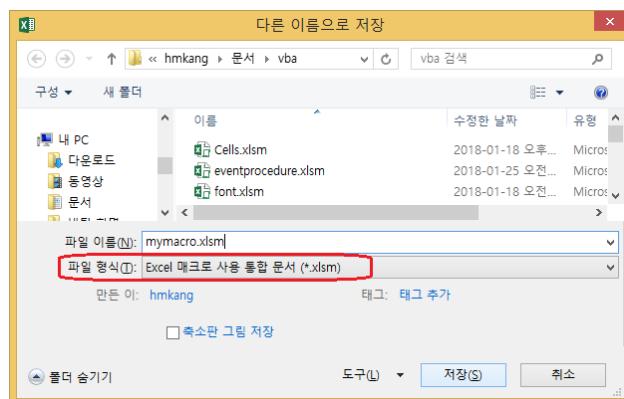


그림 7.1.2: 매크로 파일 저장

① Ctrl+문자나

② Ctrl+Shift+문자를

사용할 수 있다. 바로가기에 사용 가능한 문자는 키보드에 있는 임시 문  
자키로 0나 # 같은 특수 문자나 숫자는 사용할 수 없다. 만일 매크로가  
포함된 엑셀 파일이 열려 있을 때 매크로 실행 바로가기 키는 윈도에



그림 7.1.3: 매크로 관련메뉴

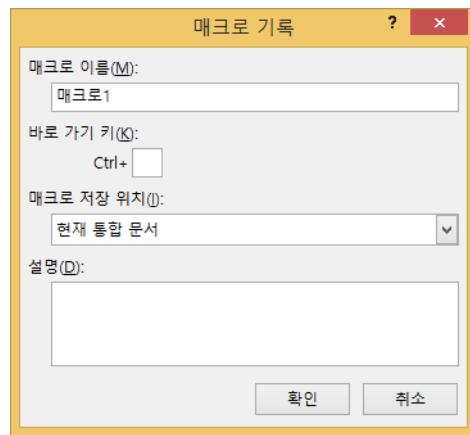


그림 7.1.4: 매크로 기록

기본값으로 설정된 바로가기 키보다 우선한다.

#### 4. 기록된 매크로 저장 방법은

- ◉ 개인용 매크로 통합문서
- ◉ 새 통합 문서
- ◉ 현재 통합 문서

가 있다. 사용자가 엑셀에서 매크로를 사용하려면 매크로 저장 위치를 개인용 매크로 통합문서에 저장한다(그림 7.5(a)). 이후 엑셀파일을 실행하고 단축 키를 누르면 개인용 매크로 통합문서에 저장된 매크로가

실행된다. 이 통합문서는 엑셀이 시작될 때 자주 사용할 매크로를 넣어 놓는 파일이다.

5. 매크로 기록은 그림 7.5(a)에서 확인 버튼을 클릭한 다음부터 시작한다. 매크로 기록 과정에서 선택한 셀은 상대 참조로 기록한다. 매크로 기록하기 전 현재 셀은 매크로에서 인식하지 못한다. 따라서 매크로를 시작하고 선택한 셀이 매크로를 기록하는데 기준 셀이다. 그림 7.5(c)는 그림 7.5(a)에서 매크로 저장위치를 개인용 매크로 통합문서로 한 결과 PERSONAL.XLSB 파일이 생겼다.
6. 함수 사용, 글꼴 설정, 정렬같은 매크로에서 저장할 동작을 수행한다.
7. 매크로 기록 종료는 기록 중지()를 클릭한다. 기록 중지 버튼은 그림 7.1.3에서 매크로 기록을 눌렀을 때 나타난다.

만든 매크로는 VBA 코드로 저장되며 Visual Basic Editor로 코드 확인과 편집이 가능하다. 다음은 각 사칙연산에 대한 매크로이다. 먼저 덧셈에 대한 매크로를 만들어 보자.

1. 그림 7.1.3에서 매크로 기록을 누르면 그림 7.1.4 창이 나타난다. 7.1.4 창에 매크로 이름은 덧셈, 바로가기 키는 a로 하고 확인 버튼을 누르면 덧셈 매크로 기록이 시작된다(그림 7.5(a)).
2. 덧셈 설정은 D3 셀에 =B3+C3를 입력하고 매크로 중지 버튼을 누른다(그림 7.5(b)). 이때 주의사항으로 중지 버튼을 누르기 전 현재 셀 위치는 D3 으로 설정해야 매크로가 정상적으로 작동한다.
3. 덧셈 매크로를 만든 동일한 셀에 뺄셈, 곱셈, 나눗셈에 대한 매크로를 만든다.
4. 이렇게 만든 4개의 매크로는 Visual Basic Editor에서 프로젝트 탐색기 → 모듈 → Module1에서 확인한다(그림 7.5(c)).

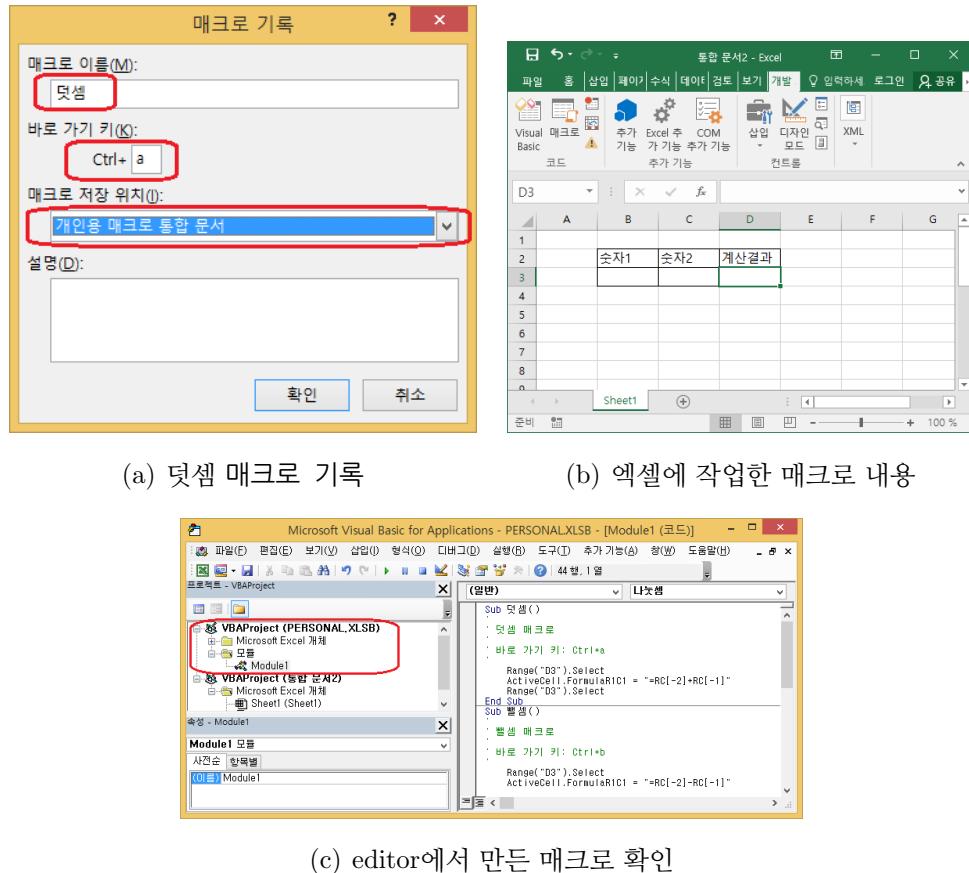


그림 7.1.5: 명령 매크로 기록 과정

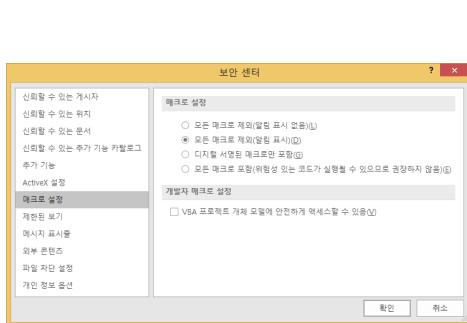
### 7.1.2 매크로 실행

매크로 실행에서 프로그래머가 악의적인 코드를 만들었다면 실행하는 컴퓨터에 치명적인 영향을 줄 수 있다. 그래서 MS Office는 매크로 실행 방법을 4 가지 제공한다. 매크로 실행의 기본 설정은 모든 매크로 제외(알림 표시)이므로 매크로가 포함된 파일을 열어도 실행되지 않는다. 엑셀에서 매크로 설정은 다음과 같다.

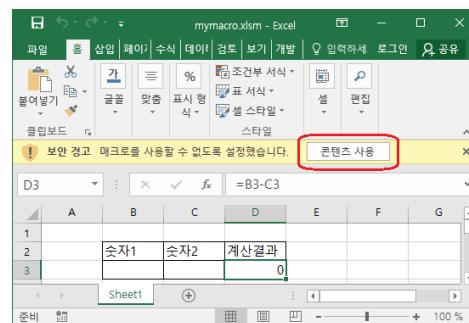
- 개발 도구 → 코드 → 매크로 보안 ( ) 메뉴 실행

① 설정 종류

- ✖ 모든 매크로 제외(알림 표시 없음) : 매크로는 실행할 수 없음
- ✖ 모든 매크로 제외(알림 표시) : 그림 7.6(b)에 매크로가 실행되도록 설정 변경
- ✖ 디지털 서명된 매크로만 포함 : 공인인증서로 매크로를 서명한 경우
- ✖ 모든 매크로 포함(위험성 있는 코드가 실행될 수 있으므로 권장하지 않음)



(a) 보안 센터



(b) 보안 경고

그림 7.1.6: 매크로 설정(보안)

### 매크로 실행 방법은

- ① 매크로 기록할 때 설정한 바로 가기 키를 누르거나(그림 7.5(a))
- ② 매크로가 연결된 단추 양식 컨트롤을 클릭한다. 단추에 매크로 연결은 개발도구→컨트롤→삽입→양식 컨트롤에서 단추 (■) 를 클릭하고 마우스 드래그로 단추가 추가할 영역을 만든다. 단추 추가와 동시에 그림 7.8(a) 매크로 지정 창이 나타난다. 목록에서 추가할 매크로를 선택한다. 이후 매크로 수정이나 변경은 추가한 단추에서 마우스 오른쪽 버튼을 누르면 보이는 메뉴 중 매크로 지정을 선택한다. 매크로 지정 창에서 그림

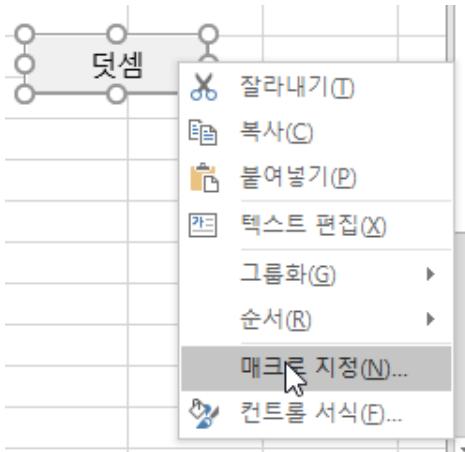


그림 7.1.7: 매크로 지정

7.8(a)와 같이 연결하려는 목록을 선택하면 그림 7.8(b) 단추에 매크로가 연결된다. 그리고 이 단추를 누르면 연결한 매크로가 실행된다.

- ◉ 개발도구→코드→매크로 보기 메뉴를 실행하고 목록에서 실행하려는 매크로를 선택한다.
- ◉ 보기→매크로→매크로를 선택하고 목록에서 실행하려는 매크로를 누른다.

사칙연산에 대한 매크로를 작성해보자. B3과 C3에 숫자를 입력하고 D3셀에 계산 결과가 출력된다. 그림 7.5(b)는 사칙연산 매크로 4 개를 만들고 단추에 각 매크로를 연결하였다. 바로가기 키는 덧셈은 a, 뺄셈은 b, 곱셈은 c, 나눗셈은 d로 하였다. 사칙연산 매크로를 다음과 같이 실행해 보자.

- ◉ 덧셈에 대한 바로 가기 키  $Ctrl + a$ , 뺄셈에 대한 바로 가기 키  $Ctrl + b$ , 곱셈에 대한 바로 가기 키  $Ctrl + c$ , 나눗셈에 대한 바로 가기 키  $Ctrl + d$ 를 차례대로 눌러 매크로를 실행한다.
- ◉ 개발도구 → 코드 → 매크로를 실행하여 기록된 매크로 덧셈, 뺄셈, 곱셈, 나눗셈 중 원하는 목록을 선택하여 실행한다.

- ⑥ 개발도구→삽입→양식 컨트롤에서 단추 버튼을 선택한다. 그리고 셀에 추가될 단추 크기를 마우스 드래그로 설정한다. 단추와 매크로 연결은 단추에 마우스로 옮기고 오른쪽 버튼을 누른다. 그리고 목록에서 연결하려는 매크로를 선택한다(그림 7.8(a)).

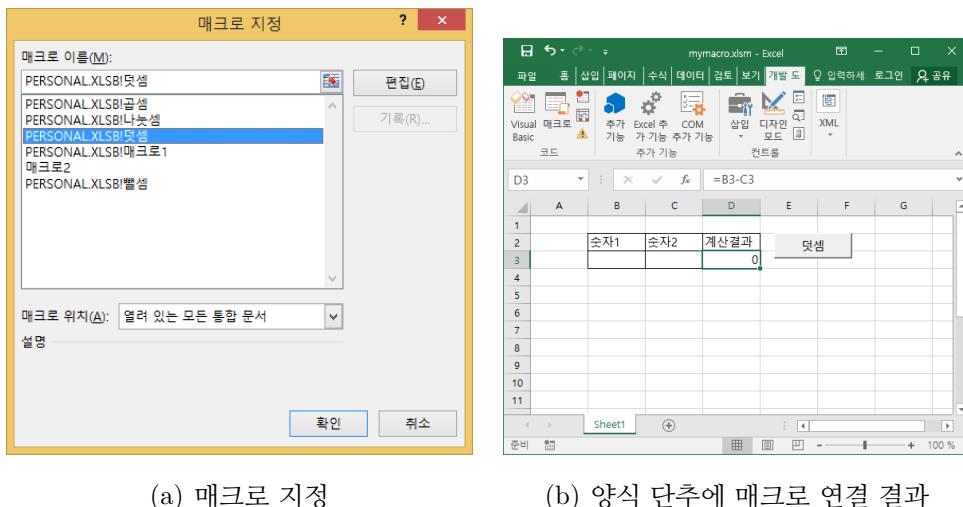


그림 7.1.8: 양식 단추에 매크로 연결

## 제 7.2 절 VBA

매크로와 VBA는 동일한 것이며 엑셀에서 마우스를 이용하여 실행과정을 기록한 것을 매크로라 부르고 Visual Basic Editor를 이용하여 코드를 직접 입력하면 VBA라 부른다. VBA(Visual Basic for Applications)는 비주얼 베이직 언어를 사용하여 MS Office 속하는 제품들(엑셀, 워드, 파워포인트, 엑서스 등)과 연동하는 것을 말한다. 우리는 엑셀과 연동하기에 응용프로그램(Application)은 엑셀이다. 비주얼 베이직 기초문법은 예전의 베이직의 문법을 토대로 발전하였다. 그러나 비주얼 베이직은 GUI(Graphic User Interface) 환경에서 작성하기 때문에 예전에 사용하던 베이직보다 사용하기가 쉽고 객체지향언어

로 되어서 이전 베이직과 차이점이 많다. 그림 10.1.1은 그림 7.1.3에서 Visual Basic Editor를 실행한 것이다.

## 연습문제

1. 다음 조건에 맞게 매크로를 작성하고 실행한다.
  - ① 개발도구→코드에서 매크로 기록을 누른다. 매크로 이름은 급수구하기고 바로 가기 키는 Ctrl+Shift+E이다.
  - ② A1 셀에 2를 입력하고 엔터키를 누른다. 다시 A1 셀로 이동한다. 현재 셀 왼쪽 하단에 있는 채우기 핸들에서 마우스 오른쪽 버튼을 누르고 9 칸 아래로 이동한다.
  - ③ 나타난 연속데이터 창에 유형에 급수, 단계값은 2를 입력하고 확인 버튼을 누른다.
  - ④ A1 셀로 이동한다.
  - ⑤ 기록중지 버튼을 누른다.
  - ⑥ 시트 Sheet1에 양식 컨트롤 단추를 추가한다. 이 단추의 텍스트는 2의 급수 구하기이다.
  - ⑦ 이 단추를 누르면 급수구하기 매크로가 실행되게 연결한다.

# 제 8 장

## 컨트롤

컨트롤은 엑셀에서 시트나 유저폼에 추가하는 GUI이다. 컨트롤 기능은 각 컨트롤 이벤트가 발생할 때 실행 코드를 사용자가 입력할 수 있다. 컨트롤 종류는 엑셀에 양식 컨트롤과 ActiveX 컨트롤이 있고 VBA 유저폼에 컨트롤 도구상자가 있다. 모든 컨트롤은 VBA 개체이고 속성과 메서드를 변경하여 사용한다.

### 제 8.1 절 ActiveX 컨트롤

ActiveX 컨트롤은 시트에 컨트롤을 추가하고 컨트롤 이벤트 프로시저에 실행할 코드를 입력한다.

#### 8.1.1 명령 단추(CommandButton)

명령 단추는 흔히 말하는 버튼이다. 명령 단추의 개체 속성에 대하여 알아보자.

- ① Accelerator 명령 단추를 실행하는 단축키 설정이다. 설정한 문자가 버튼에 있으면 밑줄이 생긴다. 실행은 Alt + 설정문자를 누른다.
- ② AutoLoad 명령 단추는 엑셀 파일이 시작되면 자동으로 로딩되도록 설정.

- AutoSize 명령한 글자 크기에 맞게 명령 단추의 크기를 자동으로 변경 할지 설정
- BackColor 명령 단추 배경색을 설정
- BackStyle 명령 단추 스타일로 기본은 불투명이고 투명으로 설정
- Caption 명령 단추에 나타날 문자 입력
- Enabled 명령 단추가 편집 상태일 때 개체를 사용할지 설정
- Font 글자체, 글자 크기, 글자 굵기 설정
- ForeColor 명령 단추 전경색 설정
- Height 명령 단추의 높이
- Left 명령 단추 위치가 시트 왼쪽에서 떨어져 있는 거리 설정
- Locked 명령 단추가 이동하거나 크기 변경의 가능과 불가능을 설정
- MouseIcon 그림으로 마우스 아이콘 설정
- MousePointer 마우스가 명령 단추 위에 놓였을 때 마우스 포인터 모양 설정
- Picture 명령 단추 위에 추가할 그림 설정
- PicturePosition 추가한 그림 위치 설정
- Placement 명령 단추가 시트에 놓이는 것으로 셀 크기를 변경이나 셀 감추기와 같이 셀을 변경하면 1은 셀과 상관없이 모양과 크기가 변함이 없고, 2는 셀과 함께 움직이며, 3은 셀과 함께 움직이며 크기도 변한다.
- PrintObject 명령 단추를 인쇄할지 설정
- Shadow 명령 단추의 그림자 모양을 추가할지 설정

- ① TakeFocusOnClick 명령 단추 클릭하고 명령 단추를 활성화 상태로 할지 설정
- ② Top 명령 단추가 위로부터 거리 설정
- ③ Visible 명령 단추가 보이도록 설정.
- ④ Width 명령 단추의 너비 설정
- ⑤ WordWrap 명령 단추에 입력한 글자가 폭을 넘어가면 줄을 바꿀 것인지 설정

워크시트에 명령 단추를 추가하고 이벤트가 발생하면 실행할 코드를 만들어 보자.

1. 그림 8.1.1에서 개발도구→ActiveX 컨트롤에서 명령 단추를 누른다.

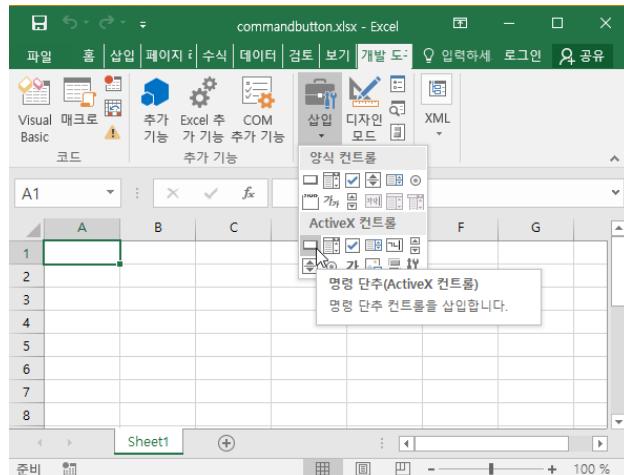


그림 8.1.1: 명령 단추1

2. 그림 8.1.2에서 명령 단추가 놓일 위치를 마우스로 드래그한다.

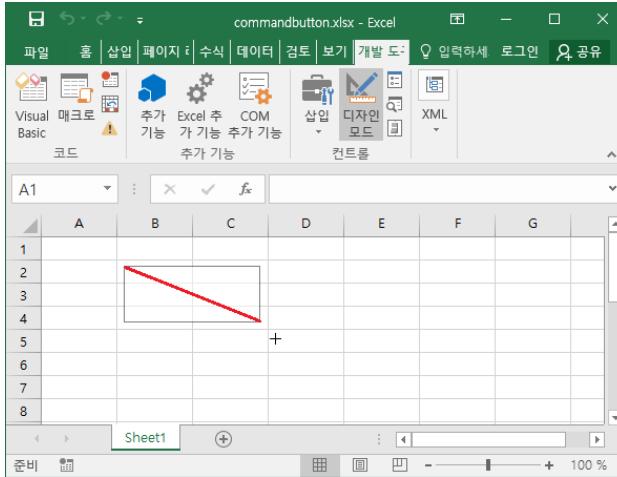


그림 8.1.2: 명령 단추2

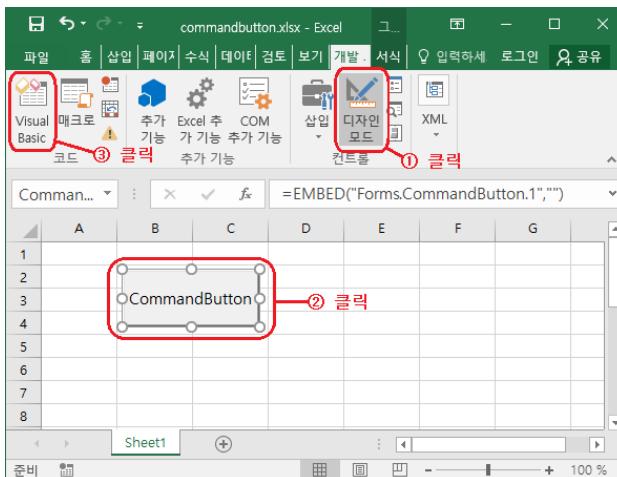


그림 8.1.3: 명령 단추3

3. 그림 8.1.3에서 컨트롤→디자인 모드를 클릭한다. Visual Basic을 클릭하면 에디터가 열린다.
4. 그림 8.1.4 에디터에서 Sheet1을 더블클릭한다. 개체 드랍다운 버튼 ②를

누르고 명령 단추 CommandButton1 ③을 클릭한다.

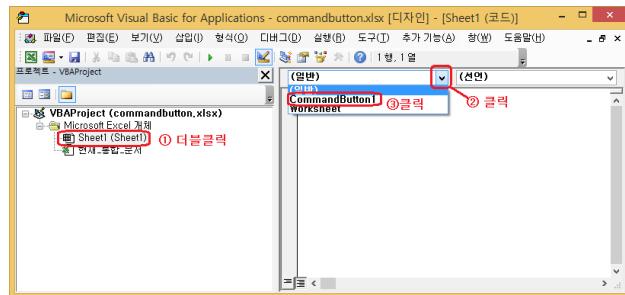


그림 8.1.4: 명령 단추4

5. 그림 8.1.5 이벤트 프로시저가 자동으로 만들어진다. 이벤트 드랍다운 버튼을 누르면 다른 이벤트 프로시저가 보인다.

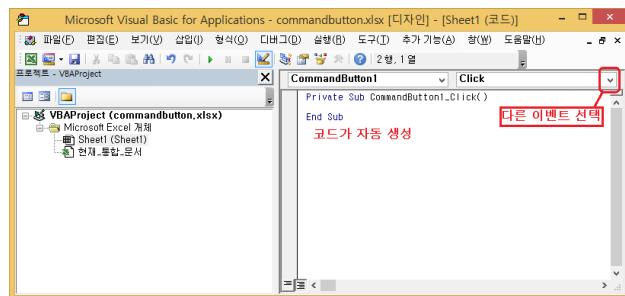


그림 8.1.5: 명령 단추5

6. 그림 8.1.6 이벤트 프로시저에 직접 실행문을 삽입한다. 보기 8.1.1에서 2 줄은 명령 단추에 적힌 문자를 B5 셀에 출력한다. Commandbutton1 개체 Caption 속성은 명령 단추에 쓰인 문자이다.

### 보기 8.1.1. [명령 단추 이벤트]

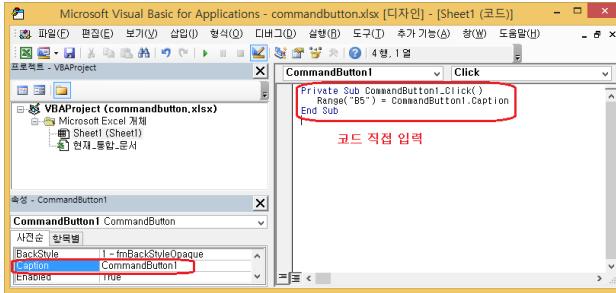


그림 8.1.6: 명령 단추6

```

1 Private Sub CommandButton1_Click()
2     Range("B5") = CommandButton1.Caption
3 End Sub

```

7. 그림 8.1.7 엑셀 창을 클릭하고 디자인 모드를 클릭하여 디자인 모드를 해제한다.

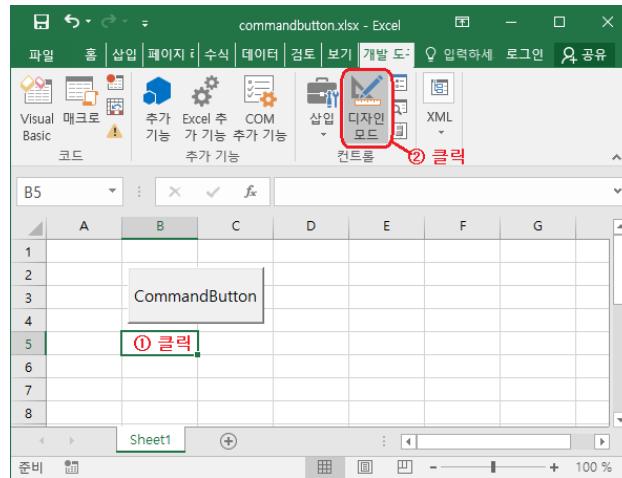


그림 8.1.7: 명령 단추7

8. 그림 8.1.8 엑셀 창에서 명령 단추를 클릭하면 이벤트 프로시저에 입력한 문이 실행된다.

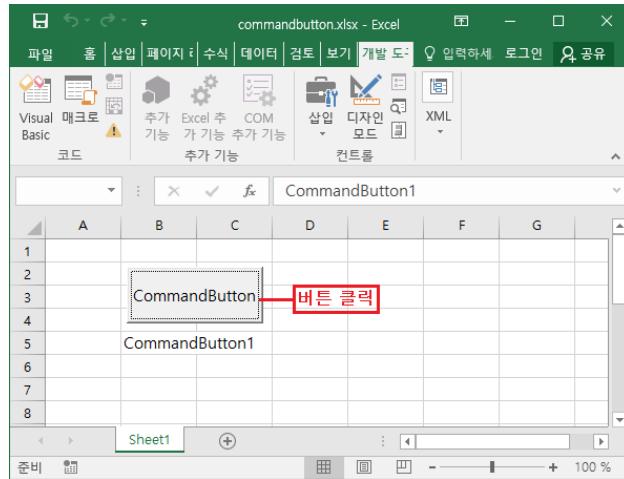


그림 8.1.8: 명령 단추8

명령 단추 이벤트 프로시저에 대하여 알아보자.

- ① BeforeDragOver 사용자가 마우스 드래그 앤 드롭하면 발생하는 이벤트
- ② BeforeDropOrPaste 사용자가 마우스를 떼고 붙여넣기하면 발생하는 이벤트
- ③ Click 명령 단추를 클릭하면 발생하는 이벤트
- ④ DblClick 명령 단추를 더블 클릭하면 발생하는 이벤트
- ⑤ Error 명령 단추 설정에서 오류가 발생하면 발생하는 이벤트
- ⑥ GetFocus 명령 단추를 선택하여 활성화되면 발생하는 이벤트
- ⑦ Keydown 명령 단추가 선택된 상태에서 키보드를 누르면 발생하는 이벤트

- ◉ KeyPress 명령 단추가 선택된 상태에서 키보드를 누르면 발생하는 이벤트
- ◉ KeyUp 명령 단추가 선택된 상태에서 키보드를 떼면 발생하는 이벤트
- ◉ LostFocus 명령 단추가 비활성화되어 포커스를 잃으면 발생하는 이벤트
- ◉ MouseDown 명령 단추를 누르면 발생하는 이벤트
- ◉MouseMove 명령 단추 위에서 마우스가 움직이면 발생하는 이벤트
- ◉ MouseUp 눌려있던 명령 단추가 위로 오르면 발생하는 이벤트

### 8.1.2 콤보 상자(ComboBox)

콤보 상자는 추가한 목록에서 한 목록을 선택하는 컨트롤이다. 다음은 콤보 상자를 엑셀 시트에 추가하고 목록을 선택하면 이벤트가 발생할 때 실행하는 코드를 만든다.

1. 그림 8.1.9에서 개발도구→ActiveX 컨트롤에서 콤보 상자를 누른다.

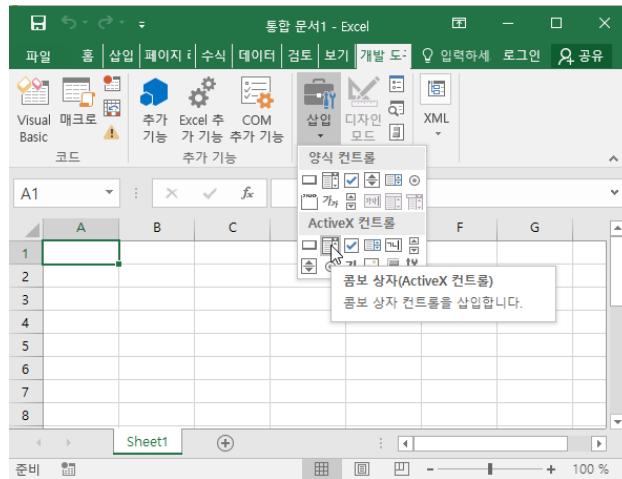


그림 8.1.9: 콤보 상자1

2. 그림 8.1.10 시트에 마우스 드래그로 콤보 상자 ①을 추가한다. 추가한 콤보 상자는 디자인 모드 ②를 클릭하고 Visual Basic ③을 클릭해야 콤보 상자의 속성이 ③에 나타난다.

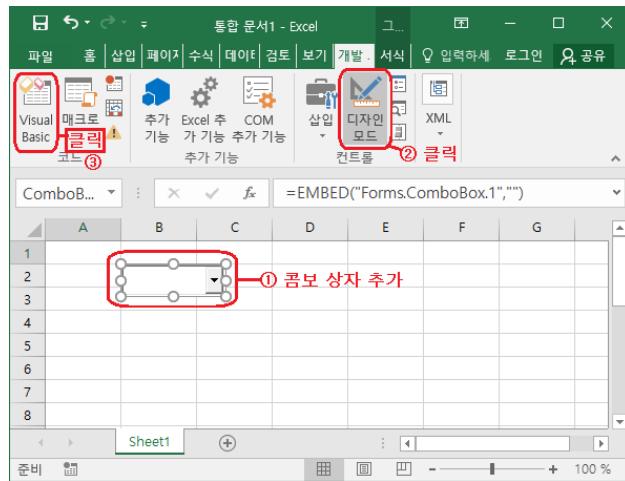


그림 8.1.10: 콤보 상자2

3. 그림 8.1.11의 프로젝트 창에서 Sheet1 (Sheet1) ①을 더블 클릭한다. 개체는 WorkSheet ②를 선택한다. 이벤트는 Activate ③을 선택하고 직접 코드를 입력한다. 코드에서 ComboBox1 개체는 속성 창 (이름)에 있는 값과 같게 입력한다. 코드 ④ 내용은 목록에 강원도, 경기도, 서울시를 추가한다. Visual Basic 에디터에서 엑셀로 이동하려면 엑셀로 돌아가기 ⑤를 클릭한다
4. 그림 8.1.12에서 디자인 모드 ①을 눌러 디자인 모드를 해제한다. 콤보 상자 목록은 드랍다운 버튼을 클릭해도 없다. 목록은 Activate 이벤트가 발생해야 추가되므로 아직 목록이 없는 것이다.
5. 그림 8.1.13에 시트 Sheet2를 추가하고 이동한다.
6. WorkSheet 개체 Activate 이벤트를 발생하려면 그림 8.1.14에서 시트 엑셀 VBA

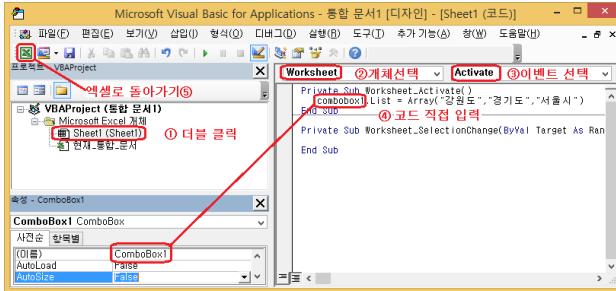


그림 8.1.11: 콤보 상자3

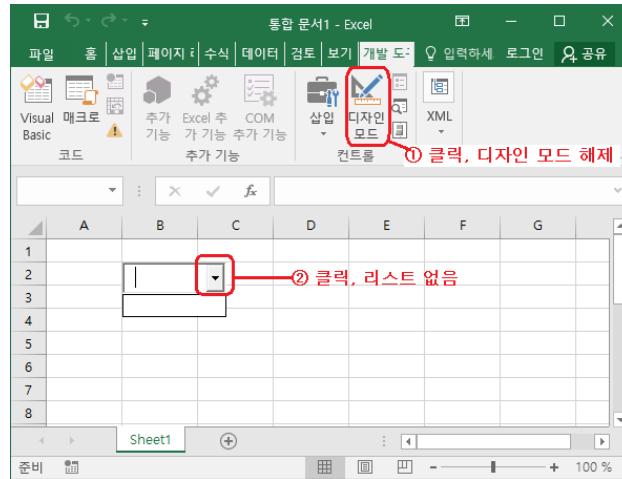


그림 8.1.12: 콤보 상자4

Sheet1 ①을 클릭한다. 콤보 상자에서 목록 확인은 드랍다운 ②를 클릭 한다. 다른 이벤트를 추가하고 그 안에 실행 코드 작성은 디자인 모드 ③과 Visual Basic ④를 클릭한다.

7. 그림 8.1.15에서 ComboBox1 개체 ①과 Change 이벤트 ②를 선택한다. Change 프로시저는 콤보 상자 목록을 변경하면 발생하는 이벤트이다. 이벤트가 발생하면 실행 코드 ③은 A5 셀에 변경한 목록을 출력한다. 보기 8.1.2에서 2 줄은 목록이 변경되면 목록을 A5 셀에 저장한다. List

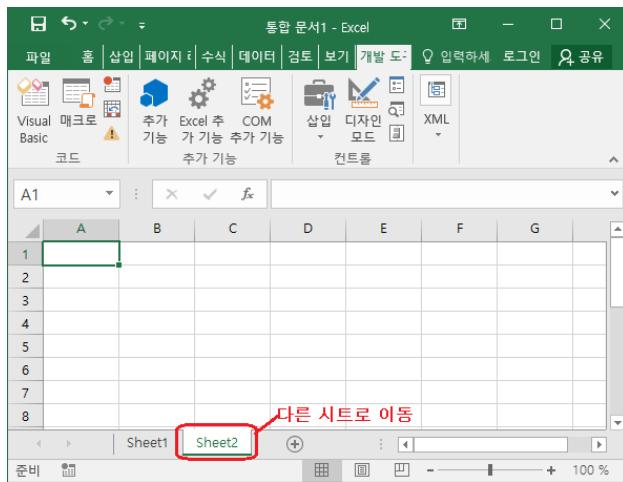


그림 8.1.13: 콤보 상자5

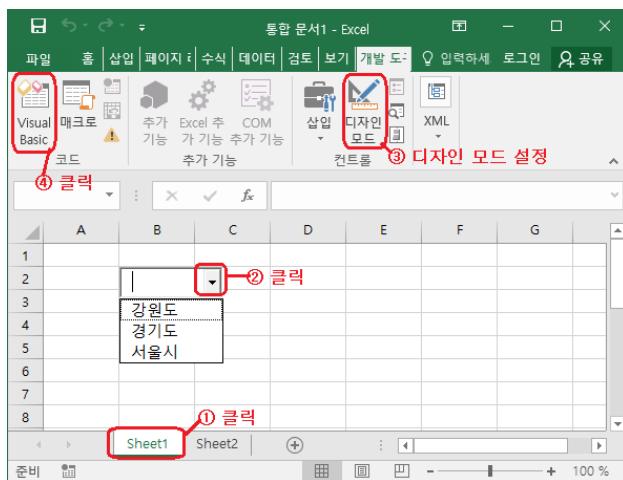


그림 8.1.14: 콤보 상자6

속성은 설정한 목록이고 ListIndex 속성은 설정한 목록을 0부터 1씩 증가하는 인덱스 값이다. 예를 들어 강원도 목록은 ListIndex 값이 0이고 List(0)은 목록에서 강원도이다. 6 줄은 ComboBox1 개체에 목록을 추가한다. 추가하는 목록은 여러 개이므로 배열 Array 함수를 이용한다.

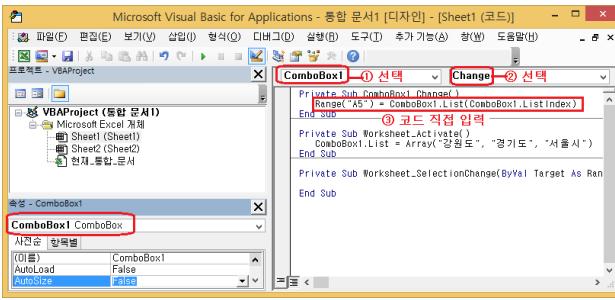


그림 8.1.15: 콤보 상자7

### 보기 8.1.2. [콤보 상자 이벤트]

```

1  Private Sub ComboBox1_Change()
2      Range("A5") = ComboBox1.List(ComboBox1.ListIndex)
3  End Sub
4
5  Private Sub Worksheet_Activate()
6      ComboBox1.List = Array("강원도", "경기도", "서울시")
7  End Sub
8
9  Private Sub Worksheet_SelectionChange(ByVal Target As Range)
10
11 End Sub

```

8. 그림 8.1.16에서 디자인 모드 ①을 해제하고 콤보 상자 ②에서 강원도를 선택하면 A5 셀에 선택한 목록 강원도가 출력된다.
9. 그림 8.1.17에서 문서는 비주얼 베이직 코드가 삽입되어 있으니 파일 형식을 Excel 매크로 사용 통합 문서로 저장한다.

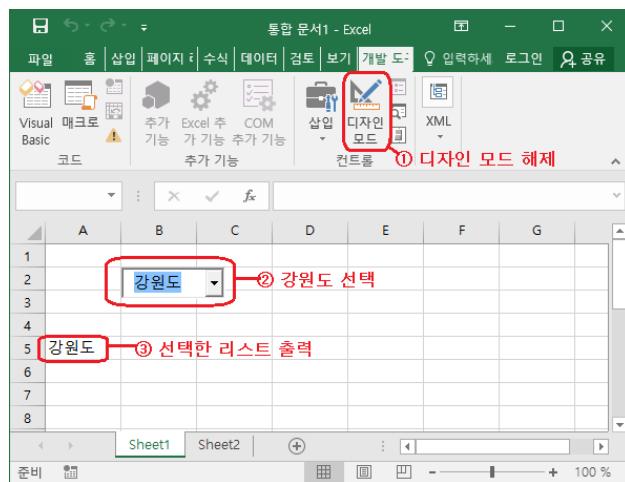


그림 8.1.16: 콤보 상자8

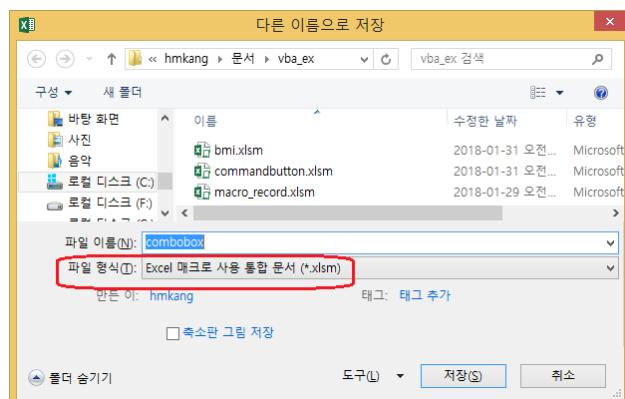


그림 8.1.17: 콤보 상자9

### 8.1.3 확인란(CheckBox)

확인란은 추가한 확인 목록에서 여러 목록을 선택하는 컨트롤이다. 다음은 확인란을 엑셀 시트에 추가하고 목록을 선택하면 이벤트가 발생할 때 실행하는 코드를 만든다.

- 그림 8.1.18에서 개발도구→ActiveX 컨트롤에서 확인란을 누른다.

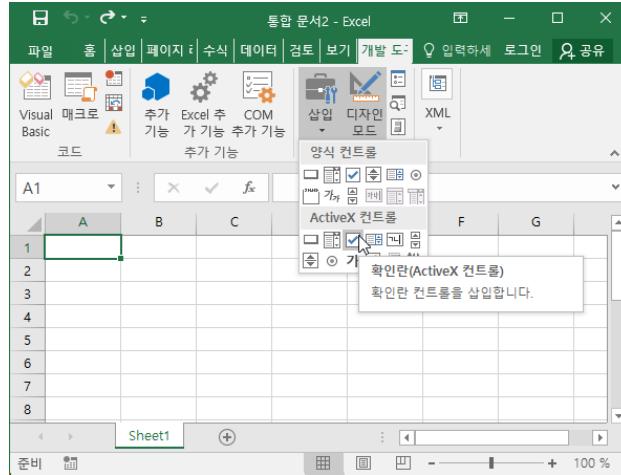


그림 8.1.18: 확인란1

2. 그림 8.1.19에서 마우스 드래그로 확인란이 추가될 영역 ①을 선택한다. 확인란의 속성은 ②를 클릭한다. 확인란 속성에서 Caption 속성값을 강원도로 변경한다. Caption 속성은 목록에서 보여지는 문자이다.

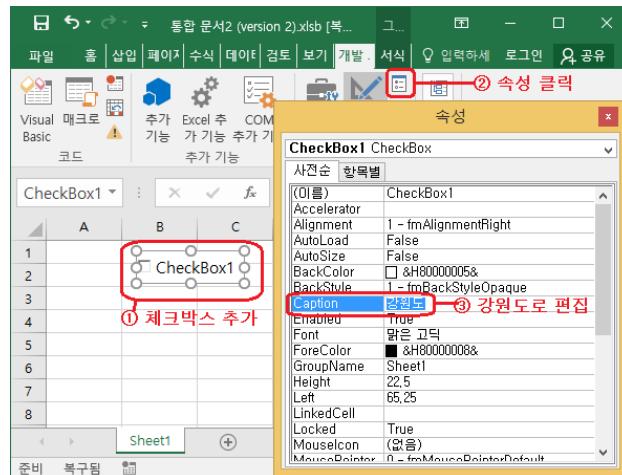


그림 8.1.19: 확인란2

3. 그림 8.1.20에서 이미 추가한 확인란 복사는 Ctrl + 확인란 테두리에서 클릭하고 마우스 드래그한다.

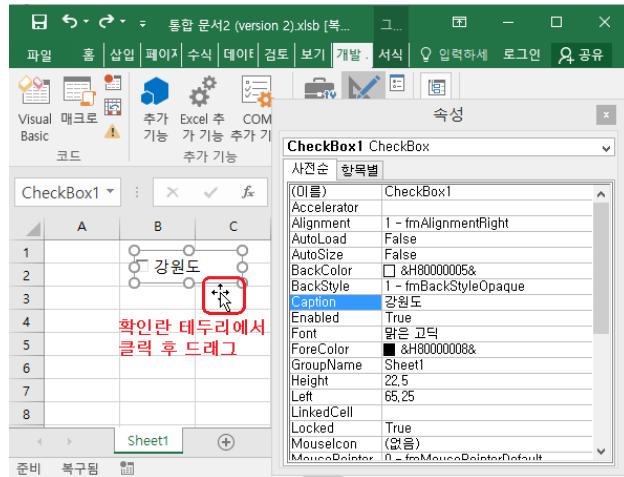


그림 8.1.20: 확인란3

4. 확인란 추가는 이전에 추가한 확인란과 가로 세로의 크기를 맞추어야지만 이미 추가한 확인란 복사는 원래의 정보와 같으므로 별도 설정없이 사용하는 장점이 있다. 그림 8.1.21은 동일한 확인란을 복사하였다. 복사한 확인란은 (이름) 속성만 제외하고 모든 속성이 같다. (이름) 속성은 변수 명으로 개체이름+숫자로 설정되며 개체가 추가되면 숫자가 1씩 자동으로 증가한다.
5. 그림 8.1.22에서 추가한 확인란은 `Caption` 속성값을 각각 경기도와 서울시로 변경한다. 코드 추가는 Visual Basic ②를 클릭하여 입력한다.
6. 그림 8.1.23에서 ①을 더블 클릭하고 `CheckBox1` 개체 ②를 선택하면 자동으로 `Click` 이벤트 프로시저가 만들어진다. 각 이벤트 프로시저에 보기 8.1.3 코드를 입력한다. 코드에서 2 줄은 `CheckBox1` 개체를 선택하면 3 줄에서 A5 셀에 `CheckBox1` 개체 `Caption` 속성을 저장한다. 4

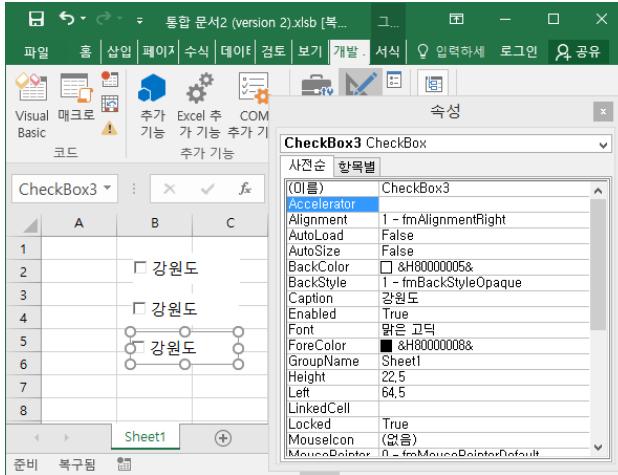


그림 8.1.21: 확인란4

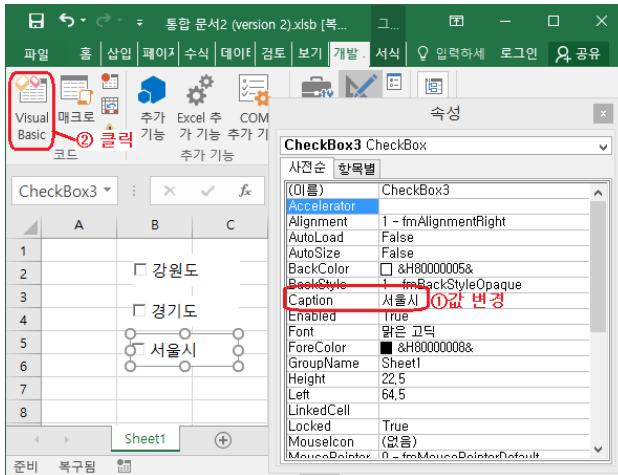


그림 8.1.22: 확인란5

줄은 CheckBox1 개체가 선택되지 않으면 A5 셀에 빈 값이 저장된다. 다시 말해 강원도를 선택하면 강원도가 A5 셀에 출력된다. 나머지 이벤트 프로시저도 동일한 내용이다.

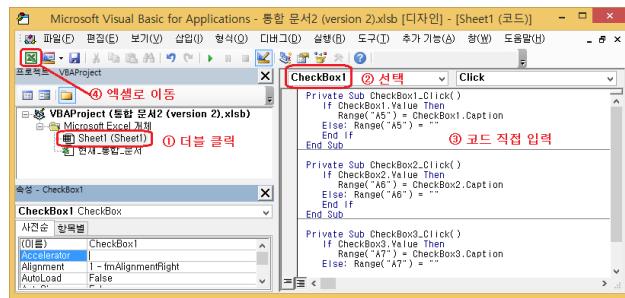


그림 8.1.23: 확인란6

### 보기 8.1.3. [확인란 이벤트]

```

1 Private Sub CheckBox1_Click()
2     If CheckBox1.Value Then
3         Range("A5") = CheckBox1.Caption
4     Else: Range("A5") = ""
5     End If
6 End Sub

8 Private Sub CheckBox2_Click()
9     If CheckBox2.Value Then
10        Range("A6") = CheckBox2.Caption
11    Else: Range("A6") = ""
12    End If
13 End Sub

15 Private Sub CheckBox3_Click()
16     If CheckBox3.Value Then
17         Range("A7") = CheckBox3.Caption
18     Else: Range("A7") = ""
19     End If
20 End Sub

```

7. 그림 8.1.24에서 디자인 모드 ①을 해제하고 추가한 확인란 개체를 선택하면 그 값이 셀에 출력된다.

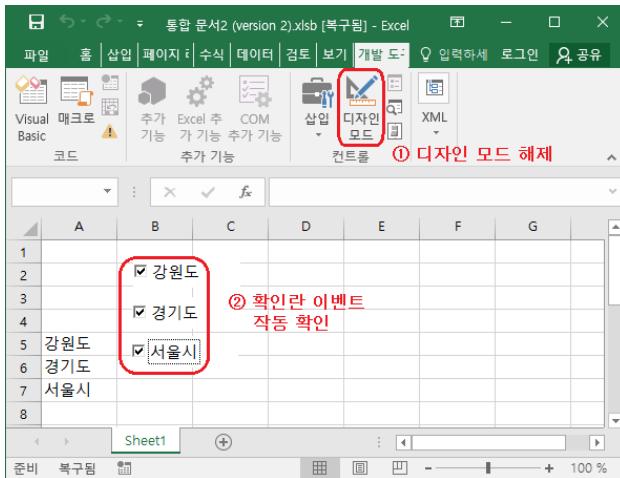


그림 8.1.24: 확인란7

#### 8.1.4 목록 상자(ListBox)

목록 상자는 추가한 목록에서 목록 하나를 선택하는 컨트롤이다. 다음은 목록 상자를 엑셀 시트에 추가하고 목록을 선택하면 이벤트가 발생할 때 실행하는 코드를 만든다.

1. 그림 8.1.25에서 개발도구→ActiveX 컨트롤에서 목록 상자를 누른다.
2. 그림 8.1.26 시트 Sheet1에 마우스 드래그로 목록 상자 ①을 추가한다. 코드는 디자인 모드를 클릭하고 Visual Basic ②을 클릭하여 Visual Basic 에디터 창에 입력한다.
3. 그림 8.1.27의 프로젝트 창에서 Sheet1 (Sheet1) ①을 더블 클릭한다. 개체는 ListBox1 ②를 선택한다. 이벤트는 Click ③을 선택하고 직접 코드를 입력한다. 코드에서 ListBox1 개체는 속성 창 (이름)에 있는 값과 같게 입력한다. 보기 8.1.4 2 줄은 Click 이벤트가 발생하였을 때 클릭한 목록 이름을 B5 셀에 저장한다. List 속성은 설정한 목록이고 ListIndex 속성은 설정한 목록을 0부터 1씩 증가하는 인덱스 값이다.

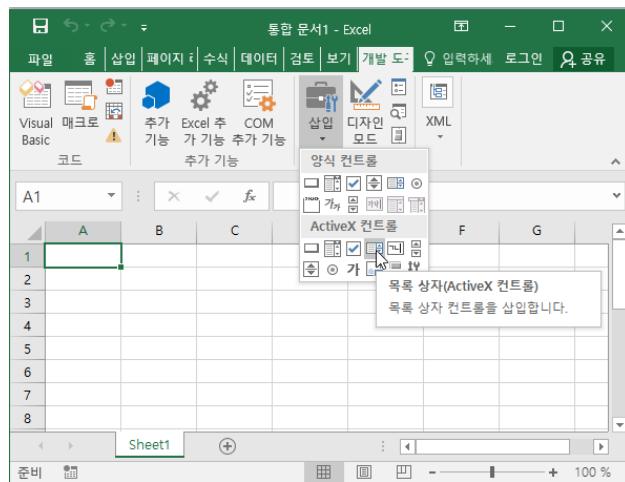


그림 8.1.25: 목록 상자1

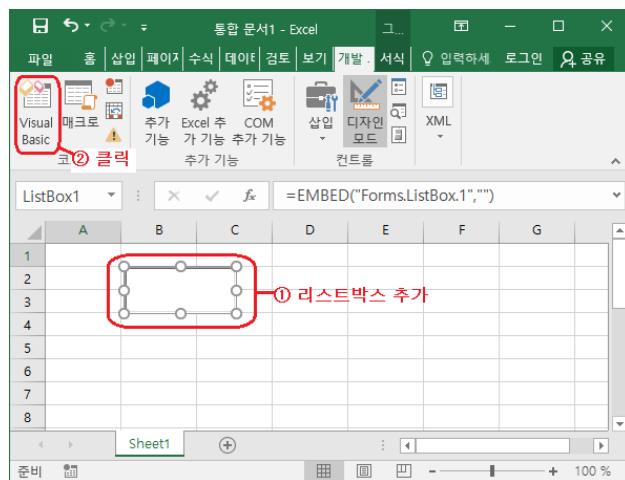


그림 8.1.26: 목록 상자2

예를 들어 강원도 목록은 `ListIndex` 값이 0이고 `List(0)`은 목록에서 강원도이다. `Activate` 이벤트는 목록에 강원도, 경기도, 서울시를 추가한다. 6 줄은 `ListBox1` 개체에 목록을 추가한다. 추가하는 목록은 여러 개이므로 배열 `Array` 함수를 이용한다.

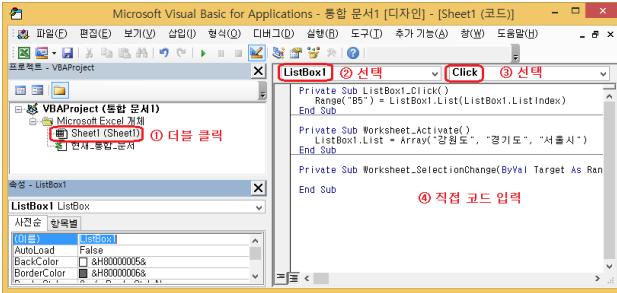


그림 8.1.27: 목록 상자3

#### 보기 8.1.4. [목록 상자 이벤트]

```

1  Private Sub ListBox1_Click()
2      Range("B5") = ListBox1.List(ListBox1.ListIndex)
3  End Sub

4
5  Private Sub Worksheet_Activate()
6      ListBox1.List = Array("강원도", "경기도", "서울시")
7  End Sub

8
9  Private Sub Worksheet_SelectionChange(ByVal Target As Range)
10
11 End Sub

```

4. 그림 8.1.28에서 디자인 모드 ①을 눌러 디자인 모드를 해제한다. 목록 상자에서 목록은 위쪽 화살표, 아래쪽 화살표 버튼을 클릭해도 없다. 목록은 Activate 이벤트가 발생해야 추가되도록 코드를 작성하였기에 아직 목록이 없다.
5. Activate 이벤트가 발생하려면 시트 Sheet1를 벌어났다가 다시 시트를 선택한다. 먼저 그림 8.1.29에 시트 Sheet2를 추가하고 이동한다.
6. WorkSheet 개체에서 Activate 이벤트를 발생하려면 그림 8.1.30에서 시트 Sheet1 ①을 클릭한다. 시트 Sheet1 목록 상자에 추가한 목록이

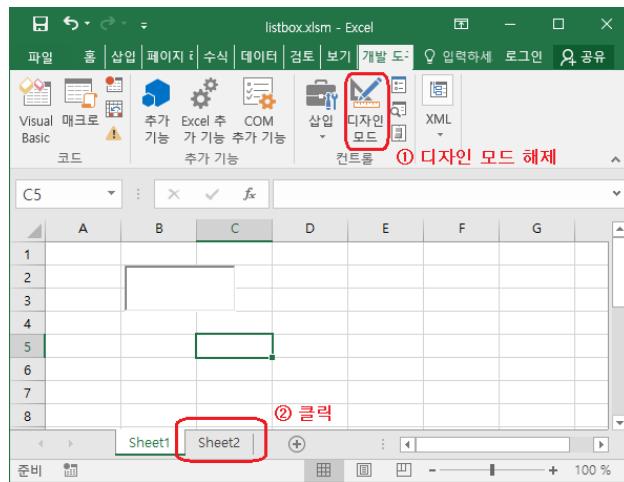


그림 8.1.28: 목록 상자4

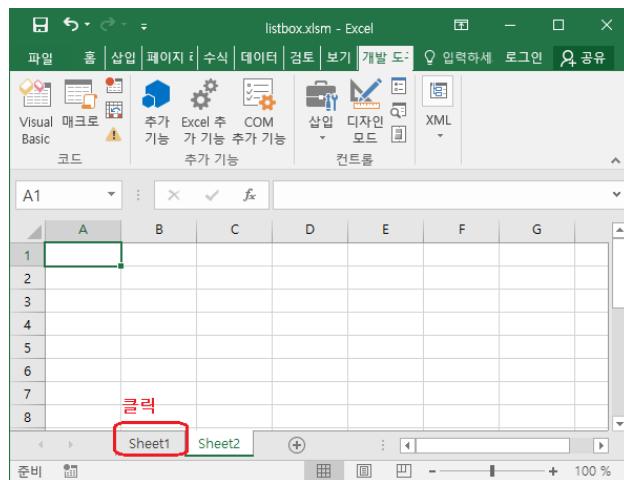


그림 8.1.29: 목록 상자5

보인다.

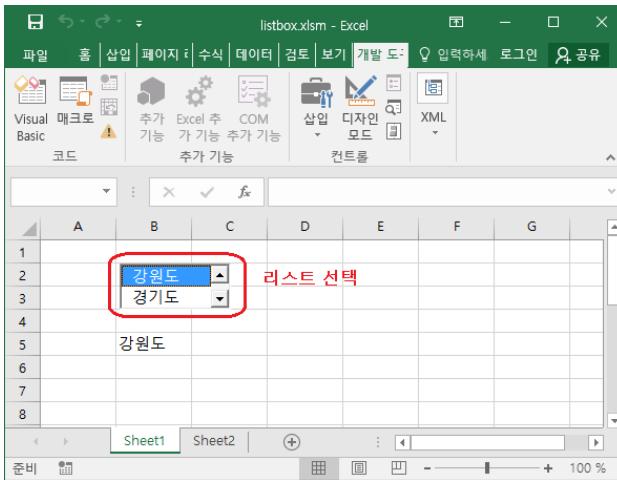


그림 8.1.30: 목록 상자6

### 8.1.5 옵션 단추(OptionButton)

옵션 단추는 추가한 옵션 목록에서 한 목록을 선택하는 컨트롤이다. 다음은 옵션 단추를 엑셀 시트에 추가하고 목록을 선택하면 이벤트가 발생할 때 실행하는 코드를 만든다.

1. 그림 8.1.31에서 개발도구→ActiveX 컨트롤에서 옵션 단추를 누른다.
2. 그림 8.1.32에서 마우스 드래그로 옵션 단추를 ①에 추가한다. 그리고 개체를 2개 복사한다. 옵션 단추 속성은 ②를 클릭한다. 옵션 단추 속성에서 Caption 속성값을 강원도, 경기도, 서울시로 변경한다. Caption 속성은 목록에 보이는 문자이다.
3. 그림 8.1.33의 프로젝트 창에서 Sheet1 (Sheet1) ①을 더블 클릭한다. 개체는 OptionButton1 ②를 선택한다. 이벤트는 Click ③을 선택하고 직접 코드를 입력한다.

#### 보기 8.1.5. [옵션 단추 이벤트1]

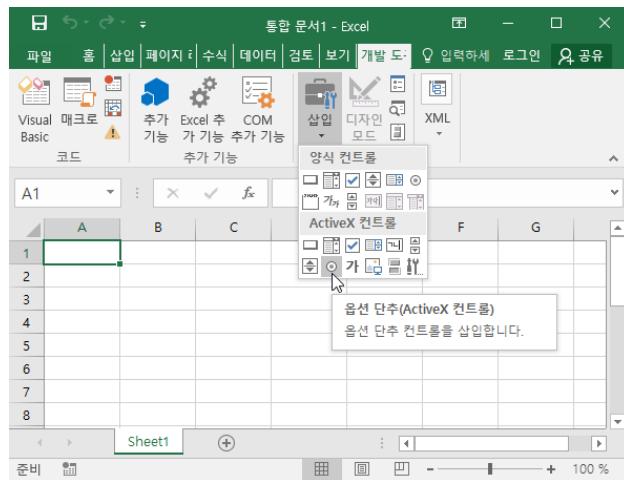


그림 8.1.31: 옵션 단추1

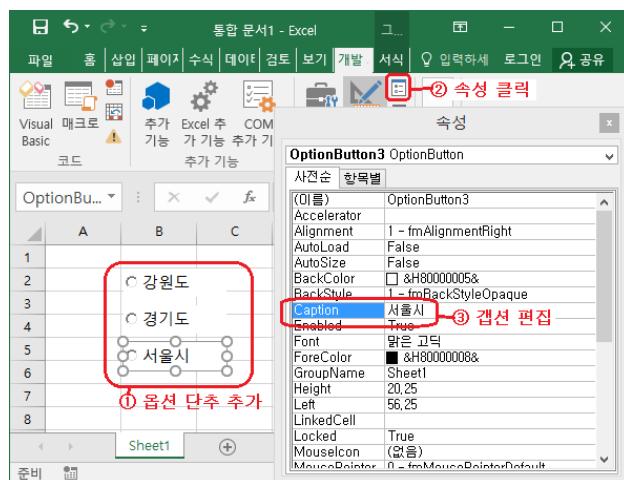


그림 8.1.32: 옵션 단추2

```

1 Private Sub OptionButton1_Click()
2     If OptionButton1.Value Then
3         Range("A3") = OptionButton1.Caption
4     Else: Range("A3") = ""

```

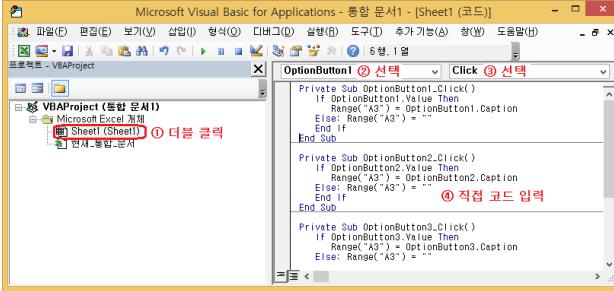


그림 8.1.33: 옵션 단추3

```

5     End If
6 End Sub
7
8 Private Sub OptionButton2_Click()
9     If OptionButton2.Value Then
10        Range("A3") = OptionButton2.Caption
11    Else: Range("A3") =
12    End If
13 End Sub
14
15 Private Sub OptionButton3_Click()
16     If OptionButton3.Value Then
17        Range("A3") = OptionButton3.Caption
18    Else: Range("A3") =
19    End If
20 End Sub

```

코드에서 OptionButton1 개체는 속성 창 (이름)에 있는 값과 같게 입력한다. 보기 8.1.5에서 2 줄은 OptionButton1 개체의 Click 이벤트가 발생하면 3 줄에서 OptionButton1 개체의 Caption인 강원도를 A3 셀에 저장한다. 옵션 단추의 Value 속성은 선택되면 True, 선택되지 않으면 False 값이다. 나머지 이벤트도 동일한 내용이다.

4. 그림 8.1.34에서 디자인 모드 ①을 해제하고 추가한 옵션 단추에서 하

나를 선택하면 옵션 단추의 Caption이 A3 셀에 출력된다.

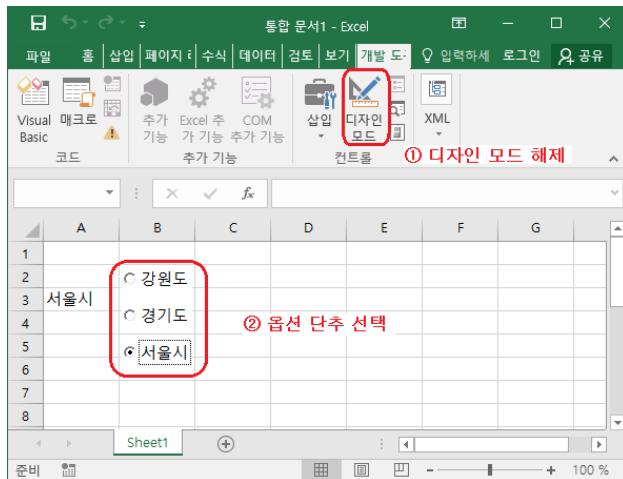


그림 8.1.34: 옵션 단추4

5. 추가한 옵션 단추는 한 집단으로 묶인다. 그러면 한 옵션 단추를 선택하면 다른 옵션 단추는 선택되지 않는다. 그림 8.1.35에서 이미 추가한 옵션 단추는 지역을 선택하는 집단이고, 추가하는 다른 집단은 셀 배경 색을 적용하는 다른 집단으로 설정한다. 옵션 단추의 집단은 `GroupName` 속성에 집단 이름을 입력하여 구분한다. 지역 집단은 기본값인 `Sheet1`이며, 색 집단은 지역 집단과 다르게 `Sheet2`로 변경한다.
6. 그림 8.1.36에서 추가한 옵션 단추 개체 ①을 선택하면 자동으로 `Click` 이벤트 프로시저가 만들어진다. 각 이벤트 프로시저에 보기 8.1.6 코드를 입력한다. 코드에서 2 줄은 `optionbutton4` 개체를 선택하면 3 줄에서 A3 셀 배경색은 노란색이 된다. 셀에 적용한 배경색 `vbYellow`는 노란색, `vbMagenta`는 자홍색, `vbCyan`는 하늘색이다. 나머지 이벤트 프로시저도 동일한 내용이다.

#### 보기 8.1.6. [옵션 단추2 이벤트]

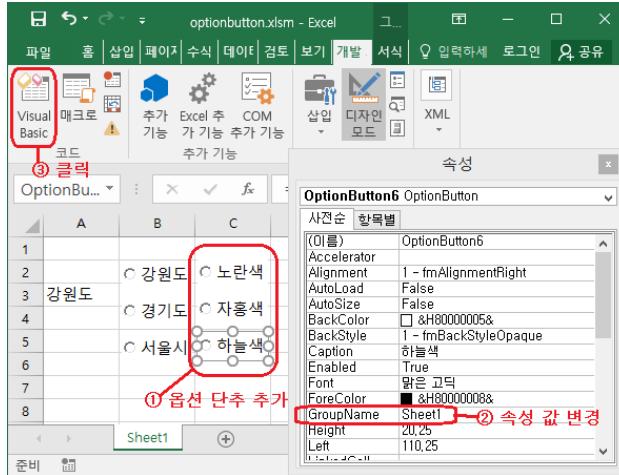


그림 8.1.35: 옵션 단추5

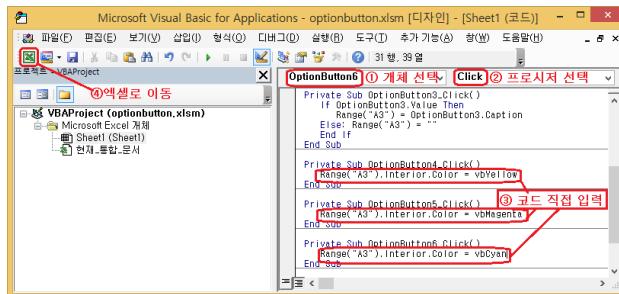


그림 8.1.36: 옵션 단추6

```

1 Private Sub OptionButton4_Click()
2     Range("A3").Interior.Color = vbYellow
3 End Sub
4
5 Private Sub OptionButton5_Click()
6     Range("A3").Interior.Color = vbMagenta
7 End Sub
8

```

```

9  Private Sub OptionButton6_Click()
10     Range("A3").Interior.Color = vbCyan
11 End Sub

```

7. 그림 8.1.37에서 추가한 옵션 단추는 두 개 집단이므로 6개 옵션 단추에서 각 집단마다 1개만 선택할 수 있다.

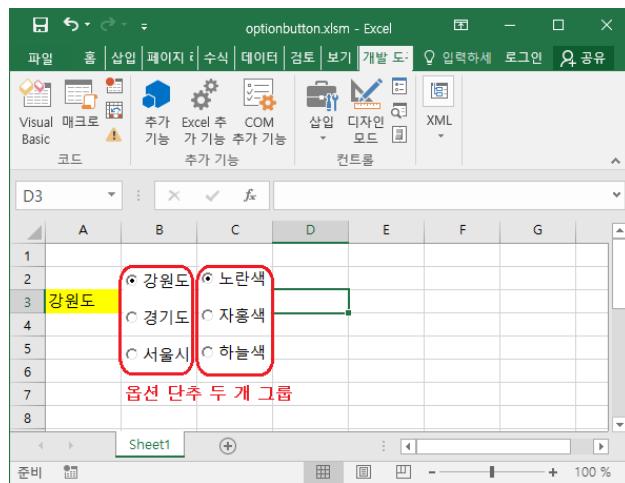


그림 8.1.37: 옵션 단추7

### 8.1.6 텍스트 상자(TextBox)

텍스트 상자는 InputBox처럼 정보를 입력받아 전달하는 컨트롤이다. 다음은 텍스트 상자에 값을 입력하면 즉시 같은 내용이 셀에 출력한다.

- 그림 8.1.38에서 개발도구→ActiveX 컨트롤에서 텍스트 상자를 누른다.
- 그림 8.1.39에서 마우스 드래그로 텍스트 상자를 ①에 추가한다. 코드는 Visual Basic ②를 클릭하여 코드 창에 입력한다.

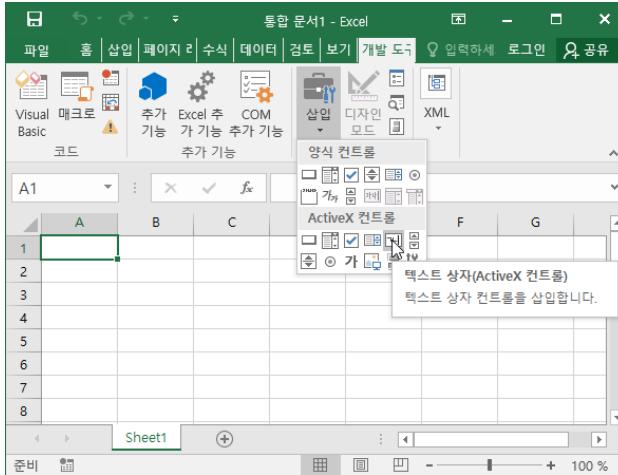


그림 8.1.38: 텍스트 상자1

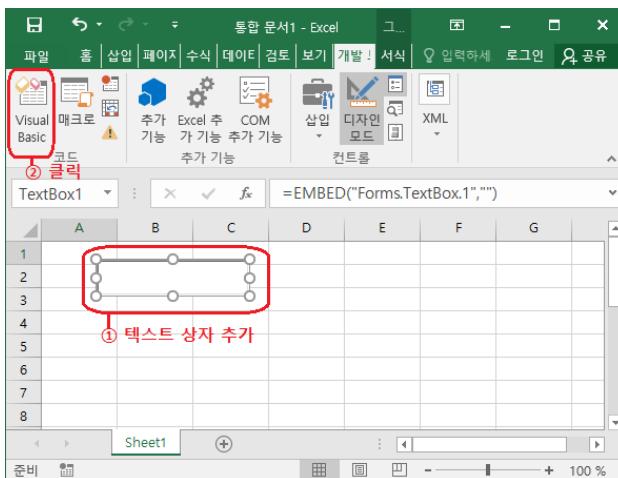


그림 8.1.39: 텍스트 상자2

3. 그림 8.1.40의 프로젝트 창에서 Sheet1 (Sheet1) ①을 더블 클릭한다. 개체는 TextBox1 ②를 선택한다. 이벤트는 Change ③을 선택하고 직접 코드를 입력한다. 보기 8.1.7에서 2 줄은 텍스트 상자에 값을 입력하면 E2 셀에 입력한 값이 나타난다.

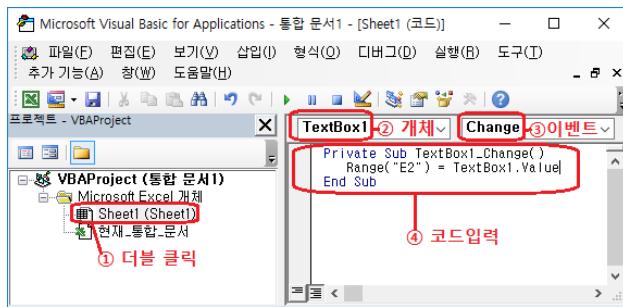


그림 8.1.40: 텍스트 상자3

### 보기 8.1.7. [텍스트 상자 이벤트]

```

1 Private Sub TextBox1_Change()
2     Range("E2") = TextBox1.Value
3 End Sub

```

4. 그림 8.1.41에서 텍스트 상자 ①에 값을 입력하면 그 결과가 ② E2 셀에 나타난다.

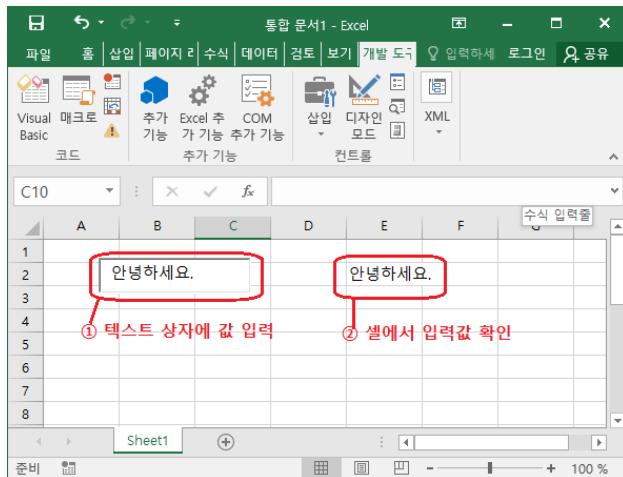


그림 8.1.41: 텍스트 상자4

### 8.1.7 스크롤 막대(ScrollBar)

스크롤 막대는 정보 내용을 일부만 화면에 보여주고 보이지 않는 내용은 위, 아래 또는 오른쪽, 왼쪽으로 이동하여 보여주는 컨트롤이다. 다음은 스크롤 막대를 엑셀 시트에 추가하고 스크롤 막대를 위, 아래로 이동할 때 스크롤 막대의 값과 배경색을 셀에 보여준다.

- 그림 8.1.42에서 개발도구→ActiveX 컨트롤에서 스크롤 막대를 누른다.

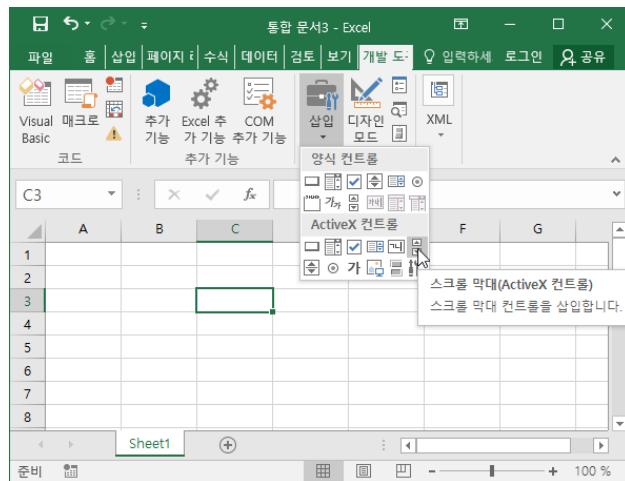


그림 8.1.42: 스크롤 막대1

- 그림 8.1.43에서 마우스 드래그로 스크롤 막대를 ①에 추가한다. 스크롤 막대 LinkedCell 속성 ②는 C3으로 변경한다. LinkedCell 속성은 Value 속성 값을 연결한 셀로 출력한다. 디자인 모드 ③과 Visual Basic ④를 클릭하고 코드를 입력한다.
- 그림 8.1.44의 프로젝트 창에서 Sheet1 (Sheet1) ①을 더블 클릭한다. 개체는 ScrollBar1 ②를 선택한다. 이벤트는 Change ③을 선택하고 직접 코드를 입력한다. 보기 8.1.8에서 2 줄은 스크롤 막대가 변경하면

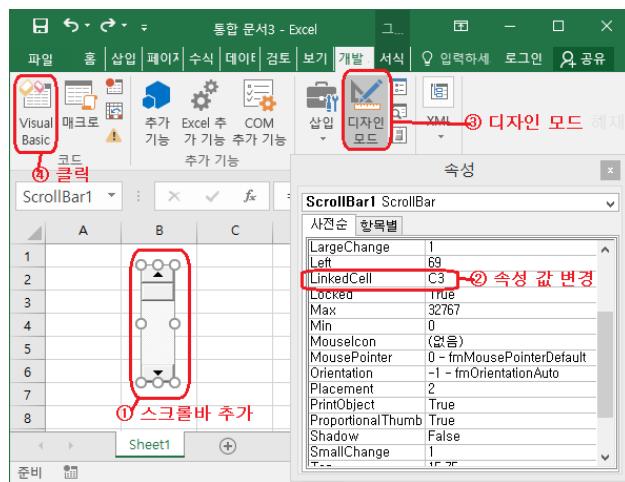


그림 8.1.43: 스크롤 막대2

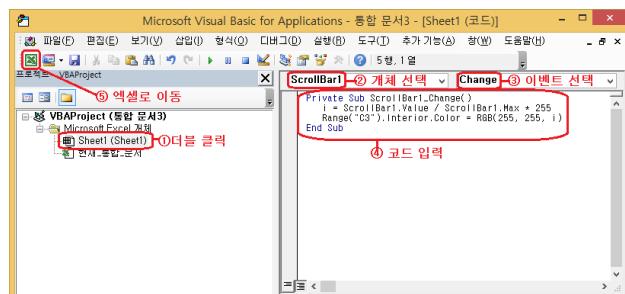


그림 8.1.44: 스크롤 막대3

스크롤 막대 현재값을 스크롤 막대 최대값으로 나누고 그 값에 255를 곱한다. 이 값 범위는 0에서 255이다. 3 줄은 C3 셀 배경색을 설정한다. 색은 변수 i 값이 0이면 노란색이고 255이면 흰색이다.

#### 보기 8.1.8. [스크롤 막대 이벤트]

```

1 Private Sub ScrollBar1_Change()
2     i = ScrollBar1.Value / ScrollBar1.Max * 255
3     Range("C3").Interior.Color = RGB(255, 255, i)

```

4 | End Sub

4. 그림 8.1.45에서 디자인 모드 ①을 해제한다. ②에서 스크롤 막대를 위, 아래로 이동시키면 이벤트가 발생하는지 확인한다.

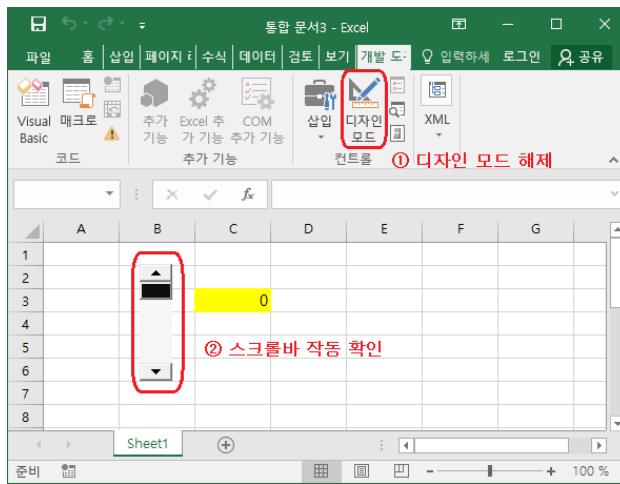


그림 8.1.45: 스크롤 막대4

### 8.1.8 스피ن 단추(SpinButton)

스핀 단추는 위 화살 버튼을 누르면 숫자가 늘어나고 아래 화살 버튼을 누르면 숫자가 줄어드는 컨트롤이다. 다음은 스피ن 단추를 엑셀 시트에 추가하고 화살표 위, 아래 버튼을 누르면 숫자가 증가, 감소하며 그 값으로 셀 배경색을 보여준다.

1. 그림 8.1.46에서 개발도구→ActiveX 컨트롤에서 스피ن 단추를 누른다.
2. 그림 8.1.47에서 마우스 드래그로 스피ن 단추를 ①에 추가한다. 속성 ②를 클릭한다. 스피ن 단추 LinkedCell 속성은 C3, 최대값 Max는 255, 최소값

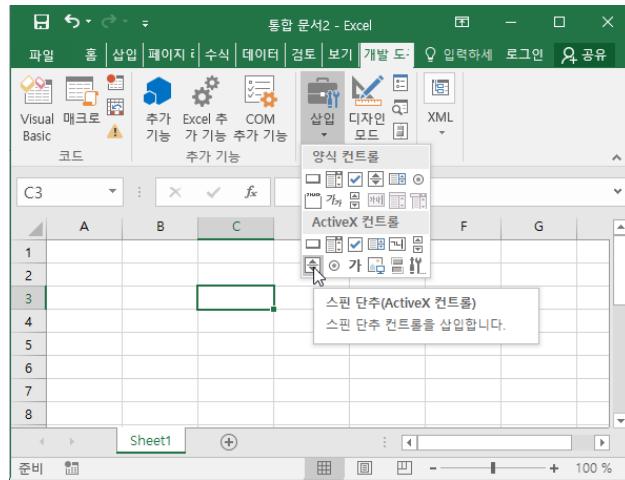


그림 8.1.46: 스판 단추1

Min은 0, 버튼을 눌렀을 때 값이 변하는 단위 SmallChange는 5로 변경한다. LinkedCell 속성은 Value 속성 값을 연결된 셀로 출력한다. Visual Basic ④를 클릭하고 코드를 입력한다.

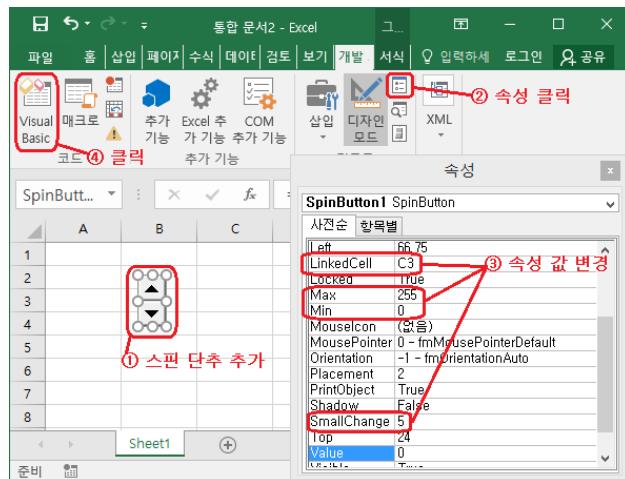


그림 8.1.47: 스크롤 막대2

3. 그림 8.1.48의 프로젝트 창에서 Sheet1 (Sheet1) ①을 더블 클릭한다. 개체는 SpinButton1 ②를 선택한다. 이벤트는 Change ③을 선택하고 직접 코드를 입력한다. 보기 8.1.9에서 2 줄은 위 화살표 버튼을 누르면

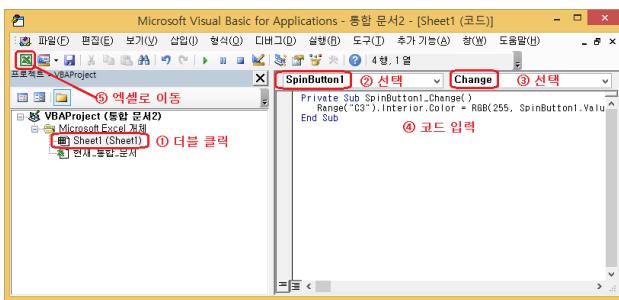


그림 8.1.48: 스픈 단추3

숫자가 5 증가, 아래 화살표는 5 감소하는 값을 RGB 함수에 입력한다. RGB는 색을 표현하는 함수이다.

#### 보기 8.1.9. [스핀 단추 이벤트]

```
1 Private Sub ScrollBar1_Change()
2     Range("C3").Interior.Color = RGB(255, SpinButton1.Value, 0)
3 End Sub
```

4. 그림 8.1.49에서 디자인 모드 ①을 해제한다. ②에서 스픈 단추를 위, 아래로 이동시키면 이벤트가 발생하는지 확인한다.

### 8.1.9 ActiveX 컨트롤을 이용한 인사관리

이 절에서 소개하는 프로그램은 콤보 상자에서 사번을 선택하면 사원 사진을 포함한 사원의 일련된 정보를 가져온다. 그림 8.1.50은 정보 시트 콤보상자에서 사번을 선택하면 DB에서 시트 사원 정보를 모두 가져온다. 프로그램을 만드는 순서는 다음과 같다.

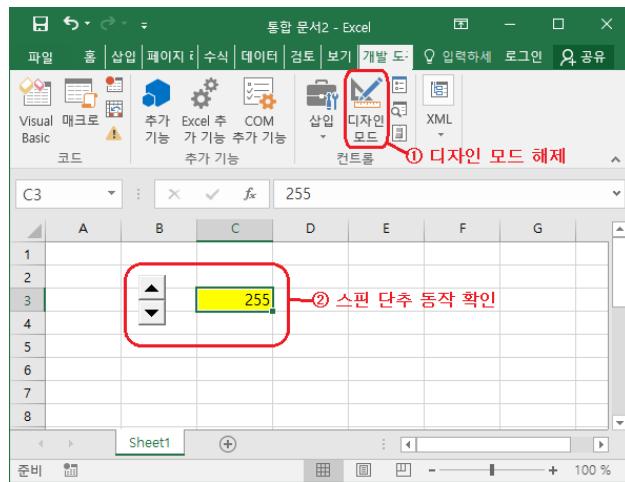


그림 8.1.49: 스펠 단추4

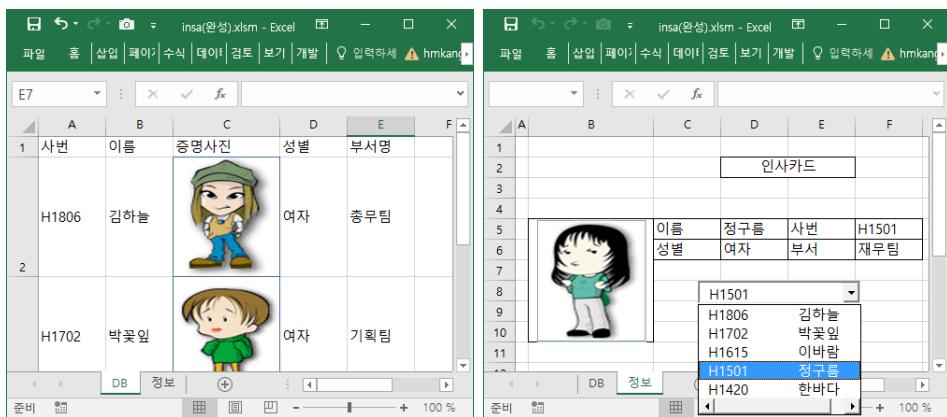


그림 8.1.50: 사원 정보

1. 콤보 상자에서 사번을 선택하면 셀에 선택한 사번이 출력된다.
2. 셀에 가져온 사번으로 이름, 성별, 부서를 VLOOKUP 함수로 가져온다.
3. 각 사람의 증명사진은 사번(A1)을 오른쪽으로 2열 아래방향으로 몇 행 이동한 위치에 있으므로 OFFSET 함수로 위치를 알아낸다.

4. 이름 관리자에 OFFSET 함수로 알아낸 위치를 등록한다.
5. 정보 시트 셀에 그림을 삽입하고 이 그림에 이름 관리자를 연결한다.
6. 콤보 상자에서 사번을 선택하면 그에 해당되는 그림이 셀에 출력된다.

프로그램을 만드는 과정을 자세하게 알아보자. 작업이 편리하게 진행되도록 여러 셀을 이름으로 정의한다. 이름은 이름 정의나 이름 관리자에서 한다. 이름 정의와 이름 관리자는 수식→정의된 이름 메뉴에 있다. 그림 8.1.51은 이름 정의와 이름 관리자로 이름을 정의하였다. 그림 8.1.51에서 ①은 정의할

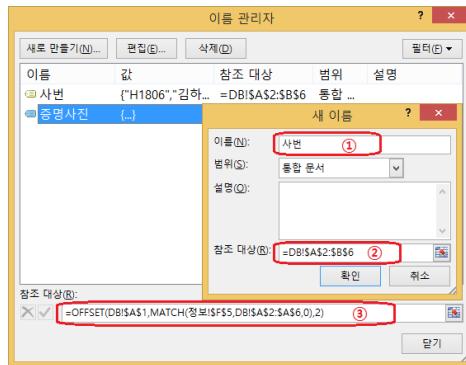


그림 8.1.51: 이름

이름이고 ②는 셀 주소이다. ③은 OFFSET, MATCH 함수 사용으로 셀 주소를 지정하였다.

OFFSET 함수는 특정한 셀 기준으로 행과 열 방향으로 이동할 셀 위치를 설정한다. 예를 들어 =OFFSET(A1,2,3)은 A1 셀 기준으로 아래쪽으로 2행 오른쪽으로 3열 이동한 D3 셀이다. 다음은 OFFSET 함수 사용법이다.

#### 사용방법 8.1.1. 【OFFSET】

```
OFFSET(reference, rows, cols, [height], [width])
```

다음은 인수에 대한 설명이다.

`reference` 기준 셀이다.

`rows` 이동할 셀의 행 위치로 양수는 아래쪽이고 음수는 위쪽이다.

`cols` 이동할 셀의 열 위치로 양수는 오른쪽이고 음수는 왼쪽이다.

`height` 이동할 셀의 높이이다. 단위는 행의 갯수이다.

`width` 이동할 셀의 너비이다. 단위는 열의 갯수이다.

MATCH 함수는 셀 범위에서 특정한 항목을 검색하여 그 항목의 상대 위치를 알려준다. 예를 들어 A1:A3 범위에 값 5, 2, 3이 있는 경우 =MATCH(2,A1:A3,0) 수식은 2가 범위에서 두 번째 항목이므로 숫자 2를 반환한다. 다음은 MATCH 함수 사용법이다.

#### 사용방법 8.1.2. 【MATCH】

```
MATCH(lookup_value, lookup_array, [match_type])
```

다음은 인수에 대한 설명이다.

`lookup_value` 일치하는지 찾을 값이다.

`lookup_array` 값이 일치하는지 검색할 셀 범위이다.

[`match_type`] 0은 일치하는 값이다. 1 또는 생략은 작거나 같은 값 중에서 최대값이다.

`lookup_array` 인수 값은 오름차순으로 정렬되어 있어야 한다. -1은 일치하는 값보다 크거나 같은 값 중에서 가장 작은 값이다. `lookup_array` 인수 값은 내림차순으로 정렬되어 있어야 한다.

그림과 같이 그림 8.1.51에서 추가한 식 ③은

```
=OFFSET(DB!$A$1,MATCH(정보!$F$5,DB!$A$2:$A$6,0),2)
```

이다. 이 식에서 OFFSET 함수에 사용한 각 인수들은

- 첫 번째 인수 DB!A1은 기준 셀이 DB 시트 A1이다.
- 두 번째 인수 MATCH(정보!F5,DB!A2:A6,0)은 정보 시트 F5 셀 값과 같은 값을 DB 시트 A2:A6 셀 범위에서 몇 번째 값과 같은지 찾는다. 다시 말해서 이동할 행의 상대적 위치이다.
- 세 번째 인수 2는 이동할 열의 상대적 위치이다.

다음은 콤보 상자를 시트에 추가한다. 콤보 상자 메뉴는 그림 8.1.9와 같이 개발도구→컨트롤→삽입→ActiveX 컨트롤에서 콤보 상자를 선택한다. 다음으로 콤보 상자가 삽입될 영역을 마우스로 선택하면 그림 8.1.10과 같이 콤보 상자가 추가된다.

추가한 콤보 상자는 그림 8.1.52와 같이 목록을 추가하고 목록과 연결할 셀을 설정한다. 그림 8.1.52에서 콤보 상자 속성 창에서

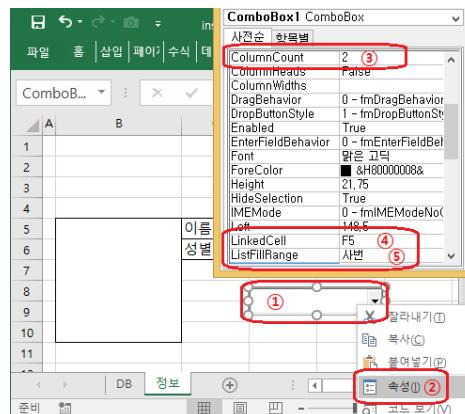


그림 8.1.52: 콤보 상자 속성

- ColumnCount는 목록에 보일 목록 갯수로 2이다.
- LinkedCell은 콤보 상자에서 선택한 첫 번째 목록을 출력할 셀이다. F5 셀은 선택한 사번이다.

- ListFillRange는 콤보 상자에 추가할 목록 범위로 이름 관리자에 추가한 사번을 입력하였다.

콤보 상자 속성 창에서 설정을 마치면 개발도구→컨트롤→디자인 모드에서 디자인 모드를 해제하고 콤보 상자에서 그림 8.1.53과 같이 아래 화설표 버튼을 눌러 추가한 목록을 확인한다. 사번으로 각 값을 가져온다. 셀 값은 정보

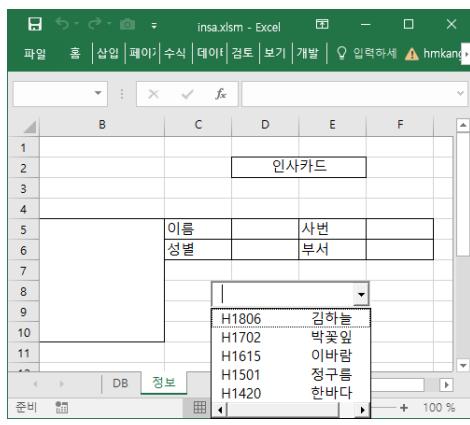


그림 8.1.53: 인사 정보 연결

시트 사번 값과 DB 시트 사번이 같을 때 각 항목을 가져온다. 이 경우 VLOOKUP 함수를 사용한다. 다음은 각 셀에 입력 값이다.

- 이름은 =VLOOKUP(F5, DB!A2:D6, 2, 0)을 입력한다. 사번이 같으면 두 번째 열 값을 가져온다.
- 성별은 =VLOOKUP(F5, DB!A2:D6, 4, 0)을 입력한다. 사번이 같으면 네 번째 열 값을 가져온다.
- 부서은 =VLOOKUP(F5, DB!A2:D6, 5, 0)을 입력한다. 사번이 같으면 다섯 번째 열 값을 가져온다.
- 사진이 들어갈 위치에 DB 시트의 김하늘 증명사진을 복사하여 붙여넣기 한다. 사진을 클릭하여 선택하고 수식 입력줄에 =증명사진을 입력한다.

다음은 VLOOKUP 함수 사용법이다.

#### 사용방법 8.1.3. 【VLOOKUP】

= VLOOKUP(조회 값, 범위, 열 번호, 찾을 값)

다음은 인수에 대한 설명이다.

조회 값 찾을 값이다.

범위 조회 값이 포함 된 셀 범위이다.

열 번호 반환 값이 포함 된 범위의 열 번호이다.

찾을 값 근사값은 TRUE 또는 정확히 일치은 FALSE를 입력한다.

#### 8.1.10 ActiveX 컨트롤을 이용한 자동필터

제 8.1 절에서 소개한 컨트롤로 자동필터를 만들자. 자동필터에 사용한 자료는 키, 몸무게, 출생년도, 종교, 성별, 결혼여부, BMI 변수 7개가 있다. 이 중에서 자동필터에 적용하는 변수는 종교, 성별, 결혼여부, BMI이다. ActiveX 컨트롤 중에서 종교는 콤보 상자, 성별은 확인란, 결혼여부는 옵션 단추, BMI는 목록 상자를 사용하였다. 그리고 설정한 자동필터 적용은 필터 명령 단추를 누르면 실행되고, 해제는 필터 해제 명령 단추를 누른다. 각 변수에 대한 자동필터의 조건 설정은 그림 8.1.54에 완성하였다. 각 컨트롤에 추가한 목록에서 종교는

- ◉ C1은 기독교
- ◉ C2는 카톨릭
- ◉ Bu는 불교
- ◉ No는 무교

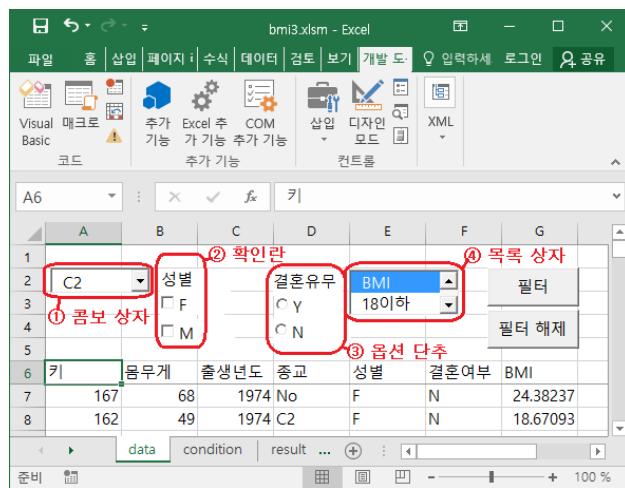


그림 8.1.54: 자동필터 완성

성별은

① F는 여자

② M은 남자

결혼여부는

③ Y는 기혼

④ N은 미혼

BMI는

⑤ 18이하

⑥ 18~25

⑦ 25~30

⑧ 30~40

### ● 40초과

이다. 보기 8.1.10은 필터 명령 단추를 클릭하면 자동으로 만들어지는 이벤트 프로시저이다.

2 - 5 줄은 자동필터에서 만들어지는 드랍다운 버튼을 보이지 않게 한다. 6 - 10 줄은 콤보 상자 목록을 자동필터 조건으로 사용한다. Criteria1은 자동필터 조건이고, ComboBox1.List는 콤보 상자에 추가한 목록을 가져온다. ComboBox1.ListIndex는 콤보 상자에 추가한 목록을 0부터 1씩 증가하는 인덱스 값이다. field:=4는 네 번째 열에 대한 조건을 설정한다.

11 - 19 줄은 확인란 목록을 자동필터 조건으로 설정한다. 확인란은 여러 목록을 선택할 수 있다. 따라서 11 줄은 두 개 목록이 모두 True이면 모든 값이 보여지도록 하였다. Caption 속성은 목록에 사용한 문자이다.

20 - 27 줄은 옵션 단추 목록을 자동필터 조건으로 사용한다. 옵션 단추는 한 목록을 선택하면 다른 목록은 선택이 해제된다. 따라서 각 목록에 대한 조건을 설정한다.

28 - 46 줄은 목록 상자 목록을 자동필터 조건으로 설정한다. 조건문은 Select Case문을 사용하였다. 자동필터에 설정한 조건은 범위이므로 조건 입력값이 2개이다. 조건이 두 개이고 범위가 공통 부분이면 교집합이다. 이런 경우 연산자는 논리곱을 사용하고 설정값은 xlAnd이다.

지금까지 코드는 너무 길어 기억하기도 힘들고 사용하기도 어렵다. 그래서 저자는 자동필터 과정을 매크로로 기록하여 어떤 함수와 그에 대한 속성, 메소드가 사용되었는지 확인하고 코드 입력에 활용하였다.

### 보기 8.1.10. 【필터 명령 단추 이벤트 프로시저】

```

1 Private Sub CommandButton1_Click()
2     For i = 1 To 7
3         ActiveSheet.Range("$A$6:$G$183").AutoFilter field:=i, _
4             VisibleDropDown:=False
5     Next i
6     If ComboBox1.ListIndex > 0 Then
7         ActiveSheet.Range("$A$6:$G$183").AutoFilter field:=4, _
8             Criteria1:=ComboBox1.List(ComboBox1.ListIndex)

```

```
9 Else: ActiveSheet.Range("$A$6:$G$183").AutoFilter field:=4
10 End If
11 If CheckBox1 And CheckBox2 Then
12     ActiveSheet.Range("$A$6:$G$183").AutoFilter field:=5
13 ElseIf CheckBox1 Then
14     ActiveSheet.Range("$A$6:$G$183").AutoFilter field:=5, _
15         Criteria1:=CheckBox1.Caption
16 ElseIf CheckBox2 Then
17     ActiveSheet.Range("$A$6:$G$183").AutoFilter field:=5, _
18         Criteria1:=CheckBox2.Caption
19 End If
20 If OptionButton1 Then
21     ActiveSheet.Range("$A$6:$G$183").AutoFilter field:=6, _
22         Criteria1:=OptionButton1.Caption
23 ElseIf OptionButton2 Then
24     ActiveSheet.Range("$A$6:$G$183").AutoFilter field:=6, _
25         Criteria1:=OptionButton2.Caption
26 Else: ActiveSheet.Range("$A$6:$G$183").AutoFilter field:=6
27 End If
28 Select Case ListBox1.ListIndex
29     Case 0
30         ActiveSheet.Range("$A$6:$G$183").AutoFilter field:=7
31     Case 1
32         ActiveSheet.Range("$A$6:$G$183").AutoFilter field:=7, _
33             Criteria1:="<=18.5"
34     Case 2
35         ActiveSheet.Range("$A$6:$G$183").AutoFilter field:=7, _
36             Criteria1:=">18.5", Operator:=xlAnd, Criteria2:="<=25"
37     Case 3
38         ActiveSheet.Range("$A$6:$G$183").AutoFilter field:=7, _
39             Criteria1:=">25", Operator:=xlAnd, Criteria2:="<=35"
40     Case 4
41         ActiveSheet.Range("$A$6:$G$183").AutoFilter field:=7, _
42             Criteria1:=">30", Operator:=xlAnd, Criteria2:="<=40"
43     Case 5
44         ActiveSheet.Range("$A$6:$G$183").AutoFilter field:=7, _
45             Criteria1:=">40"
46 End Select
```

47      End Sub

보기 8.1.11은 필터 해제 명령 단추를 클릭하면 자동으로 만들어지는 이벤트 프로시저이다. 2 줄은 자동필터가 설정되었으면 설정을 해제한다. 3 - 6 줄은 확인란과 옵션 단추 목록값을 선택하지 않은 값으로 초기화한다.

#### 보기 8.1.11. 【필터 해제 명령 단추 이벤트 프로시저】

```

1  Private Sub CommandButton2_Click()
2      ActiveSheet.Range("$A$6:$G$183").AutoFilter
3      CheckBox1.Value = False
4      CheckBox2.Value = False
5      OptionButton1.Value = False
6      OptionButton2.Value = False
7  End Sub

```

보기 8.1.12는 프로젝트 창에서 Microsoft Excel 개체 → 현재\_통합\_문서 를 클릭한다. 그러면 나타나는 코드 입력 창에서 Workbook 개체를 선택하고 Open 메소드를 선택하면 자동으로 만들어지는 이벤트 프로시저이다. 2 - 3 줄은 콤보 상자, 4 - 5 줄은 목록 상자에 사용할 목록을 추가한다. 이 프로시저는 엑셀 파일을 열면서 실행한다. 따라서 엑셀 파일을 실행하면 자연스럽게 각 컨트롤에 목록이 추가된다.

#### 보기 8.1.12. 【Workbook\_Open 이벤트 프로시저】

```

1  Private Sub Workbook_Open()
2      Worksheets("data").ComboBox1.List = Array("종교", "C1", _
3          "C2", "Bu", "No")
4      Worksheets("data").ListBox1.List = Array("BMI", "180이하", _
5          "18~25", "25~30", "30~40", "40이상")
6  End Sub

```

## 제 8.2 절 양식 컨트롤

양식 컨트롤은 VBA 코드를 사용하지 않고 셀 데이터와 값을 주고 받는 기능을 제공한다. 또한 양식 컨트롤과 매크로를 연결하고 실행도 한다. 양식 컨트롤에 추가된 목록은 INDEX 함수로 값을 가져온다.

### 8.2.1 명령 단추(CommandButton)

명령 단추는 흔히 말하는 버튼이다. 이 컨트롤은 주로 엑셀에서 매크로 연결과 실행에 사용한다. 그림 8.2.1은 VBA 코드를 만들고 명령 단추에 매크로를 연결하여 실행하였다. 명령 단추와 매크로 연결과정은 다음과 같다.

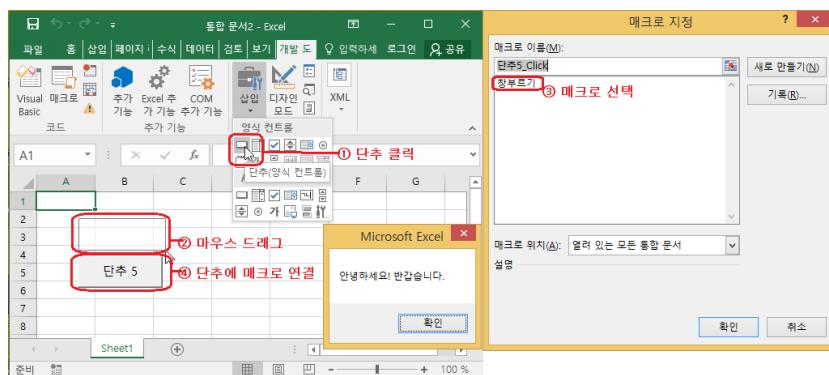


그림 8.2.1: 명령 단추

1. 양식 컨트롤에서 단추를 클릭한다.
2. 마우스 드래그로 단추가 들어갈 영역을 정한다.
3. 마우스 버튼을 놓으면 매크로 지정 창이 나타난다. 이 창에서 연결할 매크로 창부르기를 선택한다.
4. 매크로가 연결된 버튼 단추 5가 마우스 드래그한 영역에 만들어진다.

5. 단추를 클릭하면 메시지 박스가 나타난다.

보기 8.2.1은 버튼과 연결한 매크로 코드이다.

#### 보기 8.2.1. [버튼과 연결한 매크로 코드]

```

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

```

#### 8.2.2 콤보 상자(ComboBox)

콤보 상자는 추가한 목록에서 한 목록을 선택하는 컨트롤이다. 그림 8.2.2는 콤보 상자를 만들고 셀에 있는 목록을 콤보 상자에 추가한다. 추가한 목록은 드랍 다운 버튼을 눌러 선택하면 해당 목록번호가 연결된 셀에 나타난다. 콤보

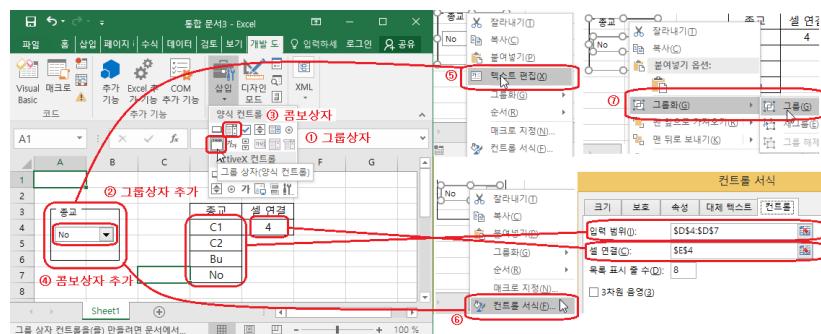


그림 8.2.2: 콤보 상자

상자 연결과정은 다음과 같다.

1. 양식 컨트롤에서 그룹 상자를 누르고 이 컨트롤이 추가될 영역을 선택 한다.
2. 양식 컨트롤에서 콤보 상자를 누르고 이 컨트롤이 추가될 영역을 선택 한다.

3. 추가한 그룹 상자는 텍스트 편집 메뉴를 누르고 종교로 변경한다.
4. 추가한 콤보 상자는 컨트롤을 서식 메뉴에서 입력 범위와 셀 연결에 해당하는 셀을 선택한다. 입력 범위는 콤보 상자에 추가할 목록이고 셀 연결은 콤보 상자에서 선택한 목록 번호가 셀에 나타난다. 목록 번호는 위에서 아래로 1, 2, 3, ... 순이다.
5. 콤보 상자와 그룹 상자는 하나로 묶어야 이동이 편리하다. 여러 컨트롤을 한 개체로 묶는 것은 여러 컨트롤을 선택하고 마우스 오른쪽 버튼 → 그룹화 → 그룹 메뉴를 선택한다.

양식 컨트롤에서 컨트롤 선택은 컨트롤에 마우스를 옮기고 오른쪽 버튼을 누른다.

### 8.2.3 확인란(CheckBox)

확인란은 추가한 확인 목록에서 여러 목록을 선택하는 컨트롤이다. 그림 8.2.3은 확인란을 만들고 확인란 값은 셀에 연결하여 보여준다. 확인란 값은 해당 확인란을 선택하면 TRUE 선택하지 않으면 FALSE다. 확인란 연결과정은 다음과

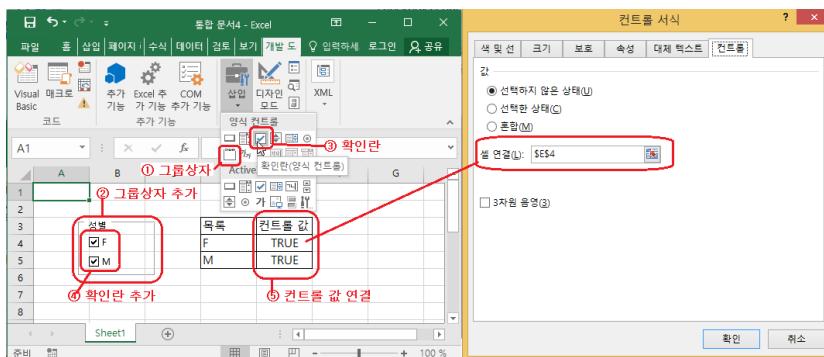


그림 8.2.3: 확인란

같다.

1. 양식 컨트롤에서 그룹 상자를 누르고 이 컨트롤이 추가될 영역을 선택 한다.

2. 양식 컨트롤에서 확인란을 누르고 이 컨트롤이 추가될 영역을 선택한다.
3. 추가한 확인란은 텍스트 편집 메뉴를 누르고 각각 F, M 문자를 입력한다.
4. 추가한 확인란은 컨트롤 서식 메뉴에서 셀 연결에 해당하는 셀을 선택한다. 셀 연결은 확인란 값이 셀에 나타난다.
5. 확인란과 그룹 상자는 하나로 묶어야 이동이 편리하다. 여러 컨트롤을 한 개체로 묶는 것은 여러 컨트롤을 선택하고 마우스 오른쪽 버튼 → 그룹화 → 그룹 메뉴를 선택한다.

#### 8.2.4 목록 상자(ListBox)

목록 상자는 추가한 목록에서 한 목록을 선택하는 컨트롤이다. 그림 8.2.4는 목록 상자를 만들고 셀에 있는 목록을 목록 상자에 추가한다. 추가한 목록은 위, 아래 버튼을 눌러 선택하면 해당 목록번호가 연결된 셀에 나타난다. 목록

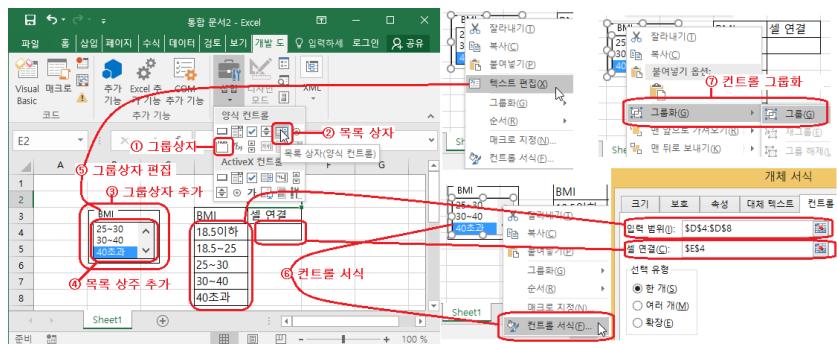


그림 8.2.4: 목록 상자

상자 연결과정은 다음과 같다.

1. 양식 컨트롤에서 그룹 상자를 누르고 이 컨트롤이 추가될 영역을 선택한다.

2. 양식 컨트롤에서 목록 상자를 누르고 이 컨트롤이 추가될 영역을 선택 한다.
3. 추가한 그룹 상자는 텍스트 편집 메뉴를 누르고 BMI로 변경한다.
4. 추가한 목록 상자는 컨트롤 서식 메뉴에서 입력 범위와 셀 연결에 해당하는 셀을 선택한다. 입력 범위는 목록 상자에 추가할 목록이고 셀 연결은 목록 상자에서 선택한 목록 번호가 셀에 나타난다. 목록 번호는 위에서 아래로 1, 2, 3, ... 순이다.
5. 목록 상자와 그룹 상자는 하나로 묶어야 이동이 편리하다. 여러 컨트롤을 한 개체로 묶는 것은 여러 컨트롤을 선택하고 마우스 오른쪽 버튼 → 그룹화 → 그룹 메뉴를 선택한다.

### 8.2.5 옵션 단추(OptionButton)

옵션 단추는 추가한 옵션 목록에서 한 목록을 선택하는 컨트롤이다. 그림 8.2.5는 옵션 단추를 만들고 옵션 단추 값을 셀에 연결하여 보여준다. 옵션 단추는 한 컨트롤을 선택하면 다른 것들은 모두 선택되지 않는다. 옵션 단추 연결과

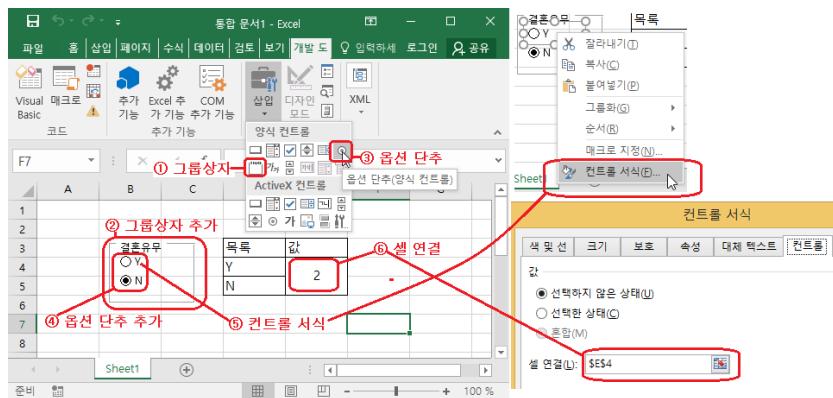


그림 8.2.5: 옵션 단추

정은 다음과 같다.

1. 양식 컨트롤에서 그룹 상자를 누르고 이 컨트롤이 추가될 영역을 선택 한다.
2. 양식 컨트롤에서 옵션 단추를 누르고 이 컨트롤이 추가될 영역을 선택 한다.
3. 추가한 옵션 단추는 기본으로 생성된 optionbutton1 문자를 클릭하여 문자 Y를 입력하고 다른 컨트롤도 동일하게 진행한다.
4. 추가한 옵션 단추는 컨트롤 서식 메뉴에서 셀 연결에 해당하는 셀을 선택한다. 셀 연결은 선택한 옵션 단추 값이 연결한 셀에 나타난다.
5. 옵션 단추와 그룹 상자는 하나로 묶어야 이동이 편리하다. 여러 컨트롤을 한 개체로 묶는 것은 여러 컨트롤을 선택하고 마우스 오른쪽 버튼 → 그룹화 → 그룹 메뉴를 선택한다.

### 8.2.6 양식 컨트롤을 이용한 고급필터

제 8.2.1 절에서 소개한 컨트롤로 고급필터를 만들자. 고급필터에 사용한 자료는 258 면, 제 8.1.10 절 자동필터에서 사용한 것과 동일하다. 이 중에서 고급필터에 적용하는 변수는 종교, 성별, 결혼여부, BMI이다. 양식 컨트롤 중에서 종교는 콤보 상자, 성별은 확인란, 결혼여부는 옵션 단추, BMI는 목록 상자를 사용하였다. 그리고 설정한 고급필터 적용은 필터 명령 단추를 누르면 실행된다.

각 변수에 대한 고급필터의 조건 설정은 그림 8.2.6에 완성하였다. 각 컨트롤에 추가한 목록에서 종교는

- ◉ C1은 기독교
- ◉ C2는 카톨릭
- ◉ Bu는 불교
- ◉ No는 무교

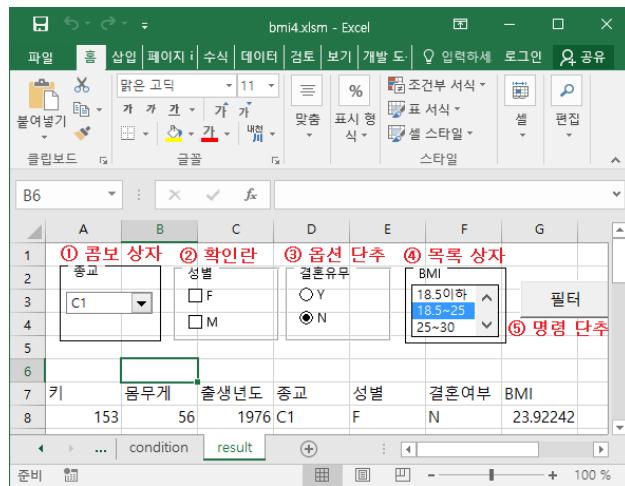


그림 8.2.6: 고급필터 완성

성별은

① F는 여자

② M은 남자

결혼유무는

③ Y는 기혼

④ N은 미혼

BMI는

① 180|하

② 18~25

③ 25~30

④ 30~40

### ④ 40초과

이다. 그림 8.2.7에서

종교	성별	결혼여부	BMI
C1	F	N	$\geq 18.5$
C2	M	N	$< 25$

종교	성별	결혼여부	BMI
2	TRUE	2	2
	TRUE		

그림 8.2.7: 고급필터 조건, 양식컨트롤 참조, 셀 연결

- ④ 양식 컨트롤 참조에서 종교는 콤보상자, BMI는 목록상자에 사용한 리스트
- ⑤ 양식 컨트롤 값은 양식 컨트롤 이벤트 값을 출력
- ⑥ 고급 필터 조건은 양식 컨트롤 값을 이용하여 필터 조건에 사용
- ⑦ BMI 범위는 고급 필터 조건에서 BMI 조건에 사용

한다. 그림 8.2.7에 있는 고급 필터 조건에서

- ⑧ 종교는 INDEX 함수로 조건의 설정 값을 만든다. D5에 설정한 콤보 상자 조건 =INDEX(A5:A8,D10)은 목록 A5:A8에서 양식 컨트롤 값 D10에 해당되는 목록 번호이다. 예를 들어 셀 D10에 2가 있으면 설정 값은 목록에서 두 번째 값 C2이다.

- ◉ 성별은 IF 함수로 조건의 설정 값을 만든다. E5에 설정한 확인란 조건은 =IF(E10,"F",IF(E11,"M",""))이며 선택한 목록은 모두 검색할 수 있다.
- ◉ 결혼여부는 IF 함수로 조건의 설정 값을 만든다. F5에 설정한 옵션 단추 조건은 =IF(F10=1,"Y",IF(F10=2,"N",""))이다. 즉 양식 컨트롤 값이 1이면 Y, 아니면 N이 설정값이다.
- ◉ BMI는 VLOOKUP 함수로 조건을 만든다. 조건은 무엇에서 무엇까지이므로 설정값이 두 개이다. =">=&VLOOKUP(G10,J5:L9,2,FALSE)는 처음 설정값으로 G10 값과 J5:J9에서 같은 값을 찾고 해당 케이스의 두 번째 값을 가져온다. 예를 들어 G10 값이 3라면 J5:J9에서 같은 값 3을 찾고 그 케이스에 있는 값이 3 25 30이므로 두 번째 값 25가 설정값이다. 설정값 ="<&VLOOKUP(G10,J5:L9,3,FALSE)는 일치하는 값에서 3 번째 값 30이다. 따라서 BMI 조건 설정값은 25 ~ 30이다.

### 보기 8.2.2. [고급필터 프로시저]

```

1 Sub 필터()
2     Rows("7:200").Clear
3     Range("data!A6:G183").AdvancedFilter Action:=xlFilterCopy, _
4         CriteriaRange:=Range("condition!D4:H6"), _
5         CopyToRange:=Range("A7"), Unique:=False
6 End Sub

```

보기 8.2.2에서 1 줄은 7 ~ 200 행에 저장된 값을 모두 삭제한다. 2 ~ 5 줄은 한 문장을 여러 줄에 작성하였다. 그 내용은 그림 8.2.8에 설정한 값이다.

- ◉ 보기 8.2.2 3 줄에서 Range("data!A6:G183").AdvancedFilter는 고급 필터하려는 자료 설정으로 그림 8.2.7에서 고급필터 목록 범위
- ◉ 보기 8.2.2 3 줄에서 Action:=xlFilterCopy는 그림 8.2.7에서 다른 장소에 복사로 고급필터 결과를 출력 장소 지정

- 보기 8.2.2 4 줄에서 CriteriaRange:=Range("condition!D4:H6")은 그림 8.2.7에서 조건 범위로 고급필터하려는 조건
- 보기 8.2.2 5 줄에서 CopyToRange:=Range("A7")은 그림 8.2.7에서 복사 위치 고급 필터 결과를 출력할 장소를 명시
- 보기 8.2.2 5 줄에서 Unique:=False는 그림 8.2.7에서 동일한 레코드는 하나만에서 체크 표시 해제

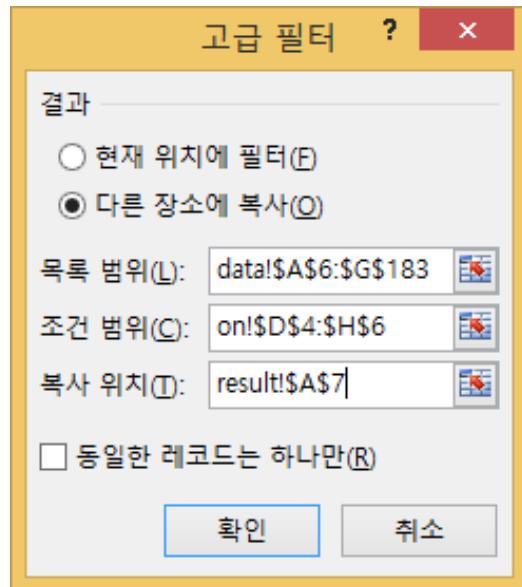


그림 8.2.8: 고급필터 조건

## 연습문제

주어진 문제는 양식 컨트롤과 ActiveX 컨트롤을 각각 사용한다. 즉 첫 번째는 양식 컨트롤만 사용하고 나머지 한 번은 ActiveX 컨트롤을 사용한다.

1. 어떤 연구자가 인구통계학적 특성에 대하여 조사표(설문지)를 다음과 같이 만든다.

(a) 당신이 살고 있는 지역은?

지역은 강원도, 경기도, 인천광역시, 서울특별시, 충청북도, 충청남도, 대전광역시, 세종특별자치시, 경상북도, 대구광역시, 울산광역시, 경상남도, 부산광역시, 전라북도, 전라남도, 광주광역시, 제주특별자치도 중에서 하나를 선택할 수 있다.

(b) 당신 나이는?

나이는 19세 이하, 20 ~ 29, 30 ~ 39, 40 ~ 49, 50 ~ 59, 60 ~ 69, 70 이상 중에서 하나를 선택할 수 있다.

(c) 당신 직장명은?

직장명은 직접 입력한다.

(d) 당신 성별은?

성별은 여자, 남자 중 하나를 선택할 수 있다.

2. 어떤 연구 대상자가 만든 조사표에 응답하면 그 결과가 다른 시트에 나타나게 만든다.

3. 인구통계학적 특성에서 나이를 직접 입력하였을 때 연령대를 구분하여 저장되도록 만들어라. 연령대 목록은 19세 이하, 20 ~ 29, 30 ~ 39, 40 ~ 49, 50 ~ 59, 60 ~ 69, 70 이상이다.



## 제 9 장

# 사용자 정의 폼(User Form)

윈도에서 응용프로그램들은 사용자에게 편리함을 주려고 GUI(Graphic User Interface)를 많이 제공한다. 응용프로그램은 창이나 패널에 버튼, 입력상자, 옵션버튼, 리스트박스, 메시지 같은 컴포넌트를 추가하고 이들의 이벤트가 발생할 때 코드를 입력하여 동작을 제어할 수 있다. 엑셀도 사용자가 창이나 패널에 다양한 인터페이스를 추가하고 이벤트에 반응하는 것을 7장 매크로에서 보았다. VBA에서 컴포넌트를 추가하고, 이벤트 반응에 대한 연동 가능한 공간을 사용자 정의 폼(UserForm, 유저폼)이라 부른다.

### 제 9.1 절 사용자 정의 폼 작성과 실행

사용자 정의 폼 작성은 Visual Basic Editor에서 삽입→사용자 정의 폼 메뉴를 선택한다(그림 9.1.1). 사용자 정의 폼이 비주얼베이직 편집기에 추가되면

- 프로젝트 창이 왼쪽 위쪽에 위치(그림 9.1.1)
- 속성 창은 사용자 정의 폼에 대한 속성(그림 9.1.1)
- 사용자 정의 폼이 코드 입력 창에 위치(그림 9.1.1)

- 도구상자가 별도의 창에 나타나며 이 창에서 버튼, 레이블, 텍스트박스, 옵션 버튼 등 컨트롤을 마우스 클릭으로 사용자 정의 폼에 추가한다.

그림 9.1.1은 사용자가 만든 사용자 정의 폼이다. 그림 9.1.1에서 폼을 클릭하면 아래에 속성이 보인다. 그림 9.1.1에 속성창 이름은 UserForm1에서 uf1로 변경하였다. 이 의미는 UserForm 개체 변수명을 uf1로 선언하였다. 보기 9.1.1에서 2 줄은 유저폼 uf1 속성 Show를 실행한다. 즉 유저폼 보여달라는 의미다. 실제로 해당 유저폼이 나타난다. 그림 9.1.1에서 Caption 속성을 유저폼 부르기로 변경하면 오른쪽 유저폼에 제목표시줄 텍스트가 유저폼 부르기로 변한다. 그림 9.1.1에서 속성 값을 변경하면 바로 결과가 반영되어 즉시 확인이 가능하다.

#### 보기 9.1.1. [유저폼 부르기]

```
1 Sub 유저폼부르기()
2     uf1.Show
3 End Sub
```

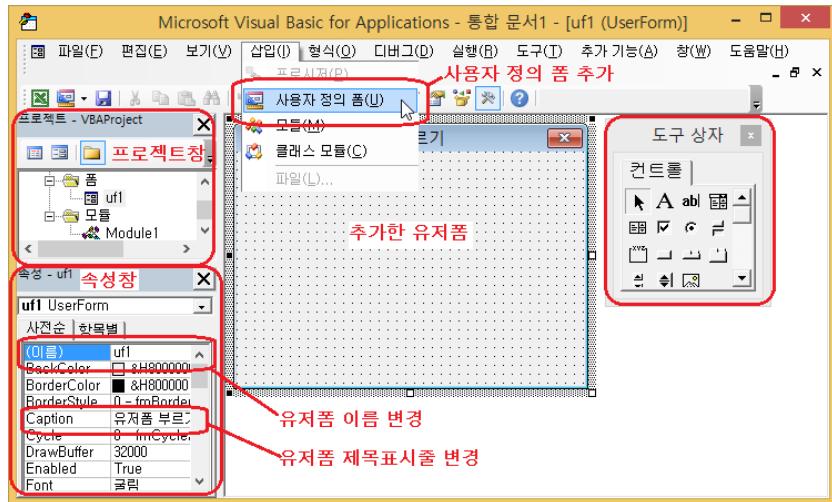


그림 9.1.1: 유저폼 추가와 속성값 변경

사용자 정의 폼을 만들고 호출하는 과정은 다음과 같다.

1. Visual Basic Editor에서 삽입→사용자 정의 폼 메뉴를 실행하여 유저 폼을 추가한다. 속성 창에 유저폼 이름은 uf1로 변경한다(그림 9.1.1).
2. 유저폼은 UserForm 객체 Show 메소드를 실행하여 호출한다. 유저폼을 호출하려면 Visual Basic Editor에서 삽입→모듈 메뉴 실행하여 모듈을 삽입한다. 이 모듈에 보기 9.1.1 코드를 입력한다(그림 9.1.2).

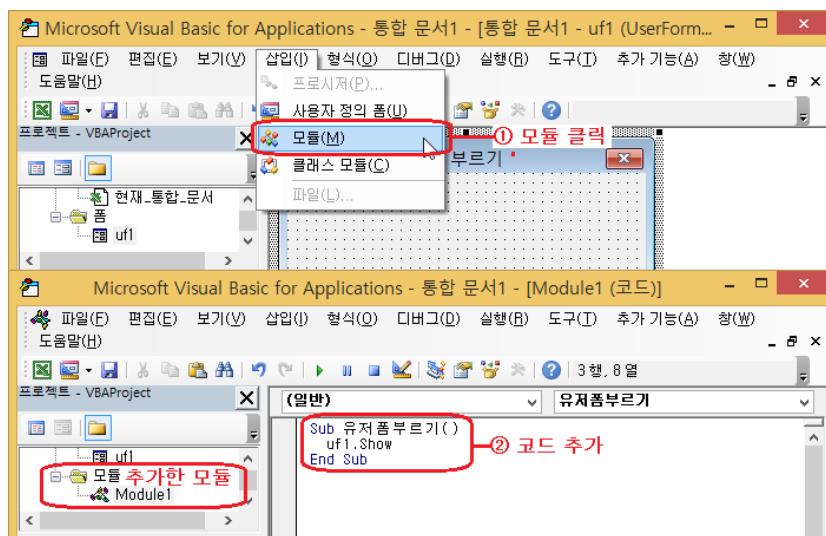


그림 9.1.2: 모듈 추가와 코드 입력

3. 엑셀에서 유저폼 호출은 첫 번째, 개발 도구 → 코드 → 매크로 메뉴를 실행하고 매크로 창에서 실행 버튼을 클릭한다. 두 번째는 엑셀에서 개발 도구 → 컨트롤 → 삽입 → 양식 컨트롤 → 명령 단추를 클릭하고 매크로 지정 창에서 유저폼부르기 매크로를 선택한다. 추가할 명령 단추는 마우스 드래그로 영역을 선택하고 명령 단추 제목을 유저폼 부르기로 변경한다. 유저폼 부르기 명령 단추를 클릭하면 연결된 유저폼 부르기 매크로가 실행된다(그림 9.1.3).

유저폼 속성은 Visual Basic Editor 속성 창에서 간편하게 설정하거나 프로그램 코드 창에서 변경한다. 다음은 사용자 정의 폼 속성 가운데 일부를 소개한다.

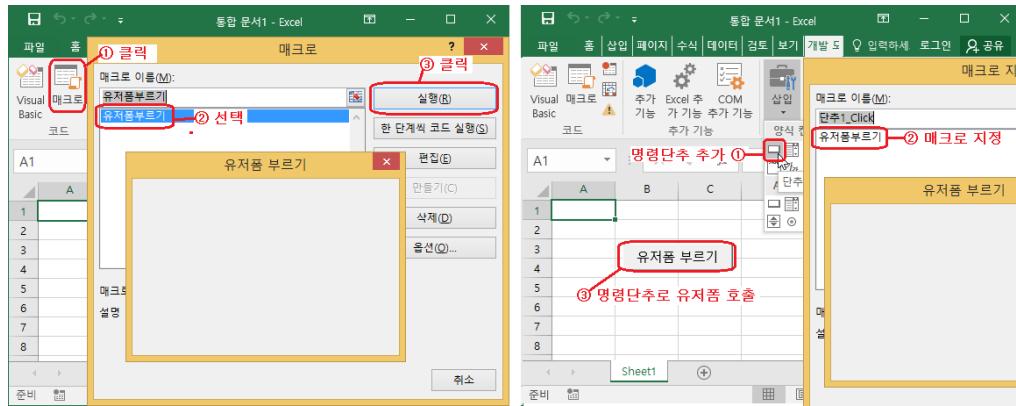


그림 9.1.3: 유저폼 호출하는 두 가지 방법

- ① (이름) : 유저폼 이름이다. 프로그래밍 언어로 표현하면 유저폼 변수명이다.
- ② BackColor : 유저폼 뒷 배경색이다.
- ③ Caption : 유저폼 제목 표시줄 문자열이다.
- ④ Font : 글자 크기, 굵기, 종류이다.
- ⑤ ForeColor : 유저폼 앞 배경색이다.
- ⑥ Height : 유저폼 높이다.
- ⑦ ScrollBars : 유저폼 스크롤바이다.
- ⑧ Width : 유저폼 너비다.

## 제 9.2 절 컨트롤 도구 상자

도구상자는 여러 GUI 컴포넌트가 있다. 이 컴포넌트는 유저폼에 추가하고 이들의 이벤트를 제어할 수 있어 컨트롤 도구상자로 부른다. 이벤트는 버튼



그림 9.2.1: 컨트롤 도구상자

클릭이나 입력창에 글쓰기, 마우스 움직임 같은 컨트롤 행동이다. 그림 9.2.1은 도구상자에 있는 컨트롤이다. 사용자 정의 폼에 컨트롤 추가는 도구상자에서 컨트롤을 마우스로 선택하고 사용자 정의 폼에 마우스 드래그로 컨트롤의 위치와 크기를 정한다.

각 컨트롤은 종류에 이벤트를 발생하는 방법이 다르다. 그 예로 단추 이벤트는 단추를 클릭, 텍스트상자는 글씨를 작성하거나 수정 그리고 삭제, 옵션상자는 여러 옵션에서 그 값을 선택하거나 해제한다.

이벤트가 발생하면 실행하는 프로시저는 사용자 정의 폼에 추가한 컨트롤을 더블 클릭한다. 그 결과 이벤트 프로시저가 코드 창에 프로시저 이름과 속성이 자동으로 만들어진다. 이 프로시저에 이벤트가 발생하면 실행할 명령문을 작성한다.

예를 들어 유저폼에 버튼을 삽입하고 이벤트는 버튼을 누르면 메시지 상자에 “버튼을 눌렀네요.”가 보이게 만들어 보자.

1. 유저폼 추가는 삽입→사용자 정의 폼 메뉴를 선택한다(그림 9.2.2).
2. 유저폼에 명령 단추를 넣는다. 도구상자에서 각 컨트롤 이름은 마우스를 컨트롤에 옮기면 해당 컨트롤 이름이 툴팁에 나타난다. 컨트롤 도구상자에서 에 마우스를 옮기면 툴팁에 명령 단추가 보인다(그림 9.3(a)).  
유저폼에 명령 단추 추가는 컨트롤 도구상자에서 를 누르고 유저폼 엑셀 VBA 툴팁은 특별히 설정한 지역에 마우스가 들어오면 간단한 설명을 보여주는 말풍선이다.

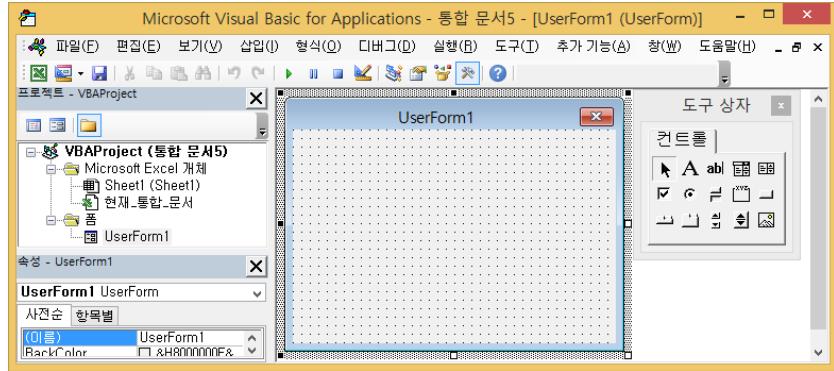


그림 9.2.2: 사용자 정의 폼 추가

에 명령 단추가 추가할 영역을 마우스 드래그로 설정하고(그림 9.3(b)) 마우스를 뗀다(그림 9.3(c)).

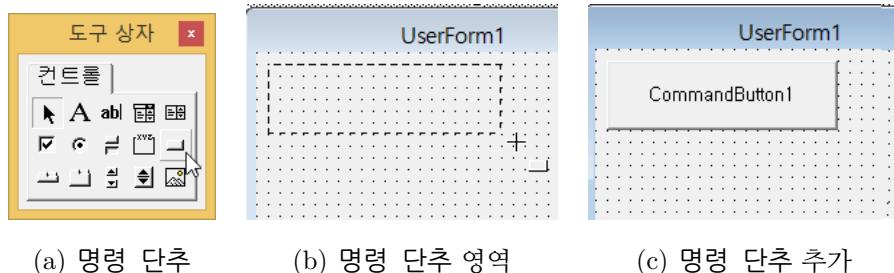


그림 9.2.3: 유저폼에 명령 단추 추가

3. 이벤트 프로시저는 유저폼에 추가한 명령 단추를 더블 클릭하면 자동으로 나타난다(그림 9.2.4). 이 프로시저가 명령 단추 이벤트 프로시저고 기본 이벤트 프로시저는 Click이다. 명령 단추 이벤트 프로시저의 이벤트 종류 변경은 그림 9.2.4에서 를 클릭하여 보이는 프로시저에서 선택한다. 실행 코드는 명령 단추 이벤트 프로시저에 보기 9.2.1을 추가한다. 명령 단추 이름은 속성 창에 CommandButton1이 기본값이고 변경도 가능하다. 이벤트 프로시저 이름은 명령 단추 변수명 + 이벤

트 종류로 정해진다. 보기 9.2.1에서 이벤트 프로시저 이름도 자동으로 `CommandButton1_Click()` 되었다.

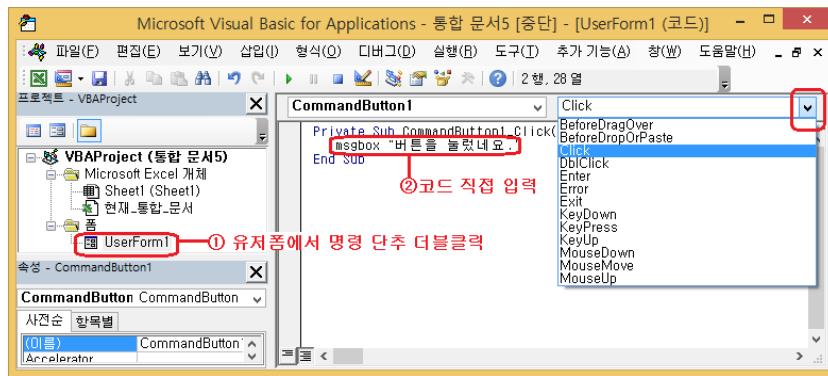


그림 9.2.4: 명령 단추 이벤트 프로시저

### 보기 9.2.1. [명령 단추 이벤트 프로시저]

```

1 Private Sub CommandButton1_Click()
2     MsgBox "버튼을 눌렀네요."
3 End Sub

```

- 표준 모듈은 삽입→모듈 메뉴를 선택하여 추가한다. 유저폼 호출 프로시저는 추가한 표준 모듈에 만든다. 그림 9.2.5는 프로젝트 창에 추가된 품과 모듈을 보여준다. 현재 그림 9.2.5는 모듈을 선택하였고 내용은 왼쪽 코드 창에 보인다. 유저폼의 내용 확인은 UserForm1, 모듈은 Module1을 더블클릭한다. 보기 9.2.2가 유저폼 호출 프로시저이다. 보기 9.2.2의 2 줄은 유저폼을 호출한다. 품 이름은 UserForm1이다. 이 이름도 품 속성 창에서 변경할 수 있다.

### 보기 9.2.2. [표준 모듈에서 유저폼 부르기]

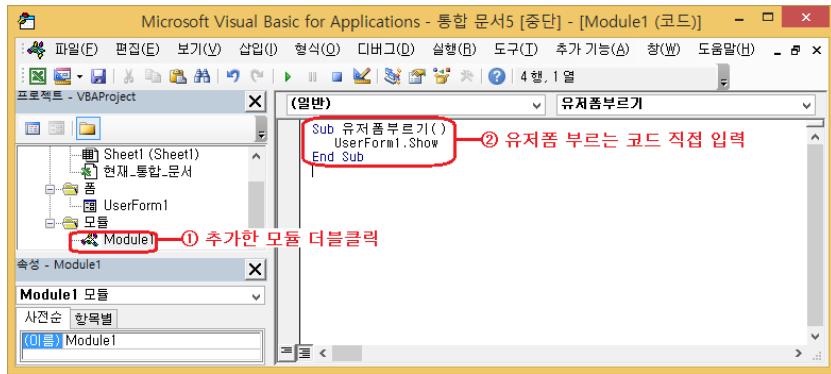


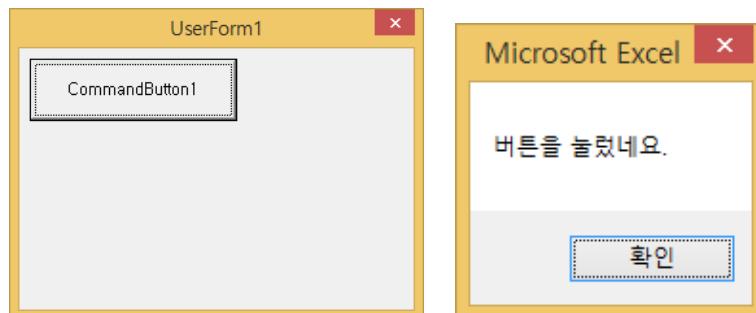
그림 9.2.5: 사용자 정의 폼 추가하고 호출

```

1 Sub 유저폼부르기()
2     UserForm1.Show
3 End Sub

```

5. 유저폼은 엑셀에서 매크로를 호출하여 실행한다. 또 Visual Basic Editor에서 폼을 선택하고 실행버튼(▶)을 눌러도 유저폼이 실행된다. 유저폼과 명령 단추 이벤트 실행결과가 그림 9.2.6에 있다.



(a) 유저폼 실행

(b) 명령 단추 클릭

그림 9.2.6: 유저폼 명령 단추 실행

사용자 정의 폼에 추가하는 컨트롤은 14 개 있으며 다음과 같다. 컨트롤은 필요에 따라 추가도 가능하다.

- ① 개체 선택(): 이것은 컨트롤이 아니고 개체 선택 버튼이다. 선택한 컨트롤을 취소하려면 이 개체 선택을 누른다.
- ② 레이블(Label, ) : 문자를 추가하는 컨트롤
- ③ 텍스트 상자(TextBox, ) : 문자를 입력받아 그 값을 되돌려 주는 컨트롤
- ④ 콤보 상자(ComboBox, ) : 여러 리스트를 추가하고 드랍 다운하여 리스트의 목록을 선택할 수 있는 컨트롤
- ⑤ 목록 상자(ListBox, ) : 여러 리스트를 추가하여 보여지는 리스트의 목록을 선택할 수 있는 컨트롤
- ⑥ 확인란(CheckBox, ) : 여러 체크 박스를 추가하고 다중 선택할 수 있는 컨트롤
- ⑦ 옵션 단추(OptionButton, ) : 옵션 버튼을 추가하는 컨트롤
- ⑧ 토클 단추(ToggleButton, ) : 버튼을 추가하는 컨트롤
- ⑨ 프레임(Frame, ) : 이 컨트롤에 여러 컨트롤을 추가하여 그룹화할 수 있는 컨트롤
- ⑩ 명령 단추(CommandButton, ) : 일반적인 버튼에 대한 컨트롤
- ⑪ 연속탭(TabStrip, ) : 여러 가지 탭을 입력할 수 있는 컨트롤
- ⑫ 다중 페이지(MultiPage, ) : 여러 페이지를 추가할 수 있는 컨트롤
- ⑬ 스크롤 막대(ScrollBar, ) : 스크롤 바를 입력할 수 있는 컨트롤

- 스픈 단추(SpinButton,  ) : 위 또는 아래로 이동할 수 있는 컨트롤
- 이미지(Image,  ) : 그림을 입력할 수 있는 컨트롤

이 있다. 이 컨트롤 추가는 그림 9.1.1 컨트롤 도구 상자에서 컨트롤을 마우스로 클릭하고 유저폼에 마우스 드래그한다.

### 제 9.3 절 사용자 정의 폼 – 홀수, 짝수 판별

사용자 정의 폼으로 만든 응용프로그램을 소개한다. 첫 번째 프로그램은 숫자를 입력하는 텍스트 상자를 만들고 그 값이 홀수인지 짝수인지 판별하여 셀에 출력한다. 프로그램을 만드는 과정은 다음과 같다.

- 먼저 사용자 정의 폼을 추가하고 그림 9.3.1에 있는 속성 가운데 일부는 설명하고 값도 변경한다.

(이름) UserForm1이 유저폼 객체 변수명이다. 매크로에서 이 변수명으로 유저폼을 호출하니 쉽게 변경하는 것이 사용에 편리하다. 이름은 uf로 변경한다.

BackColor 배경색이다. 속성 값은 에 서 버튼을 눌러 변경한다.

Caption 제목표시줄 문자열이다. 한글도 사용 가능하다. 이 속성은 홀수/짝수 판별을 입력한다.

Font 글자체 설정이다. 추가하는 컨트롤에 적용된다.

ForeColor 유저폼에서 적용하는 글자 전경색이다.

Height 유저폼 세로길이다. 직접 값 입력해도 되고 마우스로 창 크기를 위, 아래, 왼쪽, 오른쪽으로 변경하면 이 값이 변한다.

Picture 삽입하는 배경 그림 속성 값이다.



그림 9.3.1: 사용자 정의 폼 속성

Width 유저폼 가로 길이다.

유저폼 속성에서 (값)은 uf로 변경하고 Caption은 홀수/짝수 판별을 입력한다. 속성 값 변경을 적용한 유저폼은 그림 9.3.2에서 확인한다.



그림 9.3.2: 홀수/짝수 판별

2. 숫자 입력할 텍스트 상자를 유저폼에 추가한다. 텍스트 상자 추가는 컨트롤 도구상자에서 **abl**를 누르고 유저폼에서 영역을 마우스 드래그로 그림 9.3.3처럼 정한다. 추가한 텍스트 상자를 마우스로 선택하면 텍스

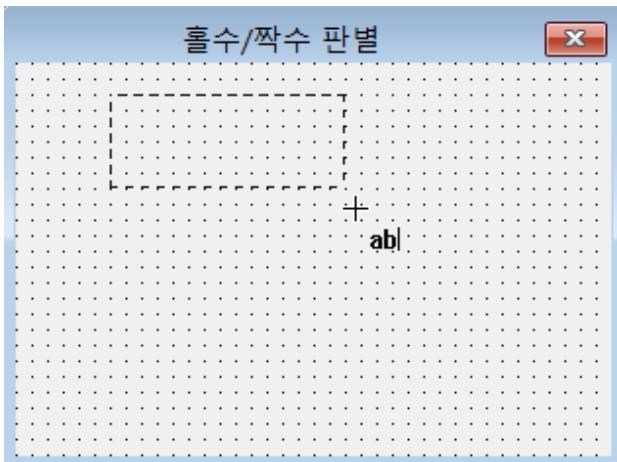


그림 9.3.3: 홀수/짝수 판별

트 상자 속성창이 그림 9.3.4에 나타난다. 다음은 텍스트 상자의 속성 중

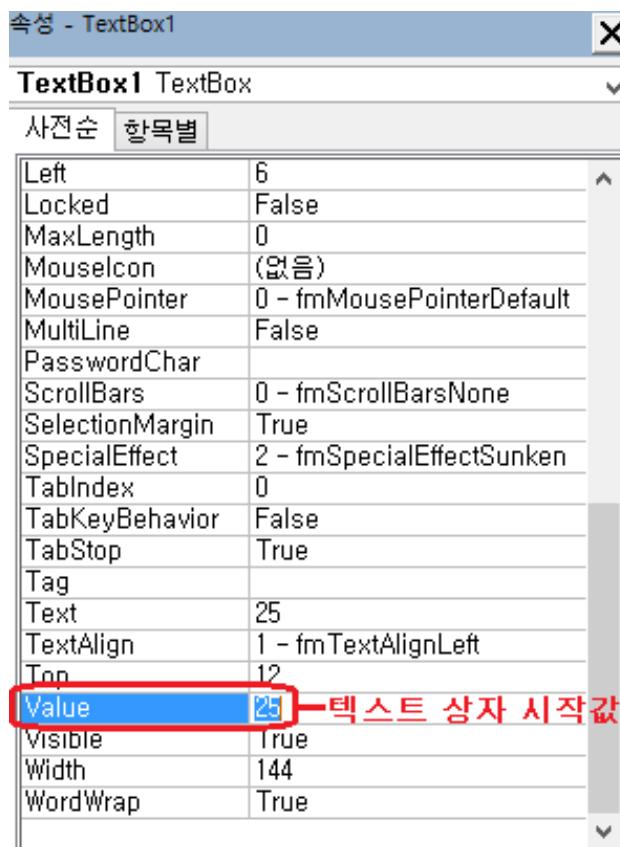


그림 9.3.4: 텍스트 상자 속성

일부만 소개하고 속성값도 변경한다.

(이름) TextBox1이] 텍스트 상자 컨트롤 변수명이다.

AutoTab 키보드에서 탭 키를 눌렀을 때 유저폼에 추가된 컨트롤로 포커스가 이동하도록 설정하거나 해제하는 속성이다.

ControlTipText 마우스가 텍스트 상자에 놓이면 입력한 문자열이 말풍선으로 보인다.

IMEMode Input Method Editor Mode 약어로 입력 문자를 한글과 영어 중 선택한다.

**PasswordChar** 실제 입력한 문자가 보이지 않게 숨겨줄 문자를 설정한다.

**TabIndex** 템을 눌렀을 때 컨트롤 포커스가 옮기게 설정한다. 순서는 0부터 시작한다.

**Text** 유저폼을 시작하면 텍스트 상자에 나타날 문자이다.

**TextAlign** 문자열 정렬 방법을 선택한다.

**Value** 텍스트 상자에 입력한 값이다.

텍스트 상자 속성에서 (이름)은 사용하기에 편리하도록 짧게 tb로 변경 한다. ControlTipText는 숫자를 입력합니다를 입력해 마우스가 텍스트 상자에 위치하면 이 문자가 말풍선에 나온다. Font는 글자체는 굴림, 크기는 18, 굵게 설정한다. Text는 25를 입력하여 시작할 때 초기값으로 사용한다.

3. 유저폼에 명령 단추 추가는 텍스트 상자를 추가하는 방법과 동일하게 진행한다. 명령 단추에서 속성 설정은 Caption에 판별을 입력하여 문자열이 명령 단추에 보이게 한다. 그림 9.3.5는 유저폼에 텍스트 상자와 명령 단추를 추가하고 속성 값을 설정하였다. 유저폼 창 크기도 추가한 컨트롤에 맞게 적절하게 조정하였다. 유저폼에 추가한 명령 단추는

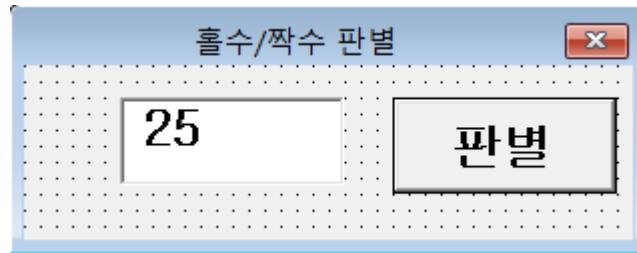


그림 9.3.5: 홀수/짝수 판별

클릭하면 이벤트 프로시저가 코드 창에 자동으로 생긴다. 이 프로시저에

- ◉ 텍스트 상자에 저장된 값을 가져온다.

- ◉ 홀수/짝수를 판별한다.
- ◉ 엑셀 A1 셀에 판별 결과를 출력한다.

내용을 프로그래밍한다. 이 내용은 보기 9.3.1에 있다.

#### 보기 9.3.1. [홀수/짝수 판별]

```

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

```

2 줄 If 조건은 텍스트 상자에 입력된 값을 2로 나누었을 때 나머지가 0이면 참이다. 이런 값은 짝수이다. Mod는 어떤 값을 나누면 몫과 나머지가 있는데 나머지를 구해주는 연산자이다. tb는 텍스트 상자 이름이고, tb.Value는 텍스트 상자에 입력된 값이다. 3 줄은 텍스트 상자 값이 짝수이면 A1 셀에 ○은(는) 짝수, 5 줄은 텍스트 상자 값이 홀수이면 A1 셀에 ○은(는) 홀수가 출력된다.

4. 유저폼을 호출하는 프로시저는 표준 모듈에 작성한다. 모듈 추가는 삽입→모듈 메뉴를 실행한다. 보기 9.3.2는 유저폼을 부르는 프로시저이다. uf는 유저폼 속성에서 설정한 유저폼 이름이다.

#### 보기 9.3.2. [유저폼 부르기]

```

1 Sub 유저폼호출()
2     uf.Show
3 End Sub

```

5. 엑셀에서 유저폼은 매크로를 사용하여 호출한다. 매크로는 개발도구→매크로 메뉴를 실행하여 매크로 창에서 유저폼호출을 실행한다. 그림

9.3.6은 유저폼 호출하고 명령 단추 클릭으로 결과를 엑셀 A1 셀에 출력 한다.

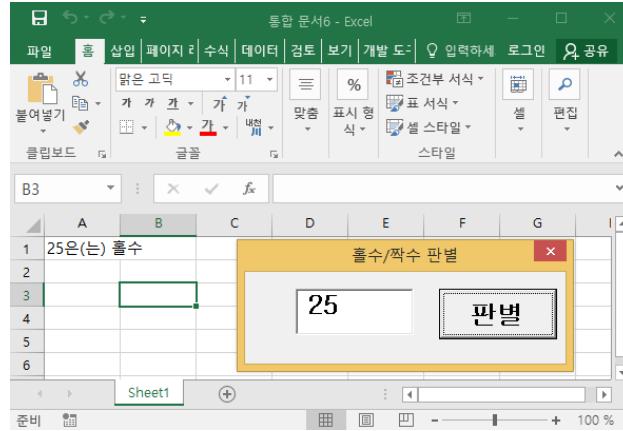


그림 9.3.6: 홀수/짝수 판별

## 제 9.4 절 사용자 정의 폼 – 월드컵 축구 프로그램

우리나라에서 개최했던 2002년 월드컵 축구 8강에 진출했던 팀 승률이 모두 같다고 가정하자. 어떤 팀이 우승할지 모의실험 프로그램을 만들어 보자.

1. 유저폼은 삽입→사용자 정의 폼 메뉴를 실행하여 추가한다. 유저폼 이름은 사용하기 편리하게 uf로 변경한다. 유저폼 제목표시줄은 Caption에 2002년 월드컵을 입력한다. Font 속성은 굵은 글꼴스타일과 크기를 12로 설정한다.
2. 8강 팀은 레이블 (**A**)로 유저폼에 추가한다. 8강 팀은 미국, 독일, 스페인, 한국, 브라질, 잉글랜드, 터키, 세네갈이다. 레이블 이름은 Label1에서 Label18까지 만든 순서에 따라 자동으로 생긴다.
3. 승리팀은 팀은 명령 단추(**■**)를 클릭하여 결정한다. 경기에서 승리팀은 승률이 같다고 가정하고 0에서 1사이 난수를 발생하여 결정한다. 8강

승리팀은 A1에서 A4 셀, 4강 승리팀은 B1에서 B2 셀, 우승팀은 C1셀에 저장한다. 8강전 명령 단추 이름은 사용하기 편리하게 cb1, Caption은 8강전으로 속성 값을 설정한다. 4강전 명령 단추 이름은 cb2, Caption은 4강전, 결승전 명령 단추 이름은 cb3, Caption은 결승전으로 설정한다. 경기를 다시 하기 위하여 모든 과정을 리셋하는 명령 단추를 만든다. 리셋 명령 단추 이름은 cb4, Caption은 다시하기로 설정한다. 유저폼을 설정한 결과가 그림 9.4.1에 있다. 보기 9.4.1은 8강전 명령 단추를 눌렀을

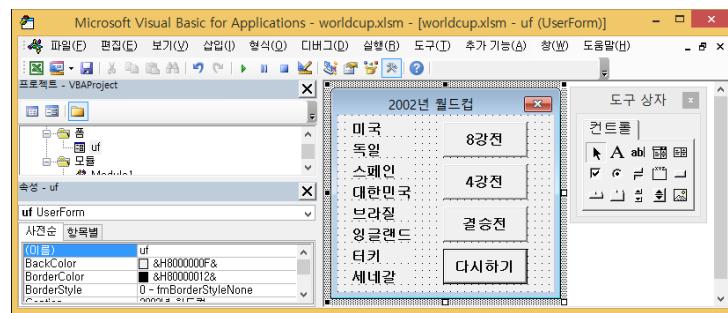


그림 9.4.1: 유저폼 부르기

때 발생하는 이벤트 프로시저이다.

#### 보기 9.4.1. [2002년 월드컵 8강전]

```

1 Private Sub cb1_Click()
2     If Rnd > 0.5 Then
3         Range("A" & 1) = Label1.Caption
4     Else
5         Range("A" & 1) = Label2.Caption
6     End If
7
8     If Rnd > 0.5 Then
9         Range("A" & 2) = Label3.Caption
10    Else
11        Range("A" & 2) = Label4.Caption
12    End If

```

```

13
14     If Rnd > 0.5 Then
15         Range("A" & 3) = Label5.Caption
16     Else
17         Range("A" & 3) = Label6.Caption
18     End If
19
20     If Rnd > 0.5 Then
21         Range("A" & 4) = Label7.Caption
22     Else
23         Range("A" & 4) = Label8.Caption
24     End If
25     cb1.Visible = False
26 End Sub

```

이 프로시저는 8강전에서 이기는 팀을 결정한다. 이벤트 프로시저는 유저폼에서 8강전 명령 단추를 누르면 자동으로 만들어진다. 2 – 6 줄은 미국과 독일에서 승리팀을 결정한다. 승리팀은 2줄 Rnd값이 0.5보다 크면 미국이고 그렇지 않으면 독일이다. Rnd는 0에서 1사이에서 난수를 발생하는 함수이다. 그러니 이 값이 0.5보다 클 확률은 0.5다. 3 줄은 Label1.Caption 속성 값 미국을 A1 셀에 저장한다. 5 줄은 Label2.Caption 속성 값 독일을 A1 셀에 저장한다. 25 줄은 8강전 승리팀이 모두 결정되면 8강전 명령 버튼을 보이지 않게하여 설정한다. cb1.Visible 속성은 명령 단추를 보여줄지 설정한다. 보기 9.4.2는 4강전 승리팀을 결정한다. 8강전 승리팀이 저장된 A1에서 A4 셀에서 승리팀을 결정하여 B1에서 B2 셀에 저장한다.

#### 보기 9.4.2. [2002년 월드컵 4강전]

```

1 Private Sub cb2_Click()
2     If Rnd > 0.5 Then
3         Range("B" & 1) = Range("A1").Value
4     Else
5         Range("B" & 1) = Range("A2").Value

```

```

6   End If
7
8   If Rnd > 0.5 Then
9       Range("B" & 2) = Range("A3").Value
10  Else
11      Range("B" & 2) = Range("A4").Value
12  End If
13  cb2.Visible = False
14 End Sub

```

13 줄은 4강전 승리팀이 결정되면 4강전 명령 버튼을 보이지 않게하여 설정한다. cb2.Visible = False는 4강전 명령 단추를 보이지 않게 설정한다. 보기 9.4.3은 결승전 승리팀을 결정한다. 4강전 승리팀이 저장된 B1과 B2 셀에서 우승팀을 결정하여 C1 셀에 저장한다.

#### 보기 9.4.3. [2002년 월드컵 결승전]

```

1 Private Sub cb3_Click()
2     If Rnd > 0.5 Then
3         Range("C" & 1) = Range("B1").Value
4     Else
5         Range("C" & 1) = Range("B2").Value
6     End If
7     cb3.Visible = False
8 End Sub

```

7 줄은 결승전 우승팀이 결정되면 결승전 명령 버튼을 보이지 않게하여 설정한다. cb3.Visible = False는 결승전 명령 단추를 보이지 않게 설정한다. 보기 9.4.4는 결승전까지 보이지 않았던 명령 단추가 보이게 한다.

#### 보기 9.4.4. [2002년 월드컵 재경기 설정]

```

1 Private Sub cb4_Click()
2     cb1.Visible = True ' 8강전 버튼

```

```

3     cb2.Visible = True ' 4강전 버튼
4     cb3.Visible = True ' 결승전 버튼
5 End Sub

```

4. 표준 모듈을 삽입→모듈 메뉴 실행하여 추가한다. 보기 9.4.5는 유저폼을 호출하는 프로시저이다.

#### 보기 9.4.5. [2002년 월드컵 유저폼 호출]

```

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

```

5. 엑셀에서 유저폼은 매크로를 실행하여 호출한다. 매크로는 개발도구→매크로 메뉴를 실행하여 매크로 창에서 유저폼호출을 실행한다. 또 매크로에 연결할 단추는 개발도구→삽입→양식컨트롤→단추 메뉴를 실행하고 시트에 삽입할 영역을 마우스 드래그하여 추가한다. 추가한 단추에 매크로 연결은 단추에서 마우스 오른쪽 클릭하여 매크로 지정에서 한다. 그럼 9.4.2는 월드컵 8강 단추 클릭으로 유저폼을 호출한다. 그리고 게임을 진행하면 결과가 엑셀 시트에 출력된다.

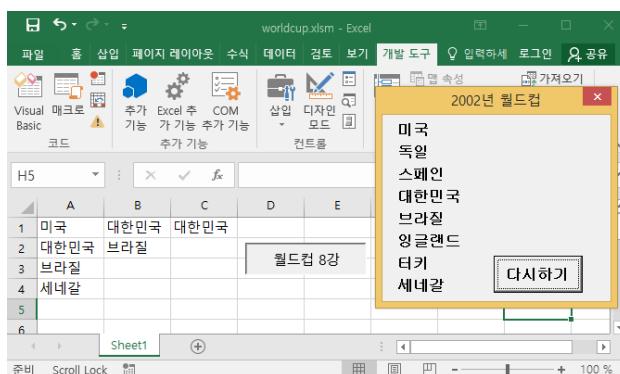


그림 9.4.2: 2002년 월드컵 게임 결과

## 제 9.5 절 사용자 정의 폼 – 문자 입·출력 프로그램

다음은 텍스트 상자에 문자를 입력하고 셀과 유저폼에 출력하는 프로그램을 만든다. 완성한 프로그램은 그림 9.5.1에 있다. 9.5.1에 추가한 콘트롤은 다음과

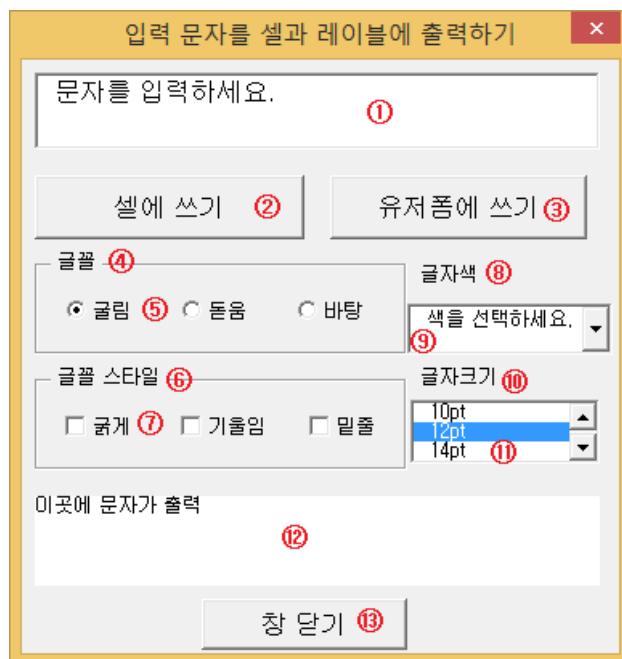


그림 9.5.1: 텍스트 상자 문자열 출력

같다.

- ◉ ①은 문자를 입력하는 텍스트 상자,
- ◉ ②는 텍스트 상자에 있는 문자를 셀에 쓰는 명령 단추,
- ◉ ③은 텍스트 상자에 있는 문자를 유저폼 ⑫ 레이블에 쓰는 명령 단추,
- ◉ ④는 글꼴 종류를 하나로 묶는 프레임,
- ◉ ⑤는 글꼴 종류를 나타내는 3개 옵션 단추,

- ⑥은 글꼴 스타일을 하나로 묶는 프레임,
- ⑦은 글꼴 스타일을 나타내는 3개 확인란,
- ⑧은 ⑨ 컨트롤의 용도를 알리는 레이블,
- ⑩은 색 종류를 담고 있는 콤보 상자,
- ⑪은 컨트롤의 용도를 알리는 레이블,
- ⑫은 글자 크기를 담고 있는 목록 상자,
- ⑬는 텍스트 상자에 있는 문자가 명령 단추 ③을 눌렀을 때 출력하는 레이블
- ⑭는 유저폼을 닫는 명령 단추이다.

다음은 코드 설명이다. 그림 9.5.1은 모듈 전체에서 사용하는 변수를 선언한다. 설명은 해당 변수에 값을 배정하는 부분에서 한다.

#### 보기 9.5.1. [문자 입력 및 출력 – 변수 선언]

```

1 Dim myFontSize As Integer
2 Dim myForeColor As Long
3 Dim outRange As Range

```

보기 9.5.2는 확인란 ⑦에서 이벤트 프로시저로 글꼴 스타일을 설정하면 처리할 문을 코딩한다. 1 - 3 줄은 글꼴 스타일에서 굵게를 선택하면 실행하는 문이다. 2 줄은 텍스트 상자의 글꼴 스타일을 굵게 한다. 5 - 7 줄은 글꼴 스타일에서 기울임을 선택하면 실행하는 문이다. 6 줄은 텍스트 상자 문자를 기울게 한다. 9 - 11 줄은 글꼴 스타일에서 밑줄을 선택하면 실행하는 문이다. 10 줄은 텍스트 상자 문자에 밑줄을 긋는다.

#### 보기 9.5.2. [문자 입력 및 출력 – 확인란]

```

1 Private Sub CheckBox1_Click()
2     TextBox1.Font.Bold = True
3 End Sub
4
5 Private Sub CheckBox2_Click()
6     TextBox1.Font.Italic = True
7 End Sub
8
9 Private Sub CheckBox3_Click()
10    TextBox1.Font.Underline = True
11 End Sub

```

보기 9.5.3은 콤보 상자 ⑨에서 색을 선택하면 실행하는 이벤트 프로시저이다. Switch 함수는 조건이 맞으면 설정한 값을 리턴한다. 4 줄은 ComboBox1.Value 가 검정색이면 vbBlack 값을 가져온다. vbBlack은 VBA에서 선언한 상수이다. 이 값은 색에서 사용하면 검정색이다.

### 보기 9.5.3. [문자 입력 및 출력 – 콤보 상자]

```

1 Private Sub ComboBox1_Change()
2     myForeColor = Switch(ComboBox1.Value = "색을 선택하세요.", _
3                           vbBlack, _
4                           ComboBox1.Value = "검정색", vbBlack, _
5                           ComboBox1.Value = "빨간색", vbRed, _
6                           ComboBox1.Value = "녹색", vbGreen, _
7                           ComboBox1.Value = "파란색", vbBlue, _
8                           ComboBox1.Value = "노란색", vbYellow, _
9                           ComboBox1.Value = "자홍색", vbMagenta, _
10                          ComboBox1.Value = "하늘색", vbCyan)
11     TextBox1.ForeColor = myForeColor
12 End Sub

```

보기 9.5.4는 명령 단추 ②를 눌렀을 때 실행문이다. 2 - 3 줄은 InputBox 함수로 Type:=8은 입력값이 Range이다. 그래서 배정하는 값도 보기 9.5.1에서 선언한 변수이다. 4 – 11 줄은 셀에 저장하는 문자 글꼴과 글꼴 스타일

설정이다.

#### 보기 9.5.4. [문자 입력 및 출력 – 명령 단추]

```

1 Private Sub CommandButton1_Click()
2     Set outRange = Application.InputBox("값을 출력할 셀 선택", _
3                                         Type:=8)
4     With outRange
5         .Value = TextBox1.Value
6         .Font.Bold = CheckBox1.Value
7         .Font.Italic = CheckBox2.Value
8         .Font.Underline = CheckBox3.Value
9         .Font.Size = myFontSize
10        .Font.Color = myForeColor
11    End With
12 End Sub

```

보기 9.5.5는 명령 단추 ③을 눌렀을 때 실행문이다. 3 줄은 텍스트 상자 ① 값을 가져오고 나머지 줄은 글꼴과 글꼴 스타일 설정이다.

#### 보기 9.5.5. [문자 입력 및 출력 – 명령 단추]

```

1 Private Sub CommandButton2_Click()
2     Label3.Caption = TextBox1.Value
3     Label3.Font.Size = myFontSize
4     Label3.ForeColor = myForeColor
5     Label3.Font.Bold = True
6     Label3.Font.Italic = CheckBox2.Value
7     Label3.Font.Underline = CheckBox3.Value
8 End Sub

```

보기 9.5.6은 명령 단추 ⑩을 누르면 유저폼이 닫힌다.

#### 보기 9.5.6. [문자 입력 및 출력 – 명령 단추]

```

1 Private Sub CommandButton3_Click()
2     Unload uf
3 End Sub

```

보기 9.5.7은 목록 상자 ⑪에 글자 크기를 설정하였다. 2 줄 `ListBox1.ListIndex`는 0부터 시작한다. 실제 저장된 값은 1부터 시작하기에 1을 더한 것이다. 4 줄은 텍스트 상자 글자 크기를 설정한다.

#### 보기 9.5.7. [문자 입력 및 출력 – 목록 상자]

```

1 Private Sub ListBox1_Click()
2     myFontSize = Choose(ListBox1.ListIndex + 1, 10, 12, 14, _
3                             16, 18, 20)
4     TextBox1.Font.Size = myFontSize
5 End Sub

```

보기 9.5.8은 글꼴 ⑤에 설정 값을 텍스트 상자에 적용하였다.

#### 보기 9.5.8. [문자 입력 및 출력 – 텍스트 단추]

```

1 Private Sub OptionButton1_Click()
2     TextBox1.Font.Name = "굴림"
3 End Sub
4
5 Private Sub OptionButton2_Click()
6     TextBox1.Font.Name = "돋움"
7 End Sub
8
9 Private Sub OptionButton3_Click()
10    TextBox1.Font.Name = "바탕"
11 End Sub

```

보기 9.5.9는 유저폼이 시작할 때 실행한다. 이 이벤트 프로시저는 유저폼을 더블 클릭하고 그림 9.5.9에서 ②를 누르고 이벤트 프로시저 `Initialize`를 선택한다. 유저폼 이름은 uf로 설정하였다. 이 프로시저는 목록 상자나 확인란은 설정 값을 자동으로 불러오지 않으므로 여기서 초기 시작값을 설정한다. 2 줄은 텍스트 상자 ①이 처음 시작할 때 보이는 문자열이다. 3 – 4 줄은 콤보상자 ⑨에 색 종류를 설정하였다. 5 줄은 콤보 상자 ⑨가 처음 시작할 때 보여주는 문자열이다. 6 – 7 줄은 목록 상자 ⑪에 글자 크기를 설정하였다. 8 줄은 목록 상자 ⑪이 처음 시작할 때 설정 값은 10pt이다.

### 보기 9.5.9. [문자 입력 및 출력 – 콤보 상자]

```

1 Private Sub UserForm_Initialize()
2     TextBox1.Value = "문자를 입력하세요."
3     ComboBox1.List = Array("검정색", "빨간색", "녹색", _
4                             "파란색", "노란색", "자홍색", "하늘색")
5     ComboBox1.Value = "색을 선택하세요."
6     ListBox1.List = Array("10pt", "12pt", "14pt", _
7                           "16pt", "18pt", "20pt")
8     ListBox1.ListIndex = 1
9 End Sub

```

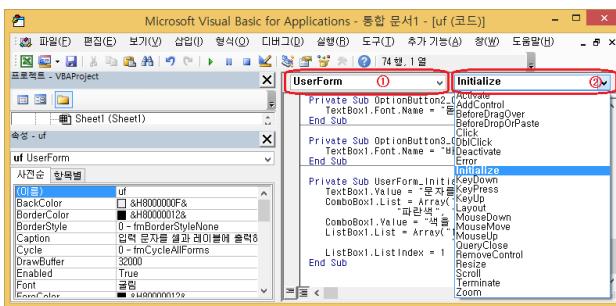


그림 9.5.2: 텍스트 상자 문자열 출력 – 유저폼

# 제 10 장

## 비주얼 베이직 입문

### 제 10.1 절 Visual Basic 시작하기

비주얼 베이직 시작은 개발 도구→코드→Visual Basic 메뉴를 선택하여 비주얼 베이직 에디터 화면(그림 10.1.1)이 나타나도록 한다. 비주얼 베이직 프

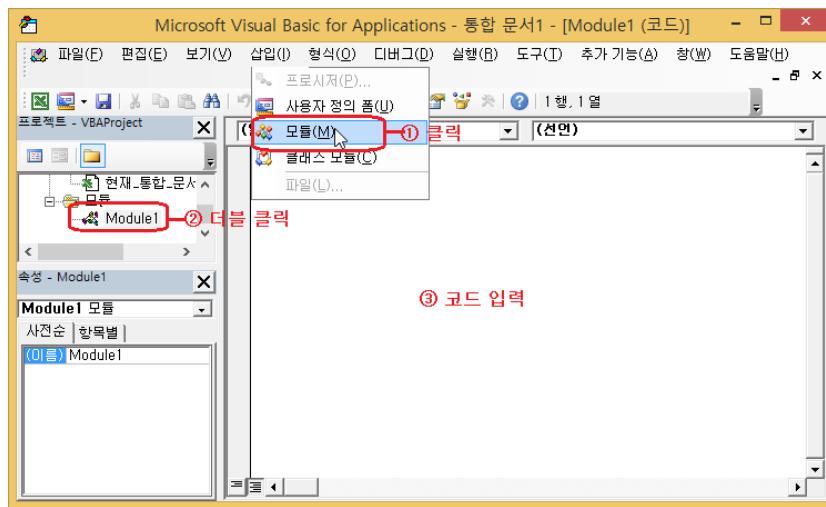


그림 10.1.1: Visual Basic Editor(2016)

로그램 작성은 그림 10.1.1의 프로젝트 창에서 삽입→모듈 메뉴 ①을 클릭하여 모듈을 추가한다. 그리고 그림 10.1.1에 추가된 모듈 ②를 더블 클릭하면 왼쪽에 코드 입력창 ③이 보이며 코드를 입력하여 프로그램을 작성한다.

## 제 10.2 절 Visual Basic 만들고 실행하기

비주얼 베이직 프로그램을 만들고 실행해 보자. 보기 10.2.1은 코드 입력창에 Sub 창나오기를 입력하고 엔터 키를 누르면 End Sub 코드가 자동으로 만들어 진다.

### 보기 10.2.1. [Sub 문]

```
1 Sub 창나오기()
2
3 End Sub
```

보기 10.2.2는 Sub와 End Sub 사이에 실행할 코드를 작성하였다.

### 보기 10.2.2. [Sub 문 작성]

```
1 Sub 창나오기()
2     MsgBox "안녕하세요"
3 End Sub
```

MsgBox는 비주얼 베이직 함수로 간단한 메시지를 팝업창에 제공한다. 그림 10.2.1은 보기 10.2.2를 실행한 결과이다. 작성한 비주얼 베이직 프로그램은 실행, 중단, 변경된 내용을 재설정 메뉴로 실행과정을 조절한다.

- Sub/사용자 정의 품 실행 또는 매크로 실행(▶) : 사용자 정의 품이나 매크로를 실행한다.
- 중단(■) : 실행중인 프로그램을 중지한다. 프로그램 오류가 있으면 중단 버튼을 누르고 수정한다.
- 재설정(□) : 프로그램을 수정하고 변경한 내용 적용은 재설정 버튼을 누른다.

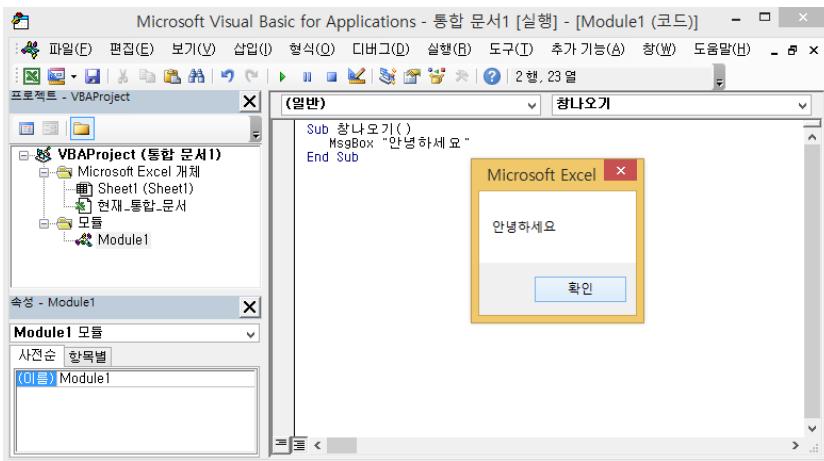


그림 10.2.1: 간단한 VBA 작성과 실행

### 제 10.3 절 Visual Basic 구성

비주얼 베이직은 이미 만들어진 개체나 메소드, 함수가 다양하고 많아 이것들을 잘 활용하여 프로그램을 작성한다. 따라서 비주얼 베이직을 잘 사용하려면 이미 만들어진 것은 무엇이 있는지 많이 알아야 한다.

비주얼 베이직 작성은 모듈이나 클래스 모듈, 사용자 정의 폼 안에 한다. 이것들이 모여서 프로젝트가 되고 프로그램에서 가장 큰 단위이다. 사용자 정의 폼은 유저폼으로도 부른다.

- ① **프로젝트** : 모듈, 클래스 모듈, 사용자 정의 폼, Microsoft Excel 개체를 포함하는 최상위 항목이다.
- ② **모듈** : 매크로를 만들고 실행하는 곳이다.
- ③ **클래스 모듈** : 개체 제공을 목적으로 작성한 모듈이다. 모듈에서 이것들을 참조하여 사용한다.
- ④ **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.

다음 큰 단위는 프로시저로 종류는 Sub, Function, Property가 있다. 프로시저는 여러 가지 개체, 함수, 메소드, 키워드를 사용하여 비주얼 베이직 구문에 맞게 프로그램을 작성한다.

- 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.

#### 제 10.4 절 엑셀과 관련된 개체, 메소드, 속성

비주얼 베이직에서 엑셀과 관련된 것을 알아보자. 엑셀 파일은 여러 시트를 통합하여 만든다. 그리고 시트는 여러 셀을 모아 만든다. VBA에 엑셀 파일은 Workbook 시트는 Worksheet 셀은 Cell로 이름지었다. 개체(object)는 이미 만들어진 Workbook, Worksheet, Cell이다. 이 개체는 메소드(method), 속성(property), 이벤트(event)를 갖는다.

- 속성은 개체 특성(characteristics)으로 셀 높이, 셀 너비, 셀 배경색을 측정하거나 설정한다.
- 메소드는 개체 행동(action)으로 종류는 셀 선택, 셀 복사, 셀 붙여넣기가 있다.
- 이벤트는 VBA 컴포넌트가 동작하는 것으로 버튼을 누르거나 리스트, 확인란, 옵션 단추, 콤보 상자에서 목록을 선택한다.

컴포넌트는 VBA  
에서 컨트롤로 부  
른다.

HEEMO KANG

엑셀 VBA

### 10.4.1 통합문서 개체

엑셀 파일은 통합문서라고 부르며 개체명은 `Workbooks`이다. 개체는 메소드, 함수, 상수를 포함하고 이것을 사용하여 프로그램을 작성한다. 다음은 통합문서 개체 사용법이다.

#### 사용방법 10.4.1. 【통합문서】

```
Workbooks  
Workbooks(index)  
Workbooks(filename)
```

이 개체에 사용한 인수들은

- ◉ *index* : 엑셀 파일이 열린 순서로 1부터 사용한다.
- ◉ *filename* : 통합문서 파일명이다. 한 파일만 열려 있으면 입력하지 않아도 된다.

보기 10.4.1은 VBA로 새로운 통합문서를 만들었다.

#### 보기 10.4.1. 【통합문서 만들기】

```
1 Sub 통합문서추가()  
2     Workbooks.Add  
3 End Sub
```

`Workbooks` 개체와 `Add` 메소드로 새로운 통합문서를 만든다. 개체와 메소드 사용방법은

**개체.메소드**

이고 개체와 속성 사용방법은

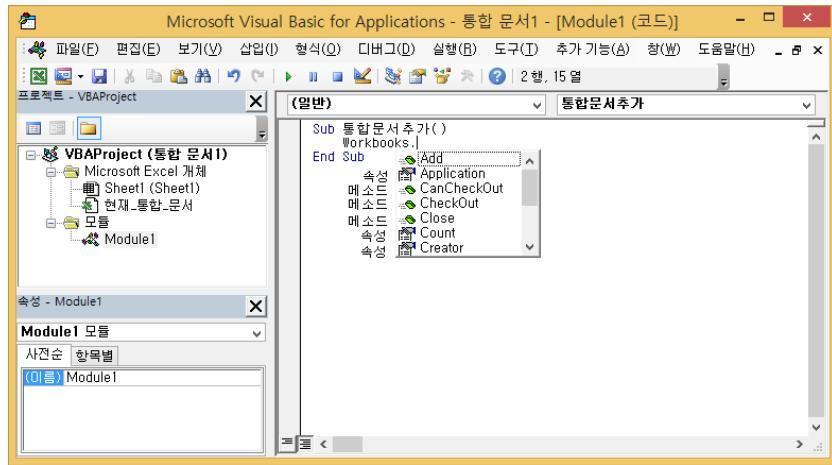


그림 10.4.1: 통합문서 추가

개체.속성 = 값

이다. 개체는 메소드와 속성보다 상위 계층이다. 상위 계층인 개체를 입력하고 점(.)을 입력하면 그림 10.4.1과 같이 해당 개체에서 사용가능한 메소드와 속성들이 나타난다. 메소드와 속성 구별은 그림 10.4.1에서 보는 것처럼 메서드와 속성 이름 앞에 있는 그림모양으로 구별한다. 속성이나 메소드 선택은 화살표 키로 사용하려는 속성이나 메소드로 옮기고 탭을 누른다. 여기서 .은 개체와 메스드나 속성을 연결하는 연산자이다.

보기 10.4.2는 `Workbooks` 개체와 `Open` 메소드를 . 연산자로 연결하여 이미 존재하는 엑셀파일을 연다.

#### 보기 10.4.2. 【통합문서 열기】

```

1 Sub 통합문서열기()
2     Workbooks.Open("C:\temp\MyExcel.xlsx")
3 End Sub

```

`Workbooks` 개체는 엑셀 파일이고 `Open` 메소드는 C:\temp 폴더 안에 있는 MyExcel.xlsx 엑셀 파일을 연다.

보기 10.4.3은 열은 엑셀 파일 중에서 특정한 엑셀 파일을 활성화한다. 열려진 엑셀 파일에서 첫 번째 파일을 활성화한다.

#### 보기 10.4.3. 【통합문서 활성화】

```

1 Sub 통합문서활성화()
2     Workbooks(1).Activate
3 End Sub

```

보기 10.4.4는 엑셀 파일을 닫는다.

#### 보기 10.4.4. 【통합문서 닫기】

```

1 Sub 통합문서닫기()
2     Workbooks.close
3 End Sub

```

`Workbooks` 개체와 `Close` 메소드로 통합문서를 닫는다. 닫을 통합문서를 지정하지 않으면 현재 활성화 상태인 엑셀파일이 닫힌다.

속성/메소드	내용
Add	새로운 통합문서를 만든다.
Close	통합문서를 닫는다.
Open	통합문서를 연다.
Count	열려있는 통합문서 갯수를 알려준다.
Save	통합문서를 저장한다.
SaveAs	새이름으로 통합문서를 저장한다.
Item	특정한 통합문서를 선택한다.

표 10.4.1: `Workbooks` 개체의 속성 및 메소드

### 10.4.2 시트 개체

시트는 Worksheets나 Sheets 개체를 사용한다. 사용 방법은

#### 사용방법 10.4.2. [시트]

```
Worksheets(index)
Worksheets(sheet_name)
Sheets(sheet_name)
```

이다. 여기서 팔호() 안에 들어갈 인수는

- ◉ *index* : 참조(인덱스) 번호
- ◉ *sheet\_name* : 시트 이름

이다. 보기 10.4.5는 시트 개체에서 사용하는 속성들이다.

#### 보기 10.4.5. [시트 참조]

```
1 Sub 시트참조1()
2     Worksheets(1).Activate
3 end Sub
4 Sub 시트참조2()
5     Worksheets("Sheets3").Activate
6 end Sub
7 Sub 시트참조3()
8     Sheets(1).Activate
9 end Sub
10 Sub 시트참조4()
11     Sheets("Sheets3").Activate
12 end Sub
13 Sub 시트참조5()
14     Worksheets("Sheet3").Select
15     Worksheets.Add after:=ActiveSheet
16     ActiveSheet.Name = "새로만든시트"
17     MsgBox Worksheets.Count
18 End Sub
```

시트참조1과 시트참조3 프로시저는 시트 번호로 해당 시트를 활성화하였고 시트참조2와 시트참조4 프로시저는 시트 이름으로 해당 시트를 활성화하였다. 14 줄은 Sheet3 시트를 활성화, 15 줄은 시트를 활성화 시트 다음에 추가, 16 줄은 활성화 시트명을 새로만든시트, 17 줄은 시트의 개수를 알려준다.

속성/메소드	내용
Add	새로운 시트를 만든다.
Move	시트를 옮긴다.
Copy	시트를 복사한다.
Delete	시트를 삭제한다.
Select	시트를 선택한다.
Count	시트 개수를 알려준다.
Item	특정한 시트를 선택한다.
Name	시트 이름을 변경한다.

표 10.4.2: Worksheets 개체의 속성 및 메소드

### 10.4.3 셀 개체

셀 개체는

- ① Cells
- ② Range
- ③ Offset

이 있다.

#### Range 개체

Range 개체는

### 사용방법 10.4.3. 【Range】

```
Range("ref")
```

이며

- ④ ref는 셀을 A1, R1C1 스타일로 입력하며, 여러 셀도 지정할 수 있다.

셀 범위 설정은

```
1 Range("B1").Select 'B1 셀 선택
2 Range("B2:F5").Select 'B2:F5 셀 선택
3 Range("B3:D4,B6:F8").Select 'B3:D4과 B6:F8 셀 선택
```

로 한다. 열 범위는

```
1 Range("A:A").Select 'A열 선택
2 Range("A:D").Select 'A열에서 D열까지 선택
3 Range("A:A,F:F").Select 'A열과 F열 선택
```

한다. 행 범위는

```
1 Range("1:1").Select '1행 선택
2 Range("1:3").Select '1행에서 3행까지 선택
3 Range("1:1,3:3").Select '1행과 3행 선택
```

한다. 셀에 값 입력은

### 보기 10.4.6. 【Range 개체로 셀 값 채우기】

```
1 Sub RangeExample()
2     Range("A1").Value = "엑셀"
3     Range("A2") = "비주얼 베이직"
4     Range("B1:F6") = "Excel & Visual Basic"
5 End Sub
```

이다. Value 속성은 생략이 가능하다. 그럼 10.4.2는 셀에 값을 입력한 보기 10.4.6의 결과이다.

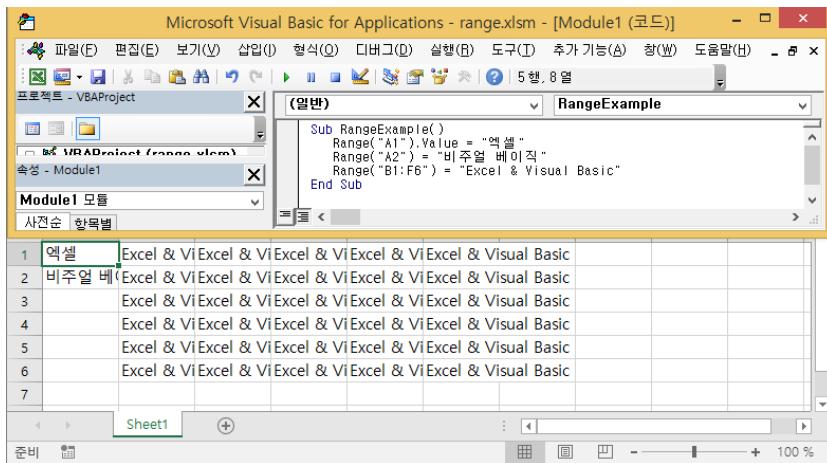


그림 10.4.2: 셀에 값 입력

### Font 개체

Font 개체는 글자 속성을 설정한다. 이 개체로 셀의 글자 속성을 변경한다. 다음 보기는 Font 개체로 셀 글자체를 설정한다.

속성/메소드	내용
Name	글자체를 설정한다.
Size	글자체 크기이다.
Bold	글자의 굵기이다.
Color	글자색이다. RGB 함수로 값을 설정한다.
ColorIndex	글자색이다. 이미 설정된 특정한 색을 인덱스로 가져온다.
Italic	글자의 기울임체 설정이다.

표 10.4.3: Font 개체의 속성 및 메소드

### 보기 10.4.7. [Font 개체로 셀 글자체 설정]

```

1 Sub 글자체설정()
2     Range("B2").Value = 123

```

```

3   Range("B3").Value = "엑셀"
4
5   Range("B2").Font.Name = "굴림체"
6   Range("B2").Font.Size = 11
7   Range("B2").Font.Bold = True
8   Range("B2").Font.ColorIndex = 5
9
10  With Range("B3").Font
11    .Name = "함초롬돋움"
12    .Size = 13
13    .Bold = True
14    .Italic = True
15    .Color = RGB(255, 0, 25)
16  End With
17 End Sub

```

그림 10.4.3은 셀에 글자체를 설정한 보기 10.4.7 결과이다. 5 – 8 줄과 10 – 15 줄은 동일한 내용이다. 5 – 8 줄은 개체와 속성을 모두 적었고, 10 – 15 줄은 동일한 내용을 With 문으로 생략하였다. With 문 사용방법은 351 페이지 11.7.4 절에 있다. 8 줄 ColorIndex는 정해진 색 번호를 붙인 것으로 표 10.4.4에 일부를 소개한다. 16 줄은 Color속성에 색 설정을 RGB 함수로 적

인덱스 번호	색	인덱스 번호	색
1	검정	2	흰색
3	빨강	4	녹색
5	파랑	6	노랑
7	자홍	8	하늘색

표 10.4.4: ColorIndex 색

용하였다. RGB 함수는 빛의 삼원색인 빨강(Red), 녹색(Green), 파랑(Blue)을 적절하게 혼합하여 색을 표현한다. 색 표현 강도는 0에서 255 사이 정수값이다. 15 줄은 빨강과 파랑을 최대 강도 255로 설정하여 혼합한 자홍색(Magenta)

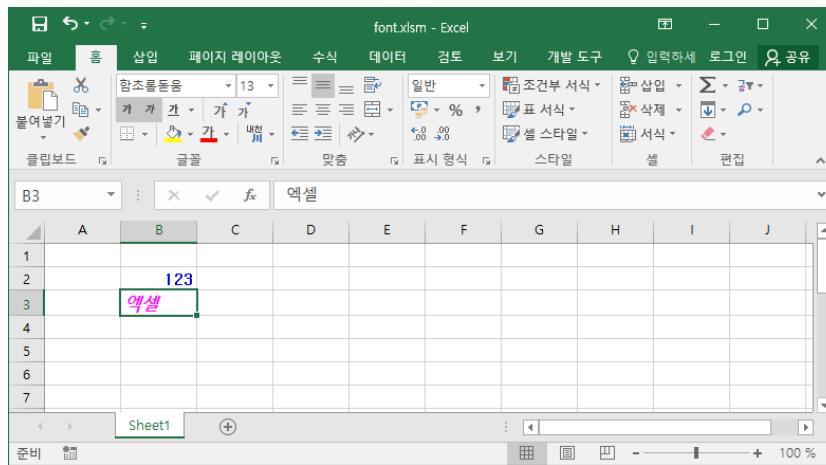


그림 10.4.3: 셀에 글자체 설정

이다. 개인용 컴퓨터에서 사용이 가능한 글자체는 다를 수 있다. 현재 사용하는 컴퓨터에서 사용가능한 글자체는 메모장 실행하고 “설정→글꼴” 메뉴에서 확인한다. 엑셀에서 셀 속성은 표시형식, 맞춤, 글꼴, 테두리, 채우기, 보호가 있다. 글꼴은 앞에서 소개하였고 맞춤, 테두리, 채우기를 계속하여 소개한다.

#### 보기 10.4.8. [셀 속성 – 셀 패턴]

```

1 Sub 셀패턴()
2     With Range("B4:B6")
3         .Interior.ColorIndex = 6
4         .Interior.Pattern = xlPatternCrissCross
5     End With
6 End Sub

```

보기 10.4.8에서 2 줄은 셀 설정 영역이 B4:B6이고, 3 줄은 선택한 셀 배경 색, 4 줄은 셀 채우기에서 무늬 스타일이다.

#### 보기 10.4.9. [셀 속성 – 셀 맞춤]

```

1 Sub 셀맞춤()
2     Range("B2").Value = "Excel VBA"

```

```

3   With Range("B2")
4     .Interior.ColorIndex = 6
5     .ColumnWidth = 25
6     .RowHeight = 25
7     .HorizontalAlignment = xlHAlignCenter
8     .VerticalAlignment = xlVAlignCenter
9   End With
10 End Sub

```

보기 10.4.9에서 4 줄은 B2 셀 배경색을 노란색, 5 줄은 셀 너비를 25, 6 줄은 셀 높이를 25, 7 줄은 셀 가로 정렬을 가운데, 8 줄은 셀 세로 정렬이 가운데이다.

#### 보기 10.4.10. [셀 속성 – 셀 테두리]

```

1 Sub 셀테두리()
2   Range("D2:G6").Select
3   With Selection.Borders(xlBottom)
4     .LineStyle = xlDash
5     .ColorIndex = 5
6   End With
7   Range("D2:G2").Select
8   With Selection.Borders(xlTop)
9     .LineStyle = xlDouble
10    .Weight = xlThick
11  End With
12 End Sub

```

보기 10.4.10에서 2 줄은 D2:G2 셀을 선택하고 3 줄은 선택한 셀 위에 4 줄은 선 종류는 데쉬선, 선 색은 파랑으로 설정하였다. 7 줄은 D2:G2 셀을 선택하고 8 줄은 선택한 셀 아래에 줄은 선 종류는 이중 실선, 선 굵기는 굵게 설정하였다. 실행결과는 그림 10.4.4에서 확인한다.

Range 개체 정렬 속성은 표 10.4.5에 정리하였다. Borders 속성 값과 셀 테두리는 표 10.4.6에 정리하였다. Weight 속성 값과 셀 테두리 선 굵기는 표 10.4.7에 정리하였다. LineStyle 속성 값과 셀 테두리 선 종류는 표 10.4.8에

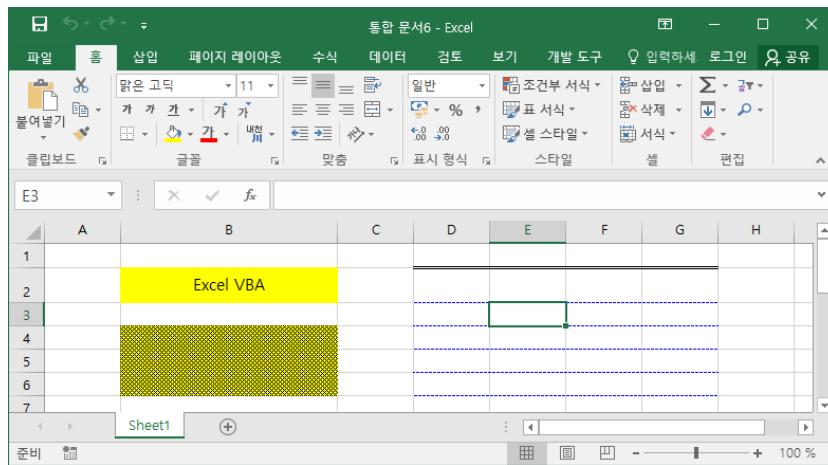


그림 10.4.4: 더 많은 셀 속성

HorizontalAlignment 속성		VerticalAlignment 속성	
속성 값	정렬 방법	속성 값	정렬 방법
xlHAlignCenter	가운데 맞춤	xlVAlignCenter	가운데 맞춤
xlHAlignLeft	왼쪽 맞춤	xlVAlignTop	위쪽 맞춤
xlHAlignRight	오른쪽 맞춤	xlVAlignBottom	아래쪽 맞춤
xlHAlignDistributed	균등 분할	xlVAlignDistributed	균등 분할
xlHAlignJustify	양쪽 맞춤	xlVAlignJustify	양쪽 맞춤

표 10.4.5: Borders 개체의 셀 정렬

정리하였다.

## Cells 개체

Cells 개체 사용방법은

### 사용방법 10.4.4. [Cells]

속성 값	셀 테두리	속성 값	셀 테두리
xlBottom	셀 아래	xlDiagonalDown	셀 45도 사선
xlLeft	셀 왼쪽	xlDiagonalUp	셀 135도 사선
xlRight	셀 오른쪽	xlInsideHorizontal	선택 셀 수평
xlTop	셀 위	xlInsideVertical	선택 셀 수직

표 10.4.6: Borders 개체의 셀 테두리

속성 값	셀 테두리 선 굵기	속성 값	셀 테두리 선 굵기
xlHairline	매우 가는 선	xlMedium	중간 선
xlThin	가는 선	xlThick	굵은 선

표 10.4.7: Weight 개체의 셀 테두리 선 굵기

속성 값	셀 테두리 선 종류	속성 값	셀 테두리 선 종류
xlDash	데쉬 선	xlDouble	이중 실선
xlDashDot	데쉬선과 점선	xlSlantDashDot	사선
xlDashDotDot	데쉬선 점선 점선	xlContinuous	실선
xlDot	점선	xlLineStyleNone	선 없음

표 10.4.8: LineStyle 개체의 셀 테두리 선 종류

Cells(*row\_num*, *col\_num*)

이다. 인수는

- ◎ *row\_num*은 행의 위치

- ◎ *col\_num*은 열의 위치

예를 들어 Cells(1,1)은 A1 셀을 나타낸다. 셀에 엑셀 함수를 사용하려면 Value, Formula, FormulaR1C1 속성을 사용한다.

- ◉ Value : 셀에 저장된 값이다. 값으로 저장된다.
- ◉ Formula : 셀에 저장된 값이다. 수식으로 저장된다. 셀 참조는 A1이다.
- ◉ FormulaR1C1 : 셀에 저장된 값이다. 수식으로 저장된다. 셀 참조는 R1C1이다.

예를 들어

```
Cells(1,1).Formula = "=RAND()"
```

를 실행하면 A1 셀에 0에서 1사이의 실수에서 난수가 출력된다.

다음은 Range 개체에 엑셀 함수를 사용한 결과이다. 1 줄은 수식이 저장되고 화면에는 두 셀을 합한 결과가 보인다. 2 줄도 수식이 저장되고 함수에서 셀 참조는 A1 참조스타일이다. 3 줄은 수식이 저장되고 함수에서 셀 참조는 R1C1 참조스타일이다. 4 줄은 Value를 저장하면 값이 저장된다. 5 줄은 Formula를 저장하면 수식이 저장된다.

```
1 Range("A3").Value = "=SUM(A1:A2)"
2 Range("B3").Formula = "=SUM(B1:B2)"
3 Range("C3").FormulaR1C1 = "=SUM(R[-2]C[0]:R[-1]C[0])"
4 Range("D3").Value = Range("A3").Value
5 Range("E3").Value = Range("B3").Formula
```

비주얼 베이직에 엑셀함수 사용은 이중 따옴표 ”와 ”사이에 수식을 입력한다. 그런데 부득이하게 엑셀함수를 사용할 때 이중 따옴표가 포함되었으면 이중 따옴표를 하나 더 입력해야 오류없이 문장이 실행한다. IF 함수에 이중 따옴표를 사용하는 경우를 보자.

```
Cells(1,2).Formula = "=IF(A1>5,\"크다\", \"작다\")"
```

A1 셀이 5보다 큰 경우는 크다 문자열이 그렇지 않은 경우는 작다 문자열이 출력되도록 하려면 이중 따옴표를 하나 더 입력한다.

#### 보기 10.4.11. [Cells 속성]

```

1 Sub CellExample()
2     Cells(1, 1).Formula = "=RAND()"
3     Cells(1, 2).Formula = "=IF(A1>5, ""크다"" , ""작다"" )"
4     Cells(1, 3).Value = "글자색"
5     Cells(1, 3).Font.ColorIndex = 5
6 End Sub

```

보기 10.4.11은 Cells 개체의 속성 값을 설정하고 결과는 그림 10.4.5에서 확인한다. 4 줄은 A3 셀에 “글자색”을 입력하고 5 줄 글자색은 파란색이다.

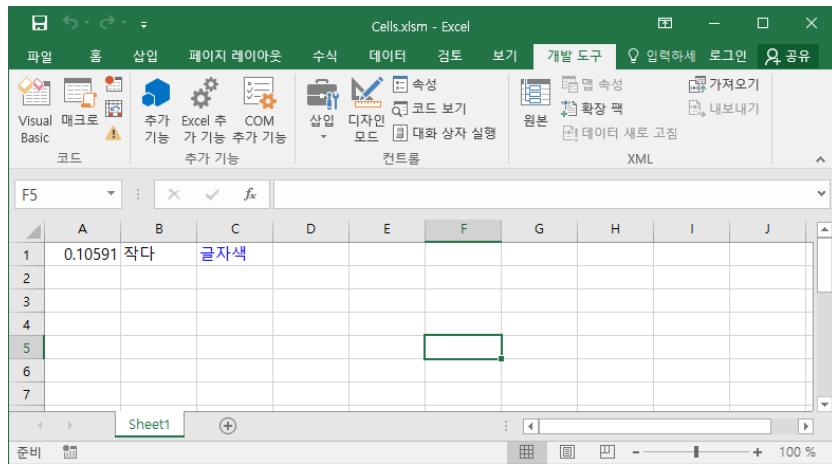


그림 10.4.5: Cells 개체

## Offset 개체

Offset 개체는 현재 셀을 기준으로 속성을 적용한다. 사용법은  
사용방법 10.4.5. [Offset]

```
expression.Offset("RowOffset, ColumnOffset")
```

이며,

- ① *expression* : Range, Cell, ActiveCell 객체 사용
- ② *RowOffset* : 정수를 입력하며, 양수는 아래, 음수는 위, 0은 기본값
- ③ *ColumnOffset* : 정수를 입력하며, 양수는 오른쪽, 음수는 왼쪽, 0은 기본값

이다.

#### 보기 10.4.12. [Offset 속성]

```
1 Sub 셀참조()
2     Worksheets("Sheet1").Activate
3     ActiveCell.Offset(2,2).Activate
4 End Sub
```

보기 10.4.12에서 2 줄은 Sheet1 시트를 활성화 상태로 3 줄은 현재 셀에서 아래로 두 행, 오른 쪽으로 두 열 옮긴 셀을 현재 셀로 하였다.

## 제 10.5 절 VBA에서 제공하는 함수 사용방법

VBA는 이미 내부에 만들어진 함수가 있다. 이 절에서 소개하는 함수는 외부에서 값을 받는 기능을 제공하는 InputBox 함수와 내부에서 입력한 값을 출력하는 MsgBox 함수이다.

### 10.5.1 MsgBox 함수

먼저 소개하는 함수는 MsgBox이다. 이 함수의 사용법은 다음과 같다.

#### 사용방법 10.5.1. [MsgBox]

상수	값	단추 설명
vbOK	1	확인 단추
vbCancel	2	취소 단추
vbAbort	3	중단 단추
vbRetry	4	다시 시도 단추
vbIgnore	5	무시 단추
vbYes	6	예 단추
vbNo	7	아니오 단추

표 10.5.1: 버튼에서 되돌려 주는 값

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

이 함수는 입력한 메시지를 사용자에게 전달하고 함수 사용이 종료되면 버튼 값을 되돌려 준다. 사용법에서 기울임꼴 단어는 함수에 붙여진 인수이다. 특히 대괄호로 묶인 인수들은 생략할 수 있으나 그 이외의 것은 반드시 값을 입력해야 한다. 이 함수에서 *prompt* 인수는 반드시 사용해야 한다. 다음은 인수의 사용법이다. 주의사항으로 실제 코드에서는 대괄호를 입력하지 않는다. 대괄호는 설명하기 편리하도록 사용한 기호이다.

- ◉ *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.

*buttons*에 사용하는 인수들은 표 10.5.2와 같이 상수 문자나 숫자로 설정한다. `MsgBox`에서 버튼을 누르면 되돌려 주는 값이 있으며 그 내용은 표 10.5.1에 있다. 함수에 인수 입력 방법은 두 가지이다.

- ◉ 함수와 메서드 인수들은 위치나 이름별로 지정한다. 위치에 따른 인수 지정은 각 인수들 사이에 쉼표를 넣어 구문에 나타난 순서대로 사용한다. 만일 중간에 있는 인수 값을 입력하지 않으려면 콤마를 두 개 입력하고 (,,) 인수의 위치가 끝이면 생략한다. 이 방법은 함수나 메서드가 값을 전달하지 않거나 결과값을 무시할 때 사용한다. 다음 함수의 사용법을 보자.

#### 보기 10.5.1. 【`MsgBox` 함수】

```
 MsgBox "안녕하세요!", 1, "인사"
```

첫 번째 *prompt* 인수 값은 “안녕하세요”이고 반드시 입력한다. 두 번째 *buttons* 인수 값은 1이고, 버튼 종류는 확인 버튼이다. 세 번째 *title* 인수 값은 제목표시줄에 나타날 문자열 “인사”이다.

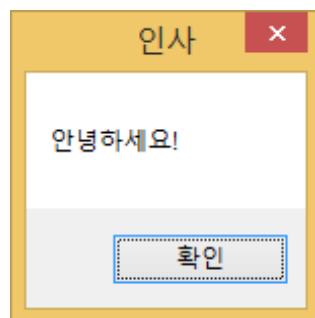


그림 10.5.1: `MsgBox` 함수

상수	값	보이는 단추
vbOKOnly	0	확인 단추
vbOKCancel	1	확인, 취소 단추
vbAbortRetryIgnore	2	중단, 다시 시도, 무시 단추
vbYesNoCancel	3	예, 아니오, 취소 단추
vbYesNo	4	예, 아니오 단추
vbRetryCancel	5	다시 시도, 취소 단추
vbCritical	16	중대 메시지 아이콘
vbQuestion	32	쿼리 경고 아이콘
vbExclamation	48	메시지 경고 아이콘
vbInformation	64	메시지 정보 아이콘
vbDefaultButton1	0	첫째 단추가 기본값
vbDefaultButton2	256	둘째 단추가 기본값
vbDefaultButton3	512	세째 단추가 기본값
vbDefaultButton4	768	넷째 단추가 기본값
vbApplicationModal	0	응용 프로그램 모달로 사용자가 현재 사용 중인 프로그램 내에서 계속 작업하면서 반드시 메시지 상자에 응답
vbSystemModal	4096	시스템 모달로 사용자가 메시지 상자에 응답할 때까지 모든 프로그램들은 정지
vbMsgBoxHelpButton	16384	메시지 상자에 도움말 단추 추가
vbMsgBoxSetForeground	65536	메시지 상자 창을 전경 창으로 설정
vbMsgBoxRight	524288	텍스트를 오른쪽 정렬
vbMsgBoxRtlReading	1048576	히브리어와 아랍어 시스템에서 오른쪽에서 왼쪽으로 읽을 수 있도록 설정

표 10.5.2: 인수로 사용하는 버튼의 특성

- ◉ 이름별 인수 지정은 인수 이름, 콜론, 등호(:=), 인수 값을 차례로 입력한다. 이 방법은 인수 배열 순서에 상관없이 인수를 지정한다. 다음 보기에는 늦게 배치된 Title 인수를 먼저 사용하고 처음에 있는 Prompt 인수를

나중에 사용한다. 중간 있는 Button은 사용하지 않았다.

### 보기 10.5.2. [MsgBox 함수2]

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

- 함수나 메서드가 값을 반환하면(리턴값이 있으면) 함수와 메서드는 인수들을 괄호로 묶는다. 다음 보기는 MsgBox 함수를 사용하여 결과값(리턴값; 반환값)을 저장한다. 2 줄은 myVar 변수에 MsgBox 리턴값을 되돌려준다. 4 줄에서 myVar 변수 값은 확인한다(그림 10.5.2).

### 보기 10.5.3. [MsgBox 함수3]

```
1 Sub Question()
2     myVar = MsgBox(Prompt:="나는 내 일을 좋아한다.", _
3                     Title:="대답 상자", Buttons:="4")
4     MsgBox myVar
5 End Sub
```

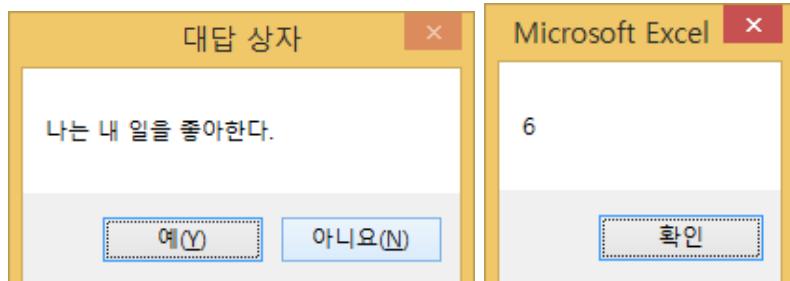
위의 함수에서 2-3 줄은 한 줄에 입력해야 되나 밑줄(.)과 빈 공간(□)을 사용하여 두 줄을 사용하였고, 괄호가 사용되었으니 함수의 리턴값을 받아야 되며 변수 myVar에 결과값이 저장된다. 4 줄은 2-3 줄에서 리턴 값이 무엇인지 확인하려고 MsgBox를 사용하였으며 그림 10.2(a)에서 예를 누르면 10.2(b)에 6이 출력된다.

## 10.5.2 InputBox 함수

다음은 입력 기능이 있는 InputBox 함수의 사용법을 알아보자.

### 사용방법 10.5.2. [InputBox]

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



(a) MsgBox에서 리턴 값 보내기      (b) MsgBox에 리턴 값 출력

그림 10.5.2: MsgBox 사용 예

이 함수에서 옵션 사용법이다.

- ◉ *prompt* : 반드시 입력하는 필수 요소로 대화 상자에서 보여주는 메시지다.
- ◉ *title* : 제목표시줄 문자열로 선택 요소이다.
- ◉ *default* : 입력란에 보여줄 기본 입력값이다.
- ◉ *xpos* : 화면 왼쪽에서 대화상자까지 간격으로 생략하면 기본값은 수평 중앙이다.
- ◉ *ypos* : 화면 위에서 대화상자까지 간격으로 생략하면 대화상자는 1/3 이다.
- ◉ *helpfile* : 대화상자에 제공하는 도움말이이다. 파일은 문자열에 입력한다. 이 옵션은 *context*와 함께 사용한다.
- ◉ *context* : 작성한 도움말 내용이 도움말 항목에 나타날 번호이다. 이 옵션은 *helpfile*과 함께 사용한다.

#### 보기 10.5.4. 【InputBox 함수】

```

1 Sub Question()
2     myVar = InputBox(Prompt:="숫자를 입력하세요.", _
3                         Default:="100", Title:="홀수, 짝수 판별")
4     MsgBox myVar
5     If myVar Mod 2 = 0 Then
6         Range("B2") = "입력한 숫자 " & myVar & "은(는) 짝수이다."
7     Else
8         Range("B2") = "입력한 숫자 " & myVar & "은(는) 홀수이다."
9     End If
10 End Sub

```

보기 10.5.4에서 입력 상자의 메시지는 숫자를 입력하세요, 입력란 기본 값은 100, 제목표시줄은 홀수, 짝수 판별로 설정하였다. 그럼 10.5.3에 기본 설정 값 100이 입력된 상태에서 확인 버튼을 누르면 myVar 변수에 100이 저장되고 4 줄에 그 값이 출력된다. B2 셀에는 홀수인지 짝수인지 판별하여 그 결과를 출력한다.

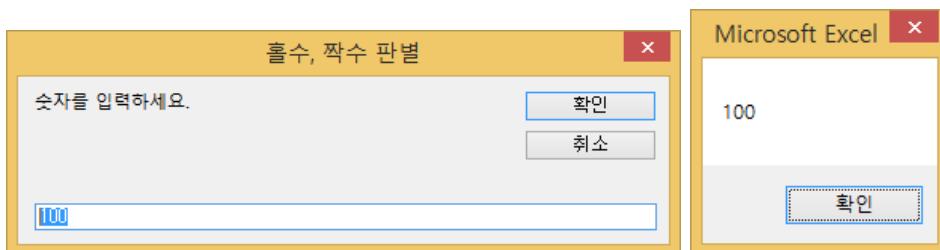


그림 10.5.3: InputBox 사용

## 연습문제

1. 다음 조건에 맞게 매크로를 작성하고 실행한다.
  - ① 개발도구→Visual Basic에서 모듈을 추가한다.
  - ② 코드 입력창에 Sub 문을 추가한다. Sub 문 이름은 메시지창부르기이다.
  - ③ MsgBox 함수로 “안녕하세요!!!”가 메시지 창에 나오게 코드를 만든다.
  - ④ 시트 Sheet1에 양식 컨트롤 단추를 추가한다. 이 단추의 텍스트는 메시지 창 부르기이다.
  - ⑤ 이 단추를 누르면 메시지창부르기 매크로가 실행되게 연결한다.
2. 다음 조건에 맞게 비주얼베이직을 작성한다.
  - ① 시트 Sheet2에서 버튼을 누른면 MsgBox 창이 나타난다. 버튼의 텍스트는 버튼종류확인이다.
  - ② MsgBox 창에서 메시지는 “당신은 대학생입니까?”이고 버튼은 예와 아니오 2개이다.
  - ③ 예를 누르면 “당신은 대학생이군요.”, 아니오를 누르면 “당신은 대학생이 아니군요.”가 MsgBox에 출력된다.
3. 다음 조건에 맞게 비주얼베이직을 작성한다.
  - ① 시트 Sheet3에서 버튼을 누른면 InputBox 창이 나타난다. 버튼의 텍스트는 인사이다.
  - ② InputBox에 이름 ○○○을 입력한다.
  - ③ MsgBox 창에 입력한 ○○○님 반갑습니다. 가 출력된다.
4. 다음 조건에 맞게 비주얼베이직을 작성한다.

- ① 시트 Sheet4에서 버튼을 누르면 InputBox 창이 나타난다. 버튼의 텍스트는 홀/짝 판별이다.
- ② 숫자를 InputBox에 입력한다. 이 창에서 창 제목은 홀수, 짝수 판별이고, 안내 메시지는 숫자를 입력하세요.이며 입력창에 기본으로 입력된 값은 100이다.
- ③ 입력한 숫자가 짝수이면 B2 셀에 “입력한 숫자 ○○은(는) 짝수이다.”, 홀수이면 B2 셀에 “입력한 숫자 ○○은(는) 홀수이다.”가 출력된다.

5. 다음 조건에 맞게 비주얼베이직을 작성한다.

- ① 시트 Sheet5에서 버튼을 누르면 InputBox 창이 나타난다. 버튼의 텍스트는 합구하기이다.
- ② 이 창에 자연수를 입력한다.
- ③ C2 셀에 “1에서 ○○까지 합은”
- ④ C3 셀에 “○○입니다.”가 출력된다.
- ⑤ MsgBox 창에 “1에서 ○○까지 합은 □□입니다.”가 출력된다.

## 제 11 장

# VBA 프로그래밍 기법

이 장은 VBA 프로그래밍 사용법을 소개한다. 프로그래밍 사용방법은 대부분 언어가 거의 유사하므로 다른 언어를 프로그래밍한다면 이 장은 학습하기가 수월하다.

## 제 11.1 절 변수와 상수

변수는 프로그램에서 데이터를 저장하는 장소이다. 말 그대로 변수는 값 변경이 가능하다. 처음에 입력된 값이 변하지 않는 수도 있다. 그 수는 상수라고 한다. 변수는 데이터 종류(data type)에 따라 여러 가지로 구분하여 사용한다 (표 11.1.1). 종류와 메모리 크기, 값의 범위가 다양한 데이터 종류는 용도에 적절하게 사용하면 메모리 사용양을 줄일 수 있다.

### 11.1.1 변수 선언

변수 선언은 Dim 키워드를 사용한다. 데이터 타입 변수의 사용방법은  
**사용방법 11.1.1. [변수 선언]**

데이터 종류	메모리 크기	자료 형태	값의 범위
Byte	1 bytes	정수	0 ~ 255
Boolean	2 bytes	논리	TRUE, FALSE
Integer	2 bytes	정수	-32,768 ~ +32,767
Long	4 bytes	정수	-2,147,483,648 ~ 2,147,483,647
Single	4 bytes	실수	-3.4e38 ~ +3.4e38
Double	8 bytes	실수	-1.8e308 ~ +1.8e308
Currency	8 bytes	통화	-922,337,203,685,477.5808 ~ 922,337,203,685,477.00
Date	8 bytes	날짜	1st January 100 ~ 31st December 9999
Object	4 bytes	객체	Reference
String	varies	문자	고정 문자는 65,500 문자, 변동 문자는 2 billion 문자
Variant	varies	가변형	숫자는 Double, 문자는 String

표 11.1.1: 데이터 형태 및 범위

```
Dim varname([([subscripts])]) [As type]
[, varname([([subscripts])]) [As type]] . . .
```

이다. Dim은 필수 키워드이고 반드시 사용하는 요소는 varname(변수 이름)이다. [subscripts]는 배열에서 참조번호이다. 배열은 11.5 절에 소개하였다. As type은 사용할 데이터 종류이다. 만일 데이터 종류를 생략하면 기본값은 Variant이다. 변수가 속한 해당 범위에서 동일한 변수명은 존재할 수 없다. 이 규칙을 어기면 에러가 발생한다. 변수 이름은 식별자 규칙으로 정한다(11.3 절 참조). 다음은 변수 선언하는 간단한 보기이다. 변수 선언을

### 보기 11.1.1. [변수 선언]

```
Dim myVar, nextVar, thirdVar
```

로 하면 변수 myVar, nextVar, thirdVar를 만든다. 이들은 모두 데이터 종류를 사용하지 않았으니 기본값인 Variant 변수로 선언된다. 그러나 모든 경우 Variant를 사용하면 데이터 형태를 지정하지 않아도 되어 편리하다. 그러나 Variant는 많은 메모리를 필요하니 작은 데이터 형식을 사용 가능한 경우라면 그 데이터 종류를 명확하게 지정한다. 예를 들어 String으로 선언하려면

#### 보기 11.1.2. 【String 변수 선언】

```
Dim myAnswer As String
```

로 한다.

여러 개 변수 선언은 변수 이름과 데이터 형태를 컴마(,)로 구분하고 변수 이름과 데이터 형태를 반복하여 적는다. 다음 보기는 모두 Integer로 선언하려고 모든 변수에 데이터 형태를 명시한다.

#### 보기 11.1.3. 【여러 개 변수 선언(데이터 형태 명시)】

```
Dim x As Integer, y As Integer, z As Integer
```

다음 보기와 같이 데이터 형태는 생략하고 변수를 여러 개 선언하면

#### 보기 11.1.4. 【여러 개 변수 선언】

```
Dim x, y, z As Integer
```

x, y는 데이터 형식을 설정하지 않았으니 기본값인 Variant 데이터 형식이고 z만 Integer 데이터 형식으로 지정된다.

### 11.1.2 상수 선언

상수 선언에 대하여 알아보자.

#### 사용방법 11.1.2. [상수 선언]

```
Const constname = value
```

상수 선언 방법은 변수와 거의 동일하다. 차이점으로 상수 선언은 상수명에 반드시 값을 배정(assignment)한다. Const는 상수 선언 키워드이고 *constname*는 사용할 상수명이며 *value* 상수명에 배정할 값이다. 다음 보기는 상수를 선언한다.

#### 보기 11.1.5. [상수 선언]

```
1 Const iMaxClient = 100  
2 Const iMinClient = 2
```

상수 iMaxClient에 100을 iMinClient에 2를 배정하였다.

### 11.1.3 배열 선언

배열 변수는 한 번에 동일한 데이터 형태를 여러 개 만든다. 배열 변수 선언은

```
Dim myArray() AS Integer
```

와 같이 선언한다. 위 보기는 정수형 배열을 선언하였으니 정수형 데이터를 여러 개 사용할 수 있다. 배열 내용은 11.5 소절을 참조한다.

## 제 11.2 절 변수에 값 배정

변수 선언은 그 변수에 값 배정(assign)의 목적이다. 변수에 값 배정은

```
변수이름 = 값
```

을 입력한다. Integer 형 변수 i에 20 배정은

```
i = 20
```

이다. Object 데이터 타입은 변수에 값 배정이 다른 것과 약간 다르다. Object 변수의 값 배정은 Set 문을 사용한다. 사용방법은

- 1 Set 변수이름 = 배정한 개체 참조
- 2 Set 변수이름 = New 개체

이다. 이미 선언하고 배정된 개체 참조 변수명이 yourObject이면

```
Set myObject = yourObject
```

이다. 한 번에 개체 변수 생성과 배정은

```
Set myObject = new Object
```

이다. 변수는 데이터 종류를 선언없이 사용할 수 있다. 즉 변수는 프로시저 만들다가 중간에 데이터 종류와 Dim 문 없이

```
변수이름 = 변수값
```

로 선언하여 사용이 가능하다. 그런데 다른 프로그래밍 언어는 변수 선언없이 사용하면 에러가 발생한다. 그렇기 때문에 다른 프로그래밍 언어를 배운 사람은 프로시저 중간에 변수 선언하면 코드 분석에 혼돈할 수도 있다. 반드시 변수 선언이 필요하게 설정하려면 모듈 처음에

```
Option Explicit
```

를 입력한다. 이후 변수 선언없이 사용하면 에러가 발생한다.

### 제 11.3 절 Visual Basic 식별자 규칙

Visual Basic 모듈에서 프로시저, 상수, 변수, 인수의 이름(식별자; identifier)을 정할 때는 다음과 같은 규칙을 따른다.

- ◉ 첫 번째 문자는 숫자를 사용할 수 없으며 영어나 한글 등의 문자를 사용한다.
- ◉ 공백이나 마침표(.), 느낌표(!), @, &, \$, # 문자는 이름에 사용할 수 없다.
- ◉ 길이는 255자를 넘을 수 없다.
- ◉ 보통 Visual Basic의 Function, Sub, Property 같은 이름은 사용할 수 없다. 즉 언어가 먼저 사용하려고 저장된 예약어인 키워드는 사용할 수 없다.
- ◉ 같은 수준의 범위에서는 이름을 반복하여 붙일 수 없다. 예를 들어 같은 프로시저에서 age라는 변수를 두 개 선언할 수 없다.

### 제 11.4 절 Visual Basic 문 작성

Visual Basic의 문(statement)은 키워드, 연산자, 변수, 상수, 식을 포함할 수 있다. 각 문은 다음 세 분야 중 하나에 속한다.

- ◉ 선언문은 변수나 상수, 프로시저의 이름을 정하고 데이터 형식을 지정한다. 선언문 작성은 Dim, ReDim, Public, Private, Static, Const 키워드로 한다.
- ◉ 선언문 변수명에 값이나 식을 배정한다. 배정하는 값은 직접 입력하거나 Let, Set 문을 사용한다.
- ◉ 동작을 초기화하는 실행문이다. 이 문은 함수에 입력값을 전달하여 그 내용을 실행한다. 또한 변수, 함수, 연산자, 조건문, 반복문으로 연산을 실행한다.

### 11.4.1 여러 줄에 계속되는 문

문은 기본값이 보통 한 줄에 쓴다. 그러나 줄 내용이 길어서 여러 줄에 한 문을 사용하려면 줄 연속 문자를 사용하여 다음 줄로 문을 연결한다. 여러 줄에 한 문을 넣으려면 한 줄의 맨 끝에 밑줄(underscore)과 빈 공간(empty space)을 차례로 붙여쓴다. 다음 MsgBox 실행문은 3 줄에 걸쳐 연결된다.

#### 보기 11.4.1. 【한 실행문을 여러 줄에 입력】

```

1 Sub DemoBox()
2     Dim myVar As String
3     myVar = "홍길동님"
4     MsgBox Prompt:="안녕하세요 " & myVar, _
5         Title:="인사", _
6         Buttons:=vbExclamation
7 End Sub

```

### 11.4.2 여러 문을 한 줄에 사용

앞 절 내용과 반대로 간단한 문들은 한 줄에 여러 실행문을 입력할 수 있다. 이런 경우 프로그램 코드가 여러 줄에 문을 사용한 것보다 보기 편할 수 있다.

여러 실행문을 한 줄에 사용하는 방법은 실행문과 실행문 사이에 콜론(:)을 삽입하면 된다.

다음과 같이 2 줄에 3 개의 문을 실행하였다. 그 내용은 세 변수에 값을 배정하고 그 결과를 출력하면 567이 출력된다.

#### 보기 11.4.2. 【한 줄에 여러 실행문 넣기】

```

1 Sub 한줄에여러문넣기()
2     i=5: j=6: k=7
3     MsgBox i & j & k
4 End Sub

```

### 11.4.3 주석 추가

가끔 프로그램에서 코드를 실행하고 싶지 않은 부분이 있다. 다음과 같이

- ◉ 작성한 코드에 대한 설명을 달아놓거나
- ◉ 특정한 코드에서 에러가 발생하는 장소이다.

따라서 컴파일러나 인터프리터가 해당 문을 실행하지 않게 설정해야 한다. 이것을 프로그래밍에서 주석(comment)이라고 부른다. 주석 사용 방법은

- ◉ 작은따옴표(‘) 다음에 작성하거나(행 중간에 사용가능)
- ◉ Rem 다음에 주석을 입력한다. 행 처음부터 시작해야 한다.

실행문과 함께 주석 사용은 실행문 뒤에 작은 따옴표를 넣는다. 주석이 설정된 문은 글자 색이 녹색이다.

#### 보기 11.4.3. [주석문 넣기]

```
1 Sub 주석문사용()
2     i=5: j=6: k=7 '한 줄에 여러 문 실행
3     Rem 세 변수에 입력된 값 출력
4     MsgBox i & j & k
5 End Sub
```

2 줄은 문을 사용하고 이후 주석을 추가하였고 3 줄은 처음부터 주석을 사용하였다.

### 11.4.4 구문 오류 검사

코드를 한 줄 입력하고 Enter키를 눌렀을 때 오류가 있으면 빨간색으로 표시된다. 잘못된 부분은 수정하고 재설정 버튼을 눌러 변경된 내용을 적용시킨다.

## 제 11.5 절 배열 사용

배열은 동일한 데이터 형식을 여러 개 사용하는 변수이다. 배열은 저장 공간이 여러 개인 단일 변수인 반면 일반 변수는 값의 저장 공간이 하나이다. 배열에서 참조는 저장된 변수 값을 가져오는 것을 말한다. 배열은 전체 또는 각 요소(element)들을 참조할 수도 있다. 요소는 배열 변수에 선언한 데이터 형식이 동일한 개개의 값이다. 다음은 배열 사용법이다.

### 사용방법 11.5.1. 【배열 선언】

```
Dim varname(subscripts) [ As type]  
Dim varname(start To end) [As type]
```

첫 번째는 변수 사용법과 동일하다. 추가로 *subscripts*는 배열의 요소 갯수이다. 요소 개수는 *subscripts* + 1이고 요소 번호(index)는 0부터 *subscripts*까지이다. 두 번째는 배열의 요소 번호가 *start*에서 *end*로 요소 갯수는 *end - start + 1*이다.

예를 들어, 매일 하루 경비를 저장할 때 365개 변수 선언보다 365개 요소를 가지는 배열 변수 한 개 선언이 효율적이다. 배열은 각 요소마다 값이 한 개 있다. 다음 문은 365개 요소로 된 배열 변수 *curExpense*를 선언한다. 기본적으로 배열의 요소번호는 0에서 시작하므로 배열의 요소 상한값은 365가 아닌 364이다.

```
Dim curExpense(364) As Currency
```

배열에서 요소의 값은 요소 번호로 구분하여 배정한다. 4 줄과 6 줄은 For 문으로 0에서 364까지 365회 반복하였다. 5 줄은 365개 각 배열 요소에 모두 20을 배정하였다.

### 보기 11.5.1. 【배열 사용】

```
1 Sub FillArray()
2     Dim curExpense(364) As Currency
3     Dim intI As Integer
4     For intI = 0 to 364
5         curExpense(intI) = 20
6     Next
7 End Sub
```

### 11.5.1 하한값 변경

배열에서 시작 인덱스를 변경하면 작업이 편리한 경우가 있다. 이것이 필요하면 모듈의 맨 위에 Option Base 문을 사용하여 첫 번째 요소의 기본 인덱스를 0에서 특정한 값으로 변경한다. 다음 보기에서 1 줄은 Option Base 문으로 첫 번째 요소번호를 1로 설정하였다. 2 줄은 Dim 문은 요소가 365개인 배열 변수 curExpense를 선언하였다.

#### 보기 11.5.2. [배열 기본 인덱스 변경]

```
1 Option Base 1
2 Dim curExpense(365) As Currency
```

또한 To를 사용하여 배열 요소번호를 명시적으로 설정할 수 있다. 다음 보기는 To로 배열 요소번호 하한값을 명확하게 설정하였다. 1 줄은 요소번호 범위를 1에서 365, 2 줄은 7에서 13까지 설정하였다.

#### 보기 11.5.3. [배열의 인덱스를 명시적으로 설정]

```
1 Dim curExpense(1 To 365) As Currency
2 Dim strWeekday(7 To 13) As String
```

### 11.5.2 배열에 값 저장

배열을 생성하고 각 요소에 값을 배정하는 방법은 두 가지가 있다. 첫 번째는 각 배열 요소에 선언한 데이터 형식을 각각 입력한다. 다음 보기는 배열요소에 값을 각각 저장한다.

#### 보기 11.5.4. [Variant 배열 형식 배정]

```

1 Dim varData(3) As Variant
2 varData(0) = "Claudia Bendel"
3 varData(1) = "4242 Maple Blvd"
4 varData(2) = 38
5 varData(3) = Format("06-09-1952", "일반 날짜")

```

두 번째는 Array 함수로 Variant 변수에 값을 배정한다.

#### 보기 11.5.5. [Array 함수로 Variant 배열 형식 배정]

```

1 Dim varData As Variant
2 varData = Array("Ron Bendel", "4242 Maple Blvd", 38, _
3 Format("06-09-1952", "General Date"))

```

어떤 방법으로 변수를 만들어도 Variant 배열 요소에 같은 동일하게 저장된다. 보기 11.5.4와 보기 11.5.5에 다음 문을 추가하고 결과가 같은지 확인하자.

#### 보기 11.5.6. [Variant 배열 요소 값 부르기]

```
MsgBox "Data for " & varData(0) & " has been recorded."
```

### 11.5.3 다차원 배열 사용

지금까지 소개한 배열은 일차원 배열이다. 2차원 이상 배열을 다차원 배열이라고 한다. 배열은 60 차원까지 선언할 수 있다. 다음 보기는 요소가  $5 \times 10$

개인 2 차원 배열이다. 즉 다차원 배열 선언은 요소 갯수를 콤마로 구분하여 입력한다.

#### 보기 11.5.7. [2 차원 배열]

```
Dim sngMulti(1 To 5, 1 To 10) As Single
```

2 차원 배열을 행렬로 표현하면 첫 번째 인수는 행을 나타내고 두 번째 인수는 열을 나타낸다.

다차원 배열에 값 배정은 For-Next 문을 여러 개 겹쳐 사용한다. 다음 보기는 2 차원 배열 요소에 Single 값은 채운다.

#### 보기 11.5.8. [2 차원 배열에 값 채우기]

```
1 Sub FillArrayMulti()
2     Dim intI As Integer, intJ As Integer
3     Dim sngMulti(1 To 5, 1 To 10) As Single
4     For intI = 1 To 5
5         For intJ = 1 To 10
6             sngMulti(intI, intJ) = intI * intJ
7             Cells(intI, intJ) = sngMulti(intI, intJ)
8         Next intJ
9     Next intI
10    End Sub
```

3 줄은  $5 \times 10$  개 2 차원 배열을 선언한다. 5에서 10 줄은 각 요소에 값을 배정한다. 8 줄은 저장된 배열 요소 값을 현재 시트 (1,1)에서 (5,10) 셀에 출력한다.

### 11.5.4 동적 배열 선언

지금까지 배열은 처음에 선언하면 배열 요소의 갯수 변경을 할 수 없었다. 동적 배열 선언은 코드가 실행하는 동안 배열의 크기를 정하고 변경도 가능하다. 다음 보기는 Dim 키워드 사용하여 요소 크기가 없는 동적 배열을 선언한다. 빈 배열 생성은 괄호 안을 공백으로 남겨둔다.

### 보기 11.5.9. 【동적 배열 선언(크기 설정 않함)】

```
Dim sngArray() As Single
```

`ReDim` 문은 임의로 배열 요소 크기를 선언한다. `ReDim` 문은 이미 선언한 배열을 다시 사용해도 되고 새로운 배열을 선언할 수 있다. 그렇기 때문에 이미 있는 배열을 다시 선언한다면 배열 이름을 잘못 쓰지 않도록 주의 한다. `ReDim` 문은 차원의 수를 변경하거나 요소의 수를 정의하고 각 차원의 상한값과 하한값을 정의한다. 다음 보기에서 오류가 발생하는 곳을 찾아보자.

### 보기 11.5.10. 【배열 선언】

```
1 Sub ArrayTest()
2     Dim i() As Integer
3     Dim j(5) As Integer
4     Dim m(2 To 5) As Integer
5     ReDim i(11) As Integer
6     ReDim j(2 To 6) As Integer
7     ReDim m(2 To 10) As Integer
8     ReDim n(2 To 10) As Integer
9 End Sub
```

동적 배열 다시 선언하면 배열에 있는 기존 값은 잃게 된다. 따라서 배열의 기존 값을 보존하고 배열을 확장하려면 `ReDim Preserve`를 사용한다. 다음 보기는 원래 요소의 현재값을 잃지 않고 배열 `varArray`를 10 요소만큼 확장한다.

### 보기 11.5.11. 【배열 요소 확장】

```
ReDim Preserve varArray(UBound(varArray) + 10)
```

`Preserve` 키워드는 동적 배열과 함께 사용하면 마지막 차원의 상한값만 변경 할 수 있지만 차원의 수는 변경할 수 없다.

## 제 11.6 절 제어문

프로그램을 작성에 사용자가 설정한 조건이 맞는 동안 일련된 명령을 실행하는 키워드가 있는데 이것을 제어문(control statement)이라고 말한다.

### 11.6.1 If-Then-ElseIf-Then-Else 문

If-Then-ElseIf-Then-Else 문(If 문)은 조건 값에 따라서 특정 문이나 문 블록을 실행하는 제어문이다. If 문은 조건이 True이면 문을 실행하고 False이면 문을 실행하지 못한다. If 문은 필요한 수만큼 반복하여 사용할 수 있다. 그러나 읽기 쉽게 하려면 여러 개로 중첩된 If 문을 사용하는 것보다는 기능이 비슷한 Select-Case 문을 사용할 수도 있다. 다음은 If 문의 사용법이다.

#### 사용방법 11.6.1. [If-Then-ElseIf-Then-Else]

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

여기서

- ① condition<sub>1</sub> : 이 조건이 True이면 statements<sub>1</sub>이 실행된다.
- ② condition<sub>2</sub> : condition<sub>1</sub>을 제외하고 이 조건이 True이면 statements<sub>2</sub> 실행된다.

조건이 여러 개이면 ElseIf condition<sub>2</sub> Then을 여러 번 사용한다. 반면 조건이 한 개면 If-Then-Else를 사용한다. Else 이후 실행할 문이 없으면 생략한다. 보기 11.6.1은 Else 키워드를 생략한 한 줄 구문이다.

### 보기 11.6.1. 【논리가 한 개인 경우 한 문장 실행 조건문】

```

1 Sub IfTest()
2   score = 90
3   If score >= 60 Then MsgBox "당신은 합격입니다."
4 End Sub

```

보기 11.6.1에서 3 줄은 score 변수값이 60 이상여면 Then 다음 문이 실행된다. If-Then-Else 문에서 한 줄 이상 코드 실행은 여러 줄에 문을 사용하고 End If 문을 마지막에 포함한다. 보기 11.6.2는 논리 조건이 한 개이고 실행문이 2 개이다.

### 보기 11.6.2. 【논리가 한 개인 경우 여러 문장 실행 조건문】

```

1 Sub IfEndIfTest()
2   score = 90
3   If score >= 60 Then
4     MsgBox "당신 점수는" & score & "점입니다."
5     MsgBox "당신의 합격을 축하드립니다."
6   End If
7 End Sub

```

보기 11.6.3은 If-Then-Else 문을 사용하여 실행문 블록 두 개를 정의한다. 조건이 True이면 설정한 블록이 실행되고 False이면 다른 블록이 실행된다.

### 보기 11.6.3. 【논리가 한 개인 경우 TRUE, FALSE마다 실행 조건문】

```

1 Sub IfElseTest()
2   score = InputBox("점수를 입력하세요", "점수", 59)
3   If score >= 60 Then
4     MsgBox "당신은 합격입니다."
5   Else
6     MsgBox "당신은 불합격입니다."
7   End If
8 End Sub

```

보기 11.6.3에서 2 줄은 사용자가 입력한 점수를 score에 저장하고, 그 값이 60 이상이면 4 줄, 미만이면 6 줄이 실행되는 코드이다.

첫 번째 조건이 False일 때 두 번째 조건 검사하는 방법은 If-Then-Else 문에 ElseIf 문을 추가한다. 이렇게 설정하면 첫 번째 조건이 False일 때 두 번째 조건을 검사한다. 보기 11.6.4는 score 값에 따라 등급을 결정한다. 만일 2 줄 score에 입력 값이 89점이면 3 줄의 조건이 만족되지 않으니 다음 5 줄 조건이 맞는지 비교하게 된다.

#### 보기 11.6.4. [논리가 두 개 이상인 경우 조건문]

```
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
```

#### 11.6.2 Select-Case 문

식을 다른 여러 값과 비교할 때 If-Then-Else 문의 ElseIf를 사용하는 것과 같이 Select-Case 문을 사용한다. Select-Case 문의 사용법은 다음과 같다.

##### 사용방법 11.6.2. [Select-Case]

```

Select Case testexpression
Case expressionlist
    [statements]
    :
Case Else
    [statements]
End Select

```

위 식에서 인수의 사용은 다음과 같다.

- ◉ *testexpression* : 반드시 입력해야 하는 값으로 수식이나 문자식
- ◉ *testexpressionlist* : Case에 반드시 입력하는 값으로 특정값이나 범위를 지정할 수 있음

Case-Else 문은 Select Case 문이 어느 Case 문의 값과도 일치하지 않을 때 실행된다. 보기 11.6.5는 Select-Case 문을 사용하였다. 내용은 if 문에서 소개한 보기 11.6.4와 동일하다. Select-Case 문은 조건에 맞는 값이 있더라도 모든 Case 문을 비교한다. 반면 If-Then-ElseIf-Then-Else 문은 조건이 맞는 If 문이나 ElseIf 문을 만나면 그 문을 실행하고 나머지는 더 이상 실행하지 않고 If-Then-ElseIf-Then-Else 문을 끝낸다.

#### 보기 11.6.5. 【여러 가지 경우에 대한 조건문】

```

1 Sub SelectCaseTest()
2     score = InputBox("점수를 입력하세요", "점수", 89)
3     Select Case score
4         Case Is >= 90
5             MsgBox "당신의 성적등급은 매우 잘함입니다."
6         Case score 80 to 90
7             MsgBox "당신은 성적등급은 잘함입니다."
8         Case score 70 to 80
9             MsgBox "당신은 성적등급은 보통입니다."

```

```
10 Case score 60 to 70
11   MsgBox "당신은 성적등급은 못함입니다."
12 Case Else
13   MsgBox "당신은 성적등급은 매우 못함입니다."
14 End Select
15 End Sub
```

## 제 11.7 절 반복문

반복문은 특정한 부분에 일정한 조건을 만족하는 동안 실행문을 반복적으로 실행하는 것을 말한다.

### 11.7.1 Do-Loop 문

Do-Loop 문은 설정한 조건을 만족하면 블럭(block) 안의 문을 반복실행한다. 이렇게 문이 반복 실행되는 것을 반복문 또는 루프(loop)라고 부른다. 경우에 따라 설정한 조건을 계속 만족하면 문이 무제한 실행된다.

이 문은 키워드에 따라

- ◉ While : 조건이 True일 때 블록문이 실행

- ◉ Until : 조건이 False일 때 블록문이 실행

한다. While이나 Until Do-Loop 문에서 조건 위치는 두 가지이다. 첫 번째는 Do-Loop 문 처음에 조건을 설정하고 두 번째는 Do-Loop 문 끝에 조건을 설정한다.

Do-Loop 문 처음에 조건 설정은 다음과 같다. 이 방법은 조건이 맞지 않으면 블럭을 전혀 실행하지 않는다.

#### 사용방법 11.7.1. [Do {While | Until} - Loop]

```

Do [{While | Until} condition ]
[statements]
[Exit Do]
Loop

```

Do-Loop 문 끝에 조건 설정은 다음과 같다. 이 방법은 조건이 맞지 않더라도 문을 한 번 실행한다. 프로그래밍 언어는 문을 위에서 아래로 실행하므로 당연한 결과이다.

#### 사용방법 11.7.2. 【Do - Loop {While | Until}】

```

Do
[statements]
[Exit Do]
Loop [{While | Until} condition ]

```

보기 11.7.1의 ChkFirstWhile 프로시저는 Do-Loop 문 처음에 조건이 True인지 False를 검사한다. 3 줄 myNum 변수 값이 20이므로 4 줄 조건 결과가 True이므로 Do-Loop 문이 실행된다. 만일 myNum 변수 값이 9라면 조건 결과가 False이므로 Do-Loop 문은 한 번도 실행하지 않는다.

보기 11.7.1 ChkLastWhile 프로시저는 조건을 반복문 끝에 설정하였으니 조건에 상관없이 반복문을 한 번 실행하고, 설정된 조건이 False가 되기 전까지 블럭을 실행한다.

#### 보기 11.7.1. 【조건이 True일 동안 문 반복】

```

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
11 Sub ChkLastWhile()
12   counter = 0
13   myNum = 9
14   Do
15     myNum = myNum - 1
16     counter = counter + 1
17   Loop While myNum > 10
18   MsgBox "루프를 "& counter &"번 반복하였다."
19 End Sub

```

Do-Loop 문에 Until을 사용하면 조건이 False일 때 반복문이 실행된다. 이 경우도 사용방법이 두 가지이며 사용법도 While과 동일하다. 보기 11.7.2에서 ChkFirstUntil 프로시저는 Do-Loop 문 처음에 조건을 설정하고 ChkLastUntil 프로시저는 Do-Loop 문 끝에 조건을 설정하였다.

### 보기 11.7.2. [조건이 False일 동안 문 반복]

```

1 Sub ChkFirstUntil()
2   counter = 0
3   myNum = 20
4   Do Until myNum = 10
5     myNum = myNum - 1
6     counter = counter + 1
7   Loop
8   MsgBox "루프를 "& counter &"번 반복하였다."
9 End Sub

10
11 Sub ChkLastUntil()
12   counter = 0
13   myNum = 1
14   Do

```

```

15     myNum = myNum + 1
16     counter = counter + 1
17 Loop Until myNum = 10
18 MsgBox "루프를 "& counter &"번 반복하였다."
19 End Sub

```

Do-Loop 문은 Exit Do으로 반복문을 탈출한다. 끝없이 반복되는 루프 탈출은 If-Then-Else 문이나 Select-Case 문에서 설정한 조건이 True일 때 Exit Do 문을 사용한다. 두 문 조건이 False이면 루프가 계속 반복 실행된다.

보기 11.7.3은 3 줄 myNum 값이 9이고 4 줄 조건은 myNum 값이 10이면 루프가 끝난다. 그런데 5 줄은 myNum 값에서 1을 빼기 때문에 무한 루프다. 그래서 7 줄은 If-Then-Else 문을 사용하여 무한 루프를 탈출한다.

### 보기 11.7.3. [Do-Loop 문 빠져 나가기]

```

1 Sub ExitExample()
2     counter = 0
3     myNum = 9
4     Do Until myNum = 10
5         myNum = myNum - 1
6         counter = counter + 1
7         If myNum < 10 Then Exit Do
8     Loop
9     MsgBox "루프를 "& counter &"번 반복하였다."
10    End Sub

```

계속 반복되는 루프는 Esc키나 Ctrl+Break를 누르면 중지된다.

## 11.7.2 For-Each-Next 문

For-Each-Next 문은 컬렉션 개체나 배열 요소마다 문 블럭을 반복한다. 루프가 실행되면 자동으로 변수가 설정된다. 사용법은 다음과 같다.

### 사용방법 11.7.3. [For-Each-Next]

```

For Each element In group
    [statements]
    [Exit For]
    Next element

```

For-Each-Next 문에 사용한 값들은

- ◉ *element* : 컬렉션이나 배열의 요소들을 반복실행에 사용하는 변수
- ◉ *group* : 개체 컬렉션이나 배열이름

이다. 다음 프로시저는 실행 중인 설정한 셀의 내용을 출력하는 것이다.

#### 보기 11.7.4. 【유저폼 닫기 반복】

```

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

```

2 – 3 줄은 Range 개체를 선언하고, 4 줄은 Range 개체의 값을 설정하였다.  
5 – 7 줄은 4 줄의 설정 셀 값을 출력한다.

### 11.7.3 For-Next 문

For-Next 문은 설정한 횟수만큼 문 블럭을 반복한다. For는 루프를 반복할 때마다 값이 증가되거나 감소되는 카운터 변수를 사용한다.

#### 사용방법 11.7.4. 【For-Next】

```

For counter=start To end [Step step ]
    [statements]
[Exit For]
Next counter

```

For-Next 문에 사용한 값은

- ◎ *counter* : 반복문에 루프 카운터로 사용할 숫자 변수명
- ◎ *start* : 반복문에서 시작값
- ◎ *end* : 반복문에서 종료값
- ◎ *step* : 반복문에서 루프에서 변화하는 값으로 생략하면 1

보기 11.7.5는 입력한 문자열을 소리로 출력한다. For 문은 카운터 변수 x와 시작값, 최종값을 지정한다. 5 줄은 변수 x 초기값을 1에서 종료값 3까지 설정한다. 8 줄 Next 문은 카운터 변수를 1씩 증가시킨다.

#### 보기 11.7.5. 【입력한 문자를 소리로 출력하기】

```

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

```

Step 키워드는 지정한 값만큼 카운터 변수를 증가/감소시킬 수 있다. 보기 11.7.6은 루프가 반복될 때마다 카운터 변수 j가 2씩 증가한다. 루프가 끝나면 3 줄 total 변수에 저장된 값은 2, 4, 6, 8, 10의 합계이다.

#### 보기 11.7.6. 【증가치를 반복해서 합계 구하기】

```

1 Sub TwoTotal()
2   For j = 2 To 10 Step 2
3     total = total + j
4   Next j
5   MsgBox "총 "& total &"이다."
6 End Sub

```

카운터 변수에서 값 감소는 Step 값을 음수로 한다. 카운터 변수의 값 감소는 최종값을 시작값보다 작게 지정한다. 보기 11.7.7 2 줄은 카운터 변수 myNum 값이 루프가 반복될 때마다 2 씩 감소된다. 루프가 끝나면 3줄 total 값은 16, 14, 12, 10, 8, 6, 4, 2의 합계이다.

#### 보기 11.7.7. 【감소치를 반복해서 합계 구하기】

```

1 Sub NewTotal()
2   For myNum = 16 To 2 Step -2
3     total = total + myNum
4   Next myNum
5   MsgBox "총 "& total &"이다."
6 End Sub

```

Exit For 문을 사용하면 최종값에 이르기 전에 For-Next 문을 빠져 나온다.

#### 11.7.4 With 문

With 문은 연속되는 문 전체에서 개체나 사용자 정의 형식을 한 번만 지정하여 사용한다. With 문은 프로시저를 빠르게 실행시키고 반복 입력을 피한다. With 문 사용법은 다음과 같다.

#### 사용방법 11.7.5. 【With】

```

With object
  [statements]
End With

```

위 사용법에서

*object*는 개체의 이름이나 사용자 정의 형식이다.

보기 11.7.8은 A1:C30 범위를 30으로 채우고(2 - 3 줄) 지정한 셀의 글꼴 유형은 굵게(4 줄), 지정한 셀 내부 색은 노란색이다(5 줄).

#### 보기 11.7.8. 【개체의 속성값 설정】

```

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

```

With 문은 중첩하여 사용할 수 있다. 보기 11.7.9는 셀 A1에 수식을 넣고 글꼴을 설정한다(2 줄). 3 줄은 셀 값에  $\sqrt{50}$ 을 출력한다. 4 – 8 줄은 셀에 적용하는 글자체이다. 5 줄은 글꼴, 6 줄은 굵은 글씨체, 7 줄은 글자 크기는 8pt이다.

#### 보기 11.7.9. 【여러 개체의 속성값 설정】

```

1 Sub MyInput()
2     With Workbooks("Book1").Worksheets("Sheet1").Cells(1, 1)
3         .Formula = "=SQRT(50)"
4         With .Font
5             .Name = "Arial"
6             .Bold = True
7             .Size = 8
8         End With
9     End With
10 End Sub

```

## 제 11.8 절 연산자

연산자는 숫자나 문자 계산에 사용하며 종류는 다음과 같다.

### 11.8.1 산술 연산자

산술 연산자는 수학 계산하는 연산자로 그 종류는 다음과 같다.

- ◉ ^ 연산자 : 숫자의 지수승을 계산한다.

$2 ^ 5$ 는 ..... 32이다.

- ◉ \* 연산자 : 두 수를 곱한다.

$2 * 5$ 는 ..... 10이다.

- ◉ / 연산자 : 두 수를 나누고 계산 결과는 실수이다.

$2 / 5$ 는 ..... 0.4이다.

- ◉ \ 연산자 : 두 수를 나누고 정수 몫을 계산한다.

$2 \backslash 5$ 는 ..... 0이다.

- ◉ Mod 연산자 : 두 수를 나누고 나머지를 계산한다.

$2 \text{ Mod } 5$ 는 ..... 2이다.

- ◉ + 연산자 : 두 수의 합이다.

$2 + 5$ 는 ..... 7이다.

- ◉ - 연산자 : 두 수의 차를 계산하거나 음수에 사용한다.

$2 - 5$ 는 ..... -3이다.

### 11.8.2 비교 연산자

비교 연산자는 비교 조건을 TRUE과 FALSE로 계산한다.

- ◉ < : 첫 번째 값이 두 번째 값보다 작으면 TRUE이고 그렇지 않으면 FALSE이다.

$2 < 5$ 는 ..... TRUE이다.

- ◉  $\leq$  : 첫 번째 값이 두 번째 값보다 작거나 같으면 TRUE이고 그렇지 않으면 FALSE이다.

$2 \leq 5$ 는 ..... TRUE이다.

- ◉  $>$  : 첫 번째 값이 두 번째 값보다 크면 TRUE이고 그렇지 않으면 FALSE이다.

$2 > 5$ 는 ..... FALSE이다.

- ◉  $\geq$  : 첫 번째 값이 두 번째 값보다 크거나 같으면 TRUE이고 그렇지 않으면 FALSE이다.

$2 \geq 5$ 는 ..... FALSE이다.

- ◉  $=$  : 첫 번째 값과 두 번째 값이 같으면 TRUE이고 그렇지 않으면 FALSE이다.

$2 = 5$ 는 ..... FALSE이다.

- ◉  $\neq$  : 첫 번째 값과 두 번째 값이 다르면 TRUE이고 그렇지 않으면 FALSE이다.

$2 \neq 5$ 는 ..... FALSE이다.

- ◉ Like : 두 문자열을 비교하는 연산자이다.

"한국" Like "한국" 은 ..... FALSE다.

- ◉ Is : 두 개체 변수가 동일한지 비교한다.

Range("A1") Is Range("A2")는 ..... FALSE다.

### 11.8.3 문자열 연결 연산자

문자열 연결 연산자는 문자열을 합한다.

- ◉ & : 두 값을 연결하여 붙인다.

"우리" & "나라"는 ..... "우리나라"이다.

- ➊ + : 두 수를 더한다.

$2+5$ 는 ..... 7이다.

#### 11.8.4 논리 연산자

논리 연산자는 논리적인 작업을 수행하는 연산자로 그 종류는 다음과 같다.

논리1	논리2	And	OR	Eqv	Imp	Xor
TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE
TRUE	FALSE	FALSE	TRUE	FALSE	FALSE	TRUE
FALSE	TRUE	FALSE	TRUE	FALSE	TRUE	TRUE
FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	FALSE

표 11.8.1: 두 논리값에 대한 논리 연산 결과

- ➊ And : 두 식 조건의 논리곱을 계산한다. 두 식 논리가 모두 TRUE면 TRUE 그 이외는 FALSE이다.

$2 > 5$  And  $2 < 5$ 는 ..... FALSE다.

- ➋ Eqv : 두 식 조건이 논리적으로 같은지 비교한다. 두 식 논리가 모두 같으면 TRUE 다르면 FALSE

$2 > 1$  Eqv  $2 < 1$ 는 ..... FALSE다.

- ➌ Imp : 두 식의 논리적 함축성을 계산한다. 첫 번째 논리가 TRUE이고 두 번째 논리가 FALSE이면 FALSE이고 나머지는 모두 TRUE이다.

Imp는 ..... TRUE이다.

- ➍ Not : 논리적 부정을 구한다. 식 논리가 FALSE이면 결과는 TRUE이고 TRUE 이면 FALSE이다.

Not  $2 > 5$ 는 ..... TRUE이다.

- Or : 두 식의 논리합을 구한다. 두 식 논리가 모두 FALSE이면 FALSE 나머지는 TRUE이다.

$2 > 5 \text{ Or } 2 < 5$ 는 ..... TRUE이다.

- Xor : 두 식이 논리적으로 다른지 비교한다. 두 식 논리가 다르면 TRUE 두 식의 논리가 같으면 FALSE가 된다.

$2 > 5 \text{ Xor } 2 < 5$ 는 ..... TRUE이다.

## 제 12 장

# VBA 프로시저

VBA에서 프로시저가 모든 문의 실행을 하나의 처리단위로 한다. 이 장은 프로시저 작성과 종류에 대하여 알아본다. 프로시저는 문을 실행하거나 변수 선언, 변수 생성, 변수에 값 대입하는 공간이다. 종류는

- ◉ Sub 프로시저
- ◉ Function 프로시저
- ◉ Property 프로시저

가 있다.

### 제 12.1 절 Sub 프로시저

Sub 프로시저는 동작은 수행하지만 리턴값이 없는 Sub와 End Sub 사이에 작성하는 Visual Basic 문이다. Sub 프로시저는 호출 프로시저에 의해 전달된 상수, 변수, 식 같은 인수를 가질수 있다. Sub 프로시저의 구성은

#### 사용방법 12.1.1. [Sub]

```
[Public | Private | Friend] [Static] Sub name [(arglist)]
[statements]
[Exit Sub]
[statement]
End Sub
```

이다. Sub 프로시저에 인수가 없으면 Sub 문에는 공백 팔호를 적는다. 다음 보기는 Sub 프로시저에 If 문을 사용하였다.

#### 보기 12.1.1. 【Sub 프로시저】

```
1 Sub GetInfo()
2 Dim answer As String
3 answer = InputBox(Prompt:="이름을 입력하십시오.")
4 If answer = Empty Then
5     MsgBox Prompt:="이름을 입력하지 않았다."
6 Else
7     MsgBox Prompt:="이름은 "& answer &"이다."
8 End If
9 End Sub
```

3 줄에 사용자가 입력한 값을 저장한다. 4 줄은 이름을 입력하지 않았으면 5 줄이 실행되고 이름을 입력하면 7 줄이 실행된다. 그림 12.1.1에 입력값이 “홍길동”일 때 결과이다. 이미 만든 Sub 프로시저 실행은

- 프로시저 이름과 선언한 데이터 형식으로 인수를 입력한다.
- Call 문을 사용한다. 이 때 모든 인수는 팔호로 묶는다.

다음 보기는 HouseCalc 프로시저를 만들고 Main 프로시저에서 만든 프로시저를 실행한다. 2 줄에서 Sub 프로시저는 리턴값이 없으니 팔호를 묶지 않고 실행한다. 3 줄은 Call 문을 사용하였으니 인수를 팔호로 묶었다. 2 줄과 3 줄 실행결과는 그림 12.1.2에서 확인한다.

#### 보기 12.1.2. 【Sub 프로시저 호출(인수 2 개)】

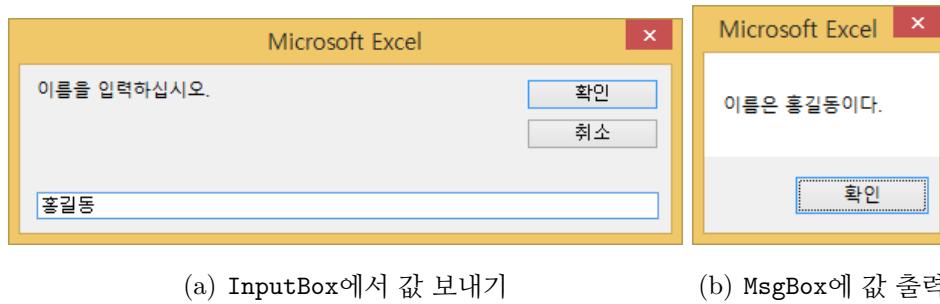


그림 12.1.1: Sub 프로시저

```

1 Sub Main()
2     HouseCalc 99800, 43100
3     Call HouseCalc(380950, 49500)
4 End Sub
5 Sub HouseCalc(price As Single, wage As Single)
6     If 2.5 * wage <= 0.8 * price Then
7         MsgBox "이 집을 살 여유가 없다."
8     Else
9         MsgBox "이 집을 살 여유가 있다."
10    End If
11 End Sub

```

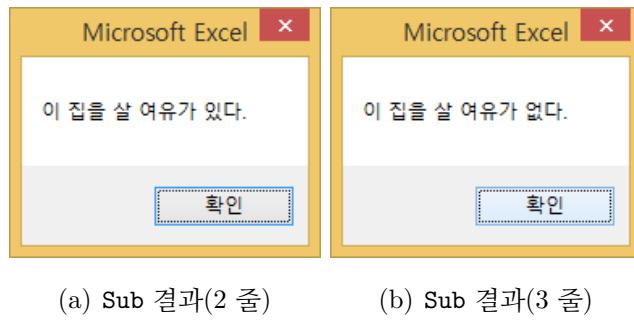


그림 12.1.2: Sub 프로시저

## 제 12.2 절 Function 프로시저

Function 프로시저는 동작을 수행한 결과 값을 반환한다. 실행문은 Function과 End Function 사이에 작성한다. Function 프로시저의 구성은

### 사용방법 12.2.1. [Function]

```
[Public | Private | Friend] [Static] Function name[(arglist)]
    [As type]
    [statements]
    [name = expression]
    [Exit Function]
    [statement]
    [name = expression]
End Sub
```

이다. Function 프로시저는 상수, 변수, 식 같은 인수를 가질 수 있으며 리턴 값의 데이터 형식을 지정해야 한다. 리턴 값의 데이터 형식을 생략하면 데이터 형식은 Variant이다. 리턴 값은 Function에 붙여진 프로시저 이름과 같다. Function 프로시저 실행은 결과값이 있어서 인수를 괄호로 묶어야 한다. 다음 보기에는 3개 항목으로 승진 점수 계산하는 Function 프로시저이다.

### 보기 12.2.1. [Function 프로시저]

```
1 Sub 점수보기()
2     MsgBox 승진점수(15, 90, 2)
3 End Sub
4 Function 승진점수(년 As Integer, 근평 As Integer, 학위 As Integer)
5     career = 년 / 20 * 70
6     If 학위 = 0 Then
7         deg = 0
8     ElseIf 학위 = 1 Then
```

```

9      deg = 1.5
10     ElseIf 학위 = 2 Then
11       deg = 3
12     End If
13     승진점수 = 캐리어 + 균평 + deg
14   End Function

```

보기 12.2.1에서 승진점수 계산결과는 Sub 프로시저가 있는 2 줄에서 확인한다. 승진점수 계산은 4 – 14 줄까지 Function 프로시저이다. 승진점수 프로시저에서 첫 번째 인수는 근무년수, 두 번째 인수는 근무평정, 세 번째 인수는 최종학력이다. 근무년수는 20년이 70점 만점이고, 근무평정은 100 점이 만점, 학위는 0은 대졸이하, 1은 석사, 2는 박사학위이다. 승진점수 계산식은 교원승진 점수를 참고하였다. 13 줄 승진점수는 4 줄 승진점수와 같아야 function 프로시저가 작동한다. 2 줄에서 근무년수 15년, 균평 90점, 박사학위를 인수로 계산하였다. 계산결과는 그림 12.2.1에 있다.

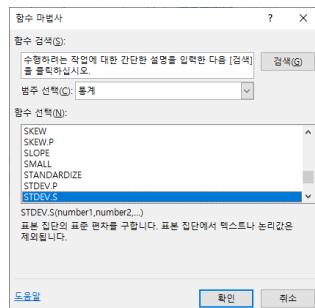


그림 12.2.1: 승진 점수

## 제 12.3 절 Property 프로시저

Property 프로시저는 개체 속성을 정의한다. 종류는 다음과 같다.

- Property Let 프로시저 : 선언한 데이터 변수에 값을 할당한다.

- Property Get 프로시저 : Let, Set에서 할당된 데이터 형식 변수의 값이나 개체 변수에 속성값을 읽는다.
- Property Set 프로시저 : 선언한 개체 변수의 속성에 값을 할당한다. 개체 변수는 Object 변수와 Variant 변수가 있다.

Get 프로시저 사용법이다.

#### 사용방법 12.3.1. [Get]

```
[Public | Private | Friend] [Static] Property Get      name
[(arglist)] [As type]
[statements]
[name = expression]
[Exit Property]
[statements]
[name = expression]
End Property
```

- *statements*는 실행문이다.
- *name = expression*에서 *name*은 Get 옆에 있는 *name*과 반드시 동일해야 한다. *expression*은 *name*에 값을 할당한다.
- Exit Property는 프로시저를 종료한다.

Let 프로시저 사용법이다.

#### 사용방법 12.3.2. [Let]

```
[Public | Private | Friend] [Static] Property Let      name  
([arglist,] value)  
[statements]  
[Exit Property]  
[statements]  
End Property
```

- ◉ *statements*는 실행문이다.
- ◉ Exit Property는 프로시저를 종료한다.

Set 프로시저 사용법이다.

#### 사용방법 12.3.3. 【Set】

```
[Public | Private | Friend] [Static] Property Set      name  
([arglist,] value)  
[statements]  
[Exit Property]  
[statements]  
End Property
```

- ◉ *statements*는 실행문이다.
- ◉ Exit Property는 프로시저를 종료한다.

프로시저는 다른 곳에서 접근을 제한하는(access modifier) 키워드가 있다.

- ◉ Private : 같은 모듈 안에서만 접근이 가능하다.

- ◉ **Public** : 모든 곳에서 접근이 가능하다.
- ◉ **Friend** : 프로젝트 안에서 접근이 가능하다. 이 키워드는 클래스 모듈에 서만 사용할 수 있다.
- ◉ **Static** : 프로시저 실행은 그 안의 변수값이 매번 초기화되어 계산되지 만 이 키워드는 최근 값을 기억하여 계산에 반영한다.

#### 보기 12.3.1. 【Static 제한자】

```

1 Function updateSales(ByVal thisSale As Decimal) As Decimal
2     Static totalSales As Decimal = 0
3     totalSales += thisSale
4     Return totalSales
5 End Function

```

보기 12.3.1에 사용한 `totalSales` 변수는 `Static` 제한자로 선언하여 처음 `updateSales` 함수 실행시 초기값이 0이 된다. 매번 이 함수를 실행하면 최근에 저장된 `totalSales` 값이 계산한다.

`Property` 프로시저와 접근 제한자 관계를 설명하려면 클래스 모듈 개념이 필요하다. 클래스 모듈과 모듈은 10.3 절에 이미 소개하였다. 그 내용은 클래스 모듈을 모듈에서 개체 변수로 선언하여 사용한다. 다른 곳에서 사용할 수 있게 프로그래밍한 것을 클래스라 하고 클래스를 변수로 선언하면 개체라고 부른다.

#### 보기 12.3.2. 【클래스 모듈】

```

1 Private myName As String
2 Private myIncome As Integer
3
4 Public Property Get Name() As String
5     Name = myName
6 End Property
7 Public Property Let Name(vName As String)
8     myName = vName

```

```
9 End Property  
10  
11 Public Property Get Income() As Integer  
12     Income = myIncome  
13 End Property  
14 Public Property Let Income(vIncome As Integer)  
15     myIncome = vIncome  
16 End Property  
17  
18 Public Function Grade() As String  
19     Dim strGrade As String  
20     If myIncome >= 1000 Then  
21         strGrade = "상류층"  
22     ElseIf myIncome >= 600 Then  
23         strGrade = "중산층"  
24     Else: strGrade = "서민층"  
25     End If  
26     Grade = strGrade  
27 End Function
```

보기 12.3.2에서 1 – 2 줄 변수는 Private 접근 제한자로 선언하여 이 클래스 내에서만 접근이 가능하다. 8 줄은 인수 vName 값을 같은 클래스 변수 myName에 배정한다. 같은 클래스에 있기에 Private 접근 제한자로 선언한 변수도 접근하여 값을 변경한다. 15 줄도 myIncome 변수에 동일한 방법으로 접근한다. 이렇게 Property Let 프로시저는 이미 선언한 변수에 값을 배정한다.

반면 Property Get 프로시저는 할당된 값을 가져온다. 5 줄은 Private 접근 제한자로 선언한 myName 변수에 저장된 값을 가져온다. 12 줄도 myIncome에 할당된 값을 가져온다.

18 – 27 줄은 15 줄을 실행하여 myIncome 변수에 할당된 값이 1000 이상이면 상류층, 600 이상이면 중산층, 그 이외는 서민층으로 판정하는 Function 프로시저이다.

보기 12.3.2 클래스 모듈에 구현한 Property 프로시저는 다른 모듈에서

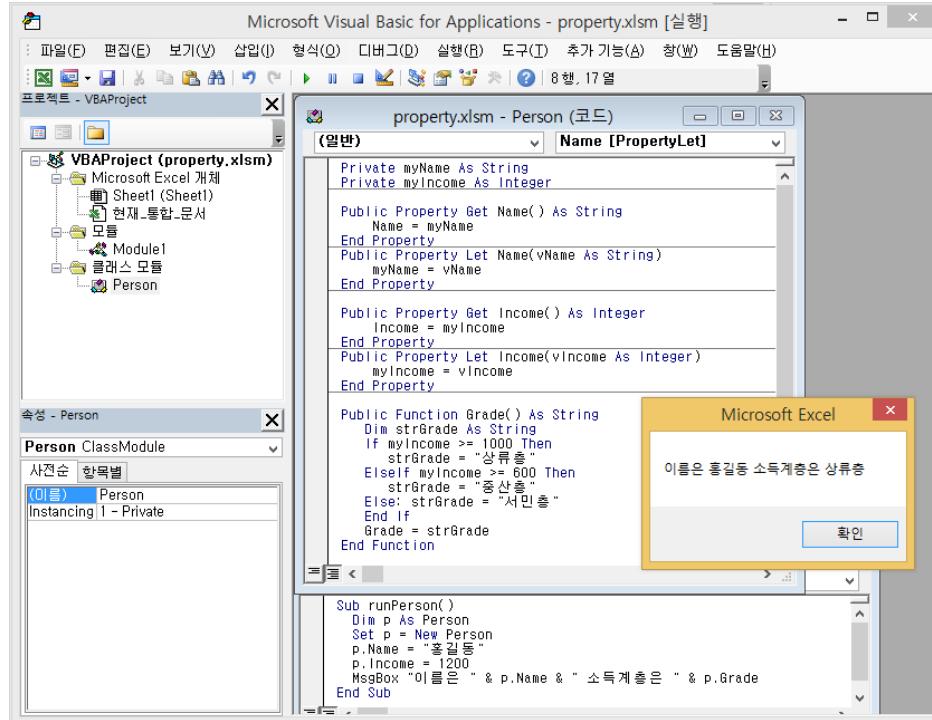


그림 12.3.1: Property 프로시저

접근해야 되므로 Public 접근 제한자로 선언한다. 보기 12.3.3은 모듈에서 클래스를 개체 변수로 선언하였다.

### 보기 12.3.3. 【모듈】

```

1 Sub runPerson()
2     Dim p As Person
3     Set p = New Person
4     p.Name = "홍길동"
5     p.Income = 1200
6     MsgBox "이름은 " & p.Name & " 소득계층은 " & p.Grade
7 End Sub

```

2 줄은 Person 클래스를 p 개체로 만든다. 3 줄은 선언한 p 개체를 초기화한다. 개체 변수 선언은 Set 문을 사용하고 클래스 이름 앞에 New 키워드를 붙인다.

4 줄은 p 개체 Name 속성에 흥길동을 배정한다. 5 줄도 p 개체 Income 속성에 1200을 배정한다. 6 줄에서 p 개체에 저장된 값을 MsgBox로 출력한다. 그럼 12.3.1에 클래스 모듈과 모듈의 코드와 출력이 있다.

## 제 12.4 절 인수 전달 방법

프로시저에서 인수 전달은 ByVal과 ByRef 두 가지가 있다. ByVal은 값에 의한 전달(call by value), ByRef는 참조에 의한 전달(call by reference)이다.

### 12.4.1 ByVal 인수 전달

ByVal은 인수 전달을 값(value)으로 한다. 값 인수 전달은 값을 넘기는 실행 프로시저와 값을 받는 프로시저가 각각 다른 메모리 영역에 값을 저장한다. 그래서 프로시저에서 값이 바꾸어도 원래 값이 변경되지 않는다.

#### 보기 12.4.1. [ByVal 인수 전달]

```
1 Sub 실행()
2     Dim num As Integer
3     MsgBox num, , "실행 시작 부분"
4     Call 값전달(num)
5     MsgBox num, , "실행 끝 부분"
6 End Sub
7 Sub 값전달(ByVal num As Integer)
8     MsgBox num, , "값전달 시작 부분"
9     num = num + 100
10    MsgBox num, , "값전달 끝 부분"
11 End Sub
```

보기 12.4.1은 값으로 인수 전달하는 프로시저를 실행하였다. 7 줄은 값전달 프로시저에서 인수 전달 방법을 ByVal로 하였다. 2 줄은 정수형 변수 num를 생성하였다. 초기값은 0이 배정된다. 3 줄은 MsgBox 제목표시줄에 실행 시작 부분이 num에 0을 출력한다. 4 줄은 값전달 프로시저를 실행한다. 그러면 8 줄

MsgBox 제목표시줄에 값전달 시작 부분이 num에 0이 출력되고 9 줄에 num에 100을 저장하고 MsgBox 제목표시줄에 값전달 끝 부분이 num에 100이 출력된다. 인수로 값을 전달했기에 4 줄을 실행해도 num 값이 변경하지 않는다. 5 줄은 MsgBox 제목표시줄에 실행 끝 부분이 num에 0을 출력한다. 그림 12.4.1에 실행 순서로 MsgBox 출력 결과가 있다.

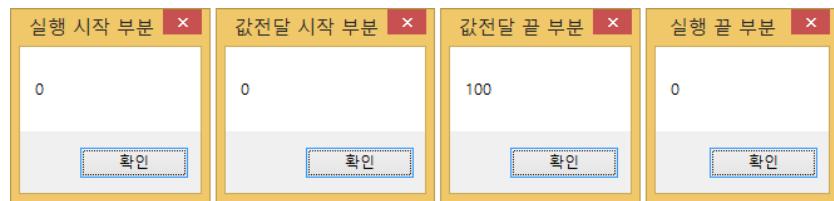


그림 12.4.1: ByVal 인수 전달

#### 12.4.2 ByRef 인수 전달

ByRef는 인수 전달을 참조(reference)로 한다. 함수에서 참조 인수 전달은 동일한 메모리 주소에 전달값을 두고 값을 주고받는 호출 방법이다. 변수는 메모리 주소와 메모리 주소가 가르키는 값이 있다. 변수 값은 메모리 주소가 가르키는 값을 변경해야 바뀐다. 함수에서 참조 인수 전달은 인수가 저장된 메모리 주소를 넘긴다. 그래서 실행하는 프로시저와 받는 프로시저가 동일한 메모리 주소를 사용한다. 프로시저 호출에서 변수 값 변경은 인수 전달 방법을 ByRef로 사용한다. 인수 전달 방법을 생략하면 기본값은 ByRef이다.

##### 보기 12.4.2. [ByRef 인수 전달]

```

1 Sub 실행()
2     Dim num As Integer
3     MsgBox num, , "실행 시작 부분"
4     Call 참조전달(num)
5     MsgBox num, , "실행 끝 부분"
6 End Sub
7 Sub 참조전달(ByRef num As

```

```

8   MsgBox num, , "참조전달 시작 부분"
9   num = num + 100
10  MsgBox num, , "참조전달 끝 부분"
11 End Sub

```

보기 12.4.2는 참조로 인수 전달하는 프로시저를 실행하였다. 7 줄은 참조전달 프로시저에서 인수 전달 방법을 ByRef로 하였다. 2 줄은 정수형 변수 num를 생성하였다. 초기값은 0이 배정된다. 3 줄은 MsgBox 제목표시줄에 실행 시작 부분이 num에 0을 출력한다. 4 줄은 참조전달 프로시저를 실행한다. 그러면 8 줄 MsgBox 제목표시줄에 참조전달 시작 부분이 num에 0이 출력되고 9 줄에 num에 100을 저장하고 MsgBox 제목표시줄에 참조전달 끝 부분이 num에 100이 출력된다. 인수로 참조를 전달했기에 4 줄을 실행하고 num 값이 변경된다. 5 줄은 MsgBox 제목표시줄에 실행 끝 부분이 num에 100을 출력한다. 그림 12.4.2에 실행 순서로 MsgBox 출력 결과가 있다.

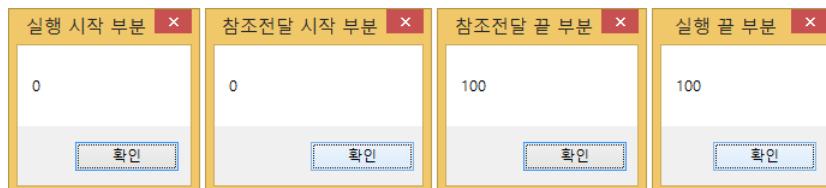


그림 12.4.2: ByRef 인수 전달

## 제 12.5 절 이벤트 프로시저

이벤트(event)는 개체에 마우스를 클릭, 드래그 행동이나 키보드에서 버튼 클릭, 통합문서 시작과 종료하는 행위이다. 이벤트 프로시저는 이벤트가 발생할 때 사용자가 입력한 코드를 실행한다. 더 많은 이벤트 내용은 9 장에 있으니 참고하자.

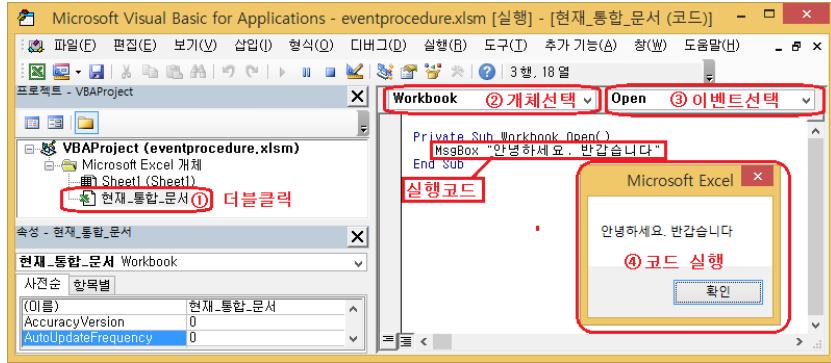


그림 12.5.1: Workbooks 이벤트 프로시저

### 12.5.1 Workbooks 개체와 이벤트 프로시저

Workbooks 개체는 305 면 10.4.1 절에서 소개하였으니 이 절에서는 이벤트 프로시저에 대하여 알아보자. Workbooks 개체 이벤트 프로시저에 실행 코드를 추가하고 실행하자.

1. 이벤트 프로시저는 개체를 클릭하고 이벤트 프로시저를 선택하면 자동으로 만들어진다. 그림 12.5.1은 파일을 열었을 때 이벤트 프로시저가 실행된다. 프로젝트 창에서 현재\_통합\_문서 ①을 클릭하면 코드 창에 Workbooks 개체와 기본 이벤트 프로시저가 보인다.
2. 개체 선택은 ②에서 한다. Workbook 개체 선택과 동시에 기본 이벤트 프로시저 Open이 만들어진다. Open 프로시저는 엑셀 파일을 열면 실행된다. 이 프로시저에 그림 12.5.1의 실행 코드를 추가한다.
3. Workbook 개체는 여러 프로시저가 있고 ③을 누르면 보이고 추가도 가능하다.
4. ④는 엑셀 파일을 열면 Open 프로시저가 실행된 결과이다.

엑셀 시트에 명령 단추를 추가하고 클릭하는 이벤트가 발생할 때 MsgBox 창에 환영합니다가 출력되도록 해보자.

1. 그림 12.5.2는 ActiveX 컨트롤에서 명령 단추를 시트에 추가하였다.

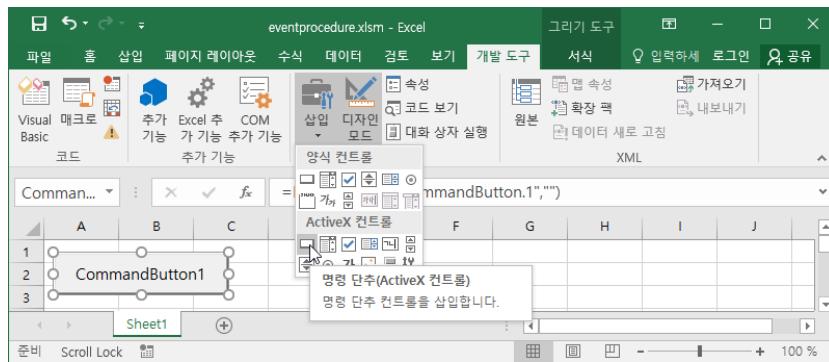


그림 12.5.2: 엑셀 시트에 명령 단추 추가

2. 그림 12.5.3은 사용 가능한 개체를 확인한다. 속성 창에서 ①에 있는 Sheet1을 누르면 사용 가능한 개체가 ②에 나타난다. 그림 12.5.2에서 추가한 명령 단추가 보인다.

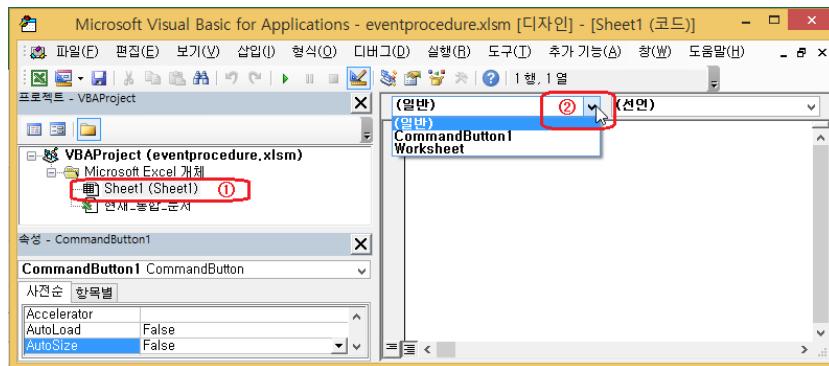


그림 12.5.3: 이용 가능한 개체

3. 그림 12.5.4는 CommandButton1을 선택하면 명령 단추 이벤트 프로시저 Click이 자동으로 추가된다. 자동으로 만들어지는 프로시저 이름은 변수

명\_이벤트명이다. 따라서 CommandButton1\_Click 프로시저가 자동으로 만들어진다. 이 프로시저에 보기 12.5.1 내용을 추가한다.

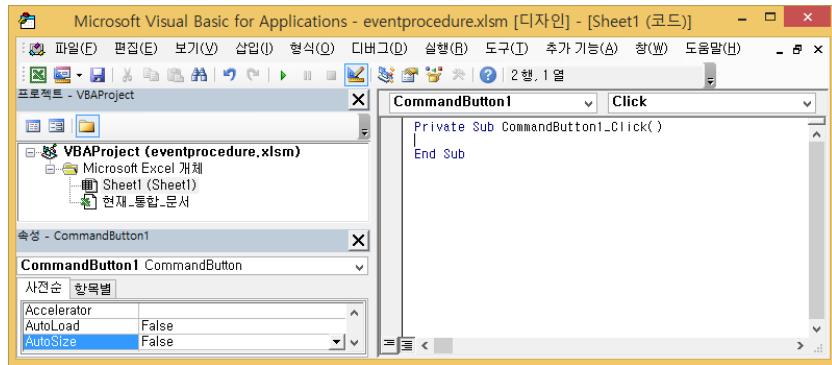


그림 12.5.4: 이벤트 프로시저 추가

### 보기 12.5.1. [명령 단추 이벤트]

```

1  Private Sub CommandButton1_Click()
2      MsgBox "버튼을 누르셨네요. 환영합니다."
3  End Sub

```

4. 그림 12.5.5는 명령 단추에서 사용할 수 있는 이벤트 종류이다.
5. 그림 12.5.6은 명령 단추를 실행하여 MsgBox에 메시지가 출력되었다.  
명령 단추 수정은 디자인 모드를 누르고 진행한다.

### 12.5.2 Worksheets 개체와 이벤트 프로시저

Worksheets 개체는 308 면 10.4.2 절에서 소개하였다. 이 절에서는 Worksheets 이벤트 프로시저에 대하여 알아보자.

1. 그림 12.5.7은 Worksheets 이벤트 프로시저 설정이다. 프로젝트 창에서 Sheet1 (Sheet1) ①을 더블클릭하면 코드 창에 Worksheets 개체와 기본 이벤트 프로시저 SelectionChange가 보인다.

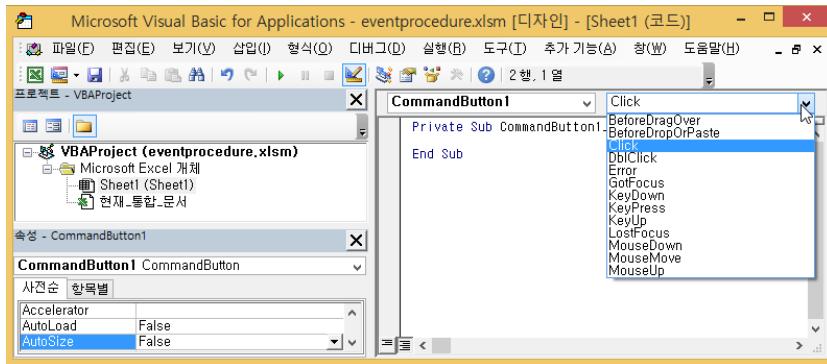


그림 12.5.5: 이벤트 프로시저 보기

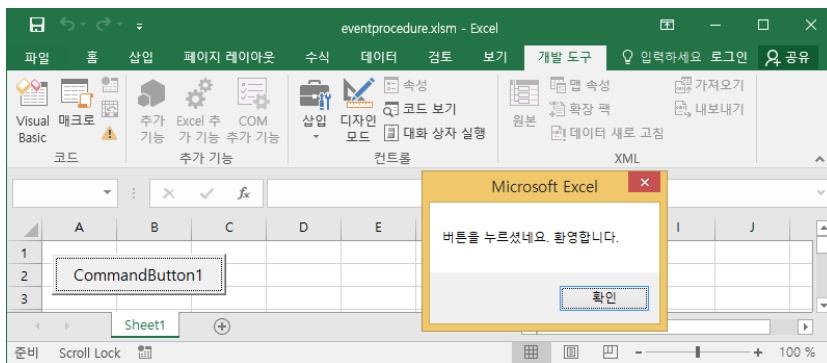


그림 12.5.6: 이벤트 프로시저 보기

2. 개체 선택은 ②에서 한다. Worksheets 개체 선택과 동시에 기본 이벤트 프로시저 SelectionChange이 만들어진다. SelectionChange 프로시저는 시트에서 셀 선택을 변경하면 실행된다. 이 프로시저는 지우고 그림 12.5.7과 같이 이벤트 프로시저를 선택하고 그곳에 실행 코드를 추가한다.
3. Worksheets 개체는 여러 이벤트 프로시저가 있으며 ③을 누르면 보이고 추가도 가능하다.

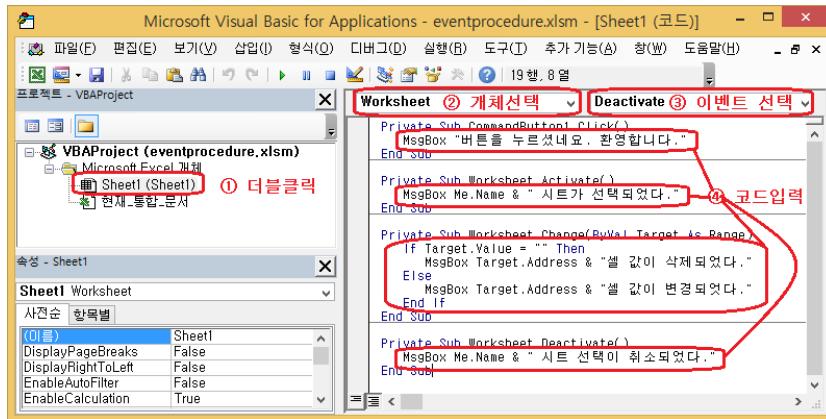


그림 12.5.7: Worksheets 이벤트 프로시저

4. ④는 이벤트 프로시저에 코드를 입력한다. 프로시저 코드는 보기 12.5.2, 보기 12.5.3, 보기 12.5.4이다.

보기 12.5.2는 Sheet1 시트를 선택하면 발생하는 이벤트 프로시저이다. 2 줄 Me는 현재 개체이다. Sheet1 시트를 선택하면 Sheet1 시트가 선택되었다는 메시지가 출력된다.

#### 보기 12.5.2. 【워크시트 이벤트】

```

1 Private Sub Worksheet_Activate()
2     MsgBox Me.Name & " 시트가 선택되었다."
3 End Sub

```

보기 12.5.3은 Sheet1 시트를 선택 취소하면 발생하는 이벤트 프로시저이다. Sheet1 시트 선택을 취소하면 Sheet1 시트 선택이 취소되었다는 메시지가 출력된다.

#### 보기 12.5.3. 【워크시트 이벤트】

```

1 Private Sub Worksheet_Deactivate()
2     MsgBox Me.Name & " 시트 선택이 취소되었다."
3 End Sub

```

보기 12.5.4는 Sheet1 시트의 셀 값이 변경되면 발생하는 이벤트 프로시저이다. 2 줄은 셀 값이 변경되었는데 그 셀이 비었으면 3 줄 셀 값이 삭제되었다. 5 줄 셀 값은 변경되었는데 빈 셀이 아니면 셀 값이 변경되었다. 메시지가 출력된다.

#### 보기 12.5.4. [워크시트 이벤트]

```

1 Private Sub Worksheet_Change(ByVal Target As Range)
2     If Target.Value = "" Then
3         MsgBox Target.Address & "셀 값이 삭제되었다."
4     Else
5         MsgBox Target.Address & "셀 값이 변경되었다."
6     End If
7 End Sub

```

그림 12.5.2에서 개발도구→컨트롤→삽입 메뉴에 컨트롤은 양식 컨트롤과 ActiveX 컨트롤이 있다. 두 컨트롤의 차이는 다음과 같다.

- ◉ 양식 컨트롤은 매크로 연결에 사용하고 ActiveX 컨트롤은 이벤트 프로시저로 사용한다
- ◉ 양식 컨트롤은 개체에 마우스를 옮기고 오른쪽 버튼을 클릭하여 편집한다. ActiveX 컨트롤은 디자인 모드를 선택하고 개체와 속성을 편집한다.

그림 12.5.8은 양식 컨트롤과 ActiveX 컨트롤에서 각각 명령 단추를 추가했다. 양식 컨트롤 명령 단추는 직접 작성한 매크로 보기 12.5.5와 연결하였다. ActiveX 컨트롤 명령 단추는 실행하면 보기 12.5.1에 작성한 이벤트 프로시저가 실행된다.

#### 보기 12.5.5. [직접 작성한 매크로]

```

1 Sub button()
2     MsgBox "단추 1을 누르셨네요."
3 End Sub

```

Worksheets 개체에 포함된 이벤트 프로시저 종류는 다음과 같다.

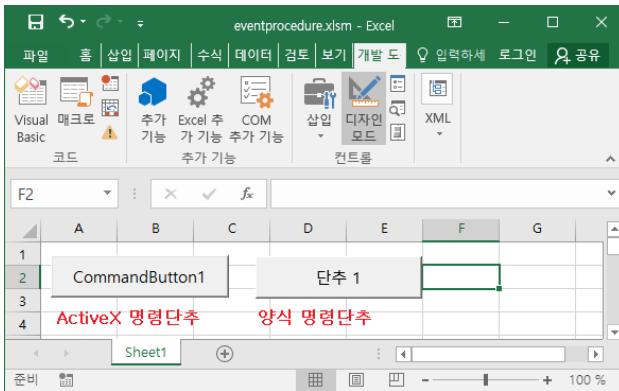


그림 12.5.8: 양식 컨트롤과 ActiveX 컨트롤

- ◉ Activate 시트가 활성화하면 발생하는 이벤트
- ◉ BeforeDelete 시트가 삭제하기 전 발생하는 이벤트
- ◉ BeforeDoubleClick 시트에서 셀을 더블클릭했을 때 발생하는 이벤트
- ◉ Calculate 시트의 셀에서 수식 계산하면 발생하는 이벤트
- ◉ Change 시트의 셀 값이 변경하면 발생하는 이벤트
- ◉ Deactivate 시트가 비활성화되면 발생하는 이벤트
- ◉ PivotTableUpdate 피벗테이블이 변경하면 발생하는 이벤트
- ◉ SelectionChange 시트에서 셀 선택이 변경하면 발생하는 이벤트

## 제 12.6 절 VBA 프로젝트에 암호 설정

Visual Basic 편집기에 입력한 코드는 다른 사람이 보지 못하게 설정할 수 있다.

1. 프로젝트 암호 설정은 그림 12.5.9와 같이 대상 프로젝트에서 마우스 오른쪽 클릭하고 VBAProject 속성 메뉴를 선택한다.

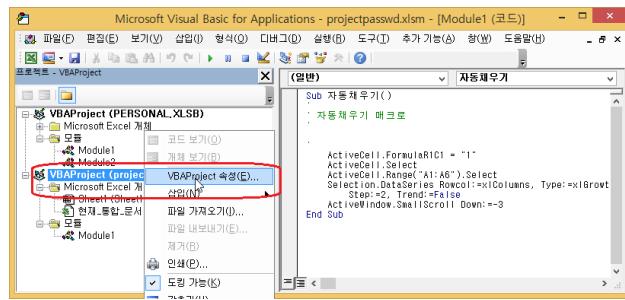


그림 12.5.9: VBA 프로젝트에 암호 설정1

2. 그림 12.6.1은 VBAProject 프로젝트 속성 창이다. 이 창에서 보호 탭을 선택한다.
  - ◉ 읽기 전용으로 프로젝트 잠금 : 다른 사람에게 매크로 실행 기능만 사용할 수 있다.
  - ◉ 프로젝트 속성 보기 암호 : 프로젝트 속성 보기 암호는 암호와 암호 확인에 문자열을 동일하게 입력한다. 암호는 읽기 암호만 설정해도 되고 쓰기 암호도 설정할 수 있다. 읽기는 코드 내용을 확인하는 기능이고 쓰기는 해당 파일을 편집하고 저장할 수 있다.
3. 그림 12.6.2와 같이 암호가 설정된 엑셀 파일의 코드 확인은 ①을 더블 클릭한다. ②에 이미 설정한 암호를 입력한다. 입력한 암호가 일치하면 코드 내용이 보인다. 쓰기암호까지 해제하였으면 내용 편집이 가능하다.

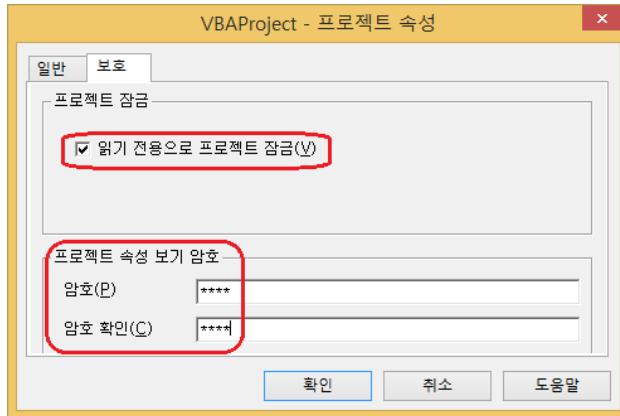


그림 12.6.1: VBA 프로젝트에 암호 설정2

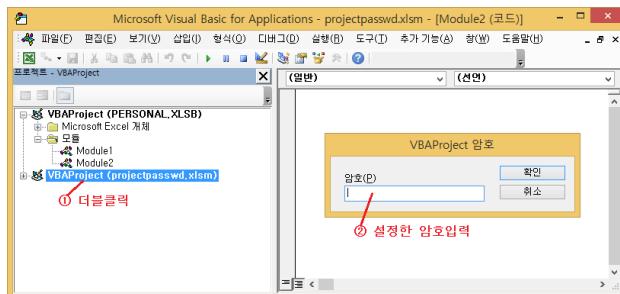


그림 12.6.2: VBA 프로젝트에 암호 설정3

# 찾아보기

$\chi^2$ 검정, 144	ALT + F4, 4
(으)름), 282	CTRL + ↓, 32
., 304	CTRL + ←, 32
-, 52	CTRL + →, 32
-, 351	CTRL + ↑, 32
<=, 53, 352	CTRL + :, 31, 88
<, 53, 351	CTRL + C, 58
<>, 53, 352	CTRL + SHIFT + :, 31, 88
>=, 53, 352	CTRL + SHIFT + ENTER, 105
>, 53, 352	CTRL + V, 58
', 62, 333	CTRL + X, 58
*, 43, 52, 351	SHIFT + ↓, 32
+, 52, 351, 353	SHIFT + ←, 32
,, 53, 54, 87	SHIFT + →, 32
-, 87	SHIFT + ↑, 32
., 331	#, 331
/, 52, 87, 351	# (표시형식), 38
0 (표시형식), 38	\$, 331
: , 53, 54, 87, 332	%, 52
;, 36	&, 53, 331, 352
=, 51, 53, 352	~, 52, 351
? (표시형식), 38	⊥, 332
ALT + ENTER, 44	¬, 43

- !, 62, 331  
 ©, 42, 331  
 \, 351  
 \(문자열), 41  
 이름상자, 29  
 ‘‘’(문자열), 41  
 A1, 29, 60  
**Accelerator**, 217  
**ACOS**, 113  
**Activate**, 372, 374  
 ActiveX 컨트롤, 217, 373  
**Add**, 303, 305, 307  
**AND**, 75  
**And**, 353  
**Array**, 227, 235, 336  
**ASIN**, 112  
**ATAN**, 114  
**AutoLoad**, 217  
**AutoSize**, 218  
**AutoTab**, 285  
**AVERAGE**, 84, 120  
**AVERAGEA**, 120  
**AVERAGEIF**, 84, 85  
**AVERAGEIFS**, 86  
**BackColor**, 218, 276, 282  
**BackStyle**, 218  
**BeforeDelete**, 374  
**BeforeDoubleClick**, 374  
**BeforeDragOver**, 223  
**BeforeDropOrPaste**, 223  
**BETA.DIST**, 133  
**BETA.INV**, 134  
**BINOM.DIST**, 130  
**BINOM.INV**, 131  
**Bold**, 309  
**Boolean**, 327  
**Borders**, 314  
 box – whisker plot, 179  
 box plot, 179  
**ByRef**, 366  
**Byte**, 327  
**ByVal**, 365  
**Calculate**, 374  
**Call**, 356  
**Caption**, 218, 240, 276, 282  
 cell, 6, 29  
**Cells**, 307, 313  
**Change**, 228, 245, 247, 250, 373, 374  
**CHAR**, 98  
**CheckBox**, 281  
**CHISQ.DIST**, 135  
**CHISQ.INV**, 135  
**CHISQ.TEST**, 144  
**CHOOSE**, 152  
**Choose**, 297  
 class module, 301  
**Click**, 221, 223, 233, 236, 238, 241

Close	305	DEGREES	111, 113
CODE	98	Delete	307
Color	309, 310	Dim	326, 331
ColorIndex	309, 310	Do-Loop	343
COMBIN	114	Double	327
ComboBox	281		
CommandButton	281	Else	339
comment	333	ElseIf	339, 341
CONFIDENCE.NORM	145	Enabled	218
CONFIDENCE.T	145	End Sub	300
Const	329, 331	Eqv	353
ControlTipText	285	Error	223
Copy	307	event	367
CORREL	126	EXACT	98
COS	113	Excel 옵션	16
COUNT	79, 127	Exit	360, 361
Count	305, 307	EXP	115
COUNTA	80, 127	EXPON.DIST	136
COUNTBLANK	128		
COUNTIF	80, 128	F 검정	147
COUNTIFS	81	F-분포	136
COVARIANCE.P	126	F.DIST	136
COVARIANCE.S	126	F.INV	137
Currency	327	F.TEST	147
		F2	31
DATE	90	FACT	118
Date	327	FALSE	75
DAY	89	field	186
DblClick	223	FIXED	99
Deactivate	372, 374	Font	218, 276, 282, 309

For-Next, 347  
For-Each-Next, 346  
ForeColor, 218, 276, 282  
form, 302  
Formula, 315  
FormulaR1C1, 315  
Frame, 281  
FREQUENCY, 175  
frequency, 175  
frequency table, 175  
Friend, 362  
Function, 302, 358  
FV, 154  
  
GAMMA, 129  
GAMMA.DIST, 138  
GAMMA.INV, 138  
GAMMALN, 129  
GCD, 115  
GEOMEAN, 120  
geometric mean, 120  
Get, 360  
GetFocus, 223  
  
HARMEAN, 121  
harmonic mean, 121  
Height, 218, 276, 282  
histogram, 177  
HLOOKUP, 151  
HorizontalAlignment, 313  
  
HOUR, 90  
hyperlink, 50  
identifier, 331  
IF, 77, 269  
If, 339  
If-Then-Else, 339, 341  
Image, 282  
IMEMode, 285  
Imp, 353  
INDEX, 152, 268  
InputBox, 321  
INT, 116  
Integer, 327  
inter-quartile-range, 179  
Interquartile Range, 180  
IQR, 179, 180  
Is, 352  
ISBLANK, 96  
ISERR, 96  
ISERROR, 96  
ISEVEN, 97  
ISLOGICAL, 97  
ISNA, 97  
ISNONTEXT, 30, 97  
ISNUMBER, 30, 97  
ISODD, 97  
ISREF, 97  
ISTEXT, 97  
Italic, 309

Item, 305, 307  
Keydown, 223  
KeyPress, 224  
KeyUp, 224  
Label, 281  
LCM, 115  
LEFT, 100  
Left, 218  
LEN, 100  
Let, 360  
Like, 352  
LineStyle, 314  
LinkedCell, 246, 249  
List, 227, 235  
ListBox, 281  
ListIndex, 227, 235  
LN, 116  
Locked, 218  
LOG, 116  
LOG10, 116  
LOGNORM.DIST, 139  
LOGNORM.INV, 140  
Long, 327  
LostFocus, 224  
LOWER, 100  
macro, 3, 207  
margin of error, 145  
MATCH, 152  
MATCHMATCH, 253  
MAX, 124  
MAXA, 124  
MDTERM, 108  
Me, 372  
MEDIAN, 121  
MID, 101  
MIN, 124  
MINA, 125  
MINUTE, 91  
MINVERSE, 109  
MMULT, 107  
Mod, 287, 351  
MODE, 121  
module, 301  
MONTH, 89  
MouseDown, 224  
MouseIcon, 218  
MouseMove, 224  
MousePointer, 218  
MouseUp, 224  
Move, 307  
MsgBox, 300, 317, 318  
MultiPage, 281  
Name, 307, 309  
NEGBINOM.DIST, 131  
NETWORKDAYS, 95  
New, 330  
NORM.DIST, 140

NORM.INV, 141  
NORM.S.DIST, 141  
NORM.S.INV, 141  
NOT, 76  
Not, 353  
NOW, 88  
Object, 327  
object, 302  
Office 단추, 4  
OFFSET, 252  
Offset, 307, 316  
Open, 305, 368  
Option Base, 335  
Option Explicit, 330  
OptionButton, 281  
OR, 76  
Or, 354  
outlier, 180  
pareto chart, 178  
PasswordChar, 286  
PI, 111  
Picture, 218, 282  
PicturePosition, 218  
PivotTableUpdate, 374  
Placement, 218  
PMT, 155  
POISSON.DIST, 132  
Preserve, 338  
PrintObject, 218  
Private, 331, 361  
procedure, 355  
project, 301  
PROPER, 101  
Property, 302, 359  
Public, 331, 362  
Q1, 179  
Q2, 179  
Q3, 179  
QUARTILE, 121  
quartile, 179  
R1C1, 30, 61  
RADIAN, 111, 112  
RAND, 315  
Range, 307  
RANK.AVG, 149  
RANK.EQ, 149  
record, 186  
ReDim, 331, 338  
reference, 366  
relative frequency, 175  
Rem, 333  
REPLACE, 102  
REPT, 103  
return, 302  
RGB, 310  
RIGHT, 103

Rnd, 290	String, 327
ROUND, 117	Sub, 300, 302, 355
ROUNDDOWN, 117	SUBTOTAL, 194
ROUNDUP, 117	SUM, 82
Save, 305	SUMIF, 80, 82
SaveAs, 305	SUMIFS, 83
scroll bar, 6	SUMPRODUCT, 119
ScrollBar, 281	SUMSQ, 119
ScrollBars, 276	SUMX2MY2, 119
SECOND, 92	SUMX2PY2, 119
Select, 307	SUMXMY2, 119
Select-Case, 341	Switch, 295
SelectionChange, 371, 374	t - 분포 신뢰구간, 145
Set, 330, 361	t 검정, 147
Shadow, 218	t-분포, 142
sheet, 6, 12	T.DIST, 142
Sheets, 306	T.INV, 142
Show, 274	T.TEST, 147
SIN, 112	TabIndex, 286
Single, 327	TabStrip, 281
Size, 309	TakeFocusOnClick, 219
SpinButton, 282	TAN, 114
SQRT, 118	Text, 286
statement, 331	TextAlign, 286
Static, 331, 362	TextBox, 281
STDEV.P, 123	Then, 339
STDEV.S, 72, 123	TIME, 92
STDEVA, 124	To, 335
STDEVPA, 123	TODAY, 88

ToggleButton, 281  
 Top, 219  
 TRANSPOSE, 105  
 TRIM, 104  
 TRIMMEAN, 121  
 trimmed mean, 121  
 TRUE, 75  
 Until, 343  
 UPPER, 104  
 UserForm, 273  
 VALUE, 104  
 Value, 240, 246, 249, 286, 315  
 value, 365  
 VAR.P, 123  
 VAR.S, 123  
 Variant, 327  
 VBA, 207  
 vbBlack, 295  
 vbBlue, 295  
 vbCyan, 295  
 vbGreen, 295  
 vbMagenta, 295  
 vbRed, 295  
 vbYellow, 295  
 VerticalAlignment, 313  
 Visible, 219, 291  
 Visual Basic, 207  
 Visual Basic Editor, 299  
 Visual Basic 문 작성, 331  
 Visual Basic 식별자 규칙, 331  
 VLOOKUP, 151, 256, 269  
 WEEKDAY, 93  
 WEIBULL.DIST, 143  
 Weight, 314  
 While, 343  
 Width, 219, 276, 283  
 With, 310, 349  
 WordWrap, 219  
 Workbook, 9  
 Workbooks, 303, 305, 368  
 WORKDAY, 94  
 worksheet, 3  
 Worksheets, 306, 307, 370  
 xlAnd, 258  
 xlBottom, 314  
 xlContinuous, 314  
 xlDash, 314  
 xlDashDot, 314  
 xlDashDotDot, 314  
 xlDiagonalDown, 314  
 xlDiagonalUp, 314  
 xlDot, 314  
 xlDouble, 314  
 xlHairline, 314  
 xlHAlignCenter, 313  
 xlHAlignDistributed, 313

x1HAlignJustify, 313	감마함수, 129
x1HAlignLeft, 313	개체, 302
x1HAlignRight, 313	Borders, 314
x1InsideHorizontal, 314	Cells, 307, 313
x1InsideVertical, 314	Color, 310
x1Left, 314	Font, 309
x1LineStyleNone, 314	HorizontalAlignment, 313
x1Medium, 314	LineStyle, 314
x1Right, 314	Me, 372
.xls, 9	Offset, 307, 316
.xlsb, 9	Range, 307
x1SlantDashDot, 314	Sheets, 306
.xlsm, 9, 208	VerticalAlignment, 313
.xlsx, 9	Weight, 314
x1Thick, 314	Workbooks, 303, 305, 368
x1Thin, 314	Worksheets, 306, 307, 370
x1Top, 314	곡선 그래프, 172
x1VAlignBottom, 313	공분산, 126
x1VAlignCenter, 313	구문 오류 검사, 333
x1VAlignDistributed, 313	글꼴, 34, 44
x1VAlignJustify, 313	기하평균, 120
x1VAlignTop, 313	논리 연산자, 353
.xml, 10	논리 함수, 74
Xor, 354	다중 페이지, 281
YEAR, 88	데이터 종류
Z 검정, 148	Boolean, 327
Z.TEST, 148	Byte, 327
감마분포, 138	Currency, 327
	Date, 327

- D**ouble, 327  
**I**nteger, 327  
**L**ong, 327  
**O**bject, 327  
**S**ingle, 327  
**S**tring, 327  
**V**ariant, 327  
 데이터베이스, 186  
 도구상자, 276  
 도수, 175  
 도수분포표, 175  
 동적 배열, 337  
 레이블, 281  
 레코드, 186  
 로그 함수, 116  
 로그정규분포, 139  
 리본메뉴, 5  
 막대 그래프, 164  
 맞춤, 34, 43  
 매크로, 3, 207  
 매크로 기록, 208  
 매크로 실행, 212  
 메모, 49  
     메모 삭제, 50  
     메모 삽입, 49  
     메모 편집, 50  
 메모 삭제, 50  
 메모 삽입, 49  
 메모 편집, 50  
 명령 단추, 217, 261, 281  
 모듈, 301  
 목록 상자, 234, 264, 281  
 목표값 찾기, 197  
 무늬, 34  
 문자열 연결 연산자, 352  
 바로 가기 키  
     ALT + ENTER, 44  
     ALT + F4, 4  
     CTRL + ↓, 32  
     CTRL + ←, 32  
     CTRL + →, 32  
     CTRL + ↑, 32  
     CTRL + :, 31, 88  
     CTRL + C, 58  
     CTRL + SHIFT + :, 88  
     CTRL + SHIFT + ENTER, 105  
     CTRL + V, 58  
     CTRL + X, 58  
 배열, 334  
     다차원 배열, 336  
     동적 배열, 337  
     하한값 변경, 335  
 베타분포, 133  
 보호, 34, 45  
 부분합, 194  
 비교 연산자, 52, 351  
 비주얼 베이직, 207  
 빠른실행도구모음, 5

- 사분위수, 179  
사분위수 범위, 180  
사용자 정의 폼, 273  
산술 연산자, 52, 351  
산술평균, 120  
산점도, 172  
삼각 함수, 111  
상관계수, 126  
상대 참조, 60  
상대도수, 175  
상자 그림, 179  
상자 수염 그림, 179  
상태표시줄, 6  
새로 만들기, 11  
셀, 6, 29  
    번지, 29  
    서식, 33  
    수식 입력, 51  
셀 번지, 29  
셀 번지  
    A1, 29, 60  
    R1C1, 30, 61  
서식  
    테두리, 34  
셀 서식, 33  
    글꼴, 34, 44  
    맞춤, 34, 43  
    무늬, 34  
    보호, 34, 45  
    채우기, 45  
    테두리, 45  
    표시형식, 34  
    셀 선택, 32  
    셀 이동, 32  
    셀 입력, 30  
    셀 참조, 59  
        상대 참조, 60  
        시트 참조, 61  
        외부 참조, 59  
        원격 참조, 59  
        절대 참조, 60  
        통합문서 참조, 62  
        혼합 참조, 60  
    셀 채우기, 54  
    셀 편집, 30  
    수식입력줄, 7  
    수학 함수, 111  
    순열, 114  
    스크롤 막대, 246, 281  
    스크롤바, 6  
    스핀 단추, 248, 282  
    시나리오, 202  
    시트, 12  
    시트, 6  
    시트 보호, 45  
    시트 참조, 61  
    식별자, 331  
    양식 컨트롤, 217, 261, 373

- 엑셀 파일 종류, 9  
 연산, 51  
 연산자, 350
  - ., 304
  - , 351
  - <=, 352
  - <, 351
  - <>, 352
  - >=, 352
  - >, 352
  - \*, 351
  - +, 351, 353
  - /, 351
  - =, 352
  - And, 353
  - Eqv, 353
  - Imp, 353
  - Is, 352
  - Mod, 351
  - Not, 353
  - Or, 354
  - Xor, 354
  - &, 352
  - ^, 351
  - \, 351
  - Like, 352
  - 논리 연산자, 353
  - 문자열 연결 연산자, 352
  - 비교 연산자, 52, 351
 산술 연산자, 52, 351  
 참조 연산자, 53  
 텍스트 연산자, 53
  - 연속 데이터, 54
  - 연속탭, 281
  - 오차한계, 145
  - 옵션 단추, 238, 265, 281
  - 와이블분포, 143
  - 외부 참조, 59
  - 워크북, 9
  - 워크시트, 3
  - 원 그래프, 163
  - 원격 참조, 59
  - 유저폼, 273
  - 유효성 검사, 192
  - 음이항분포, 131
  - 이름관리자, 33
  - 이름상자, 7, 33
  - 이미지, 282
  - 이벤트, 367
  - 이상점, 180
  - 이항분포, 130
  - 인쇄, 15
  - 인수 전달
    - ByRef, 366
    - ByVal, 365
  - 절대 참조, 60
  - 절삭평균, 121
  - 정규분포, 140

- 정규분포 신뢰구간, 145  
정렬, 186  
제 1 사분의수, 179  
제 2 사분의수, 179  
제 3 사분의수, 179  
제목표시줄, 4  
조건부 서식, 46  
조화평균, 121  
주석, 333  
중앙값, 120, 179, 180  
중첩 함수, 78  
지수분포, 136  
차트, 162  
참조 연산자, 53  
채우기, 45  
채우기 핸들, 54  
최대공약수, 115  
최댓값, 124, 125, 179  
최빈값, 120  
최소공배수, 115  
최솟값, 124, 179  
추가기능, 200  
카이제곱분포, 135  
컨트롤, 217, 276  
    CheckBox, 229, 263, 281  
    ComboBox, 224, 262, 281  
    CommandButton, 217, 261, 281  
    Frame, 281  
    Image, 282  
Label, 281  
ListBox, 234, 264, 281  
MultiPage, 281  
OptionButton, 238, 265, 281  
ScrollBar, 246, 281  
SpinButton, 248, 282  
TabStrip, 281  
TextBox, 243, 281  
ToggleButton, 281  
ActiveX 컨트롤, 217, 373  
레이블, 281  
명령 단추, 281  
목록 상자, 281  
스크롤 막대, 281  
스핀 단추, 282  
양식 컨트롤, 217, 373  
연속탭, 281  
옵션 단추, 281  
이미지, 282  
콤보 상자, 281  
텍스트 상자, 281  
토글 단추, 281  
프레임, 281  
확인란, 281  
컨트롤 도구상자, 217, 276  
콤보 상자, 224, 262, 281  
클래스 모듈, 301  
테두리, 34, 45  
텍스트 나누기, 67

- 텍스트 마법사, 68  
 텍스트 상자, 243, 281  
 텍스트 연산자, 53  
 토글 단추, 281  
 통계 함수, 120  
 통합문서, 9, 303  
 통합문서 참조, 62  
 파레토 그림, 178  
 파일 암호 입력, 12  
 파일 열기, 10  
 파일 저장, 12  
 편집창, 5  
 평균, 120, 123  
 포아송분포, 132  
 폼, 302  
 표시형식, 34  
 표준정규분포, 141  
 프레임, 281  
 프로시저, 355  
     Get, 360  
     Let, 360  
     Set, 361  
 프로젝트, 301  
 필드, 186  
 필터, 188  
 하이퍼링크, 50  
 하한값 변경, 335  
 함수, 67  
     Choose, 297  
     InputBox, 321  
     MATCH, 253  
     MsgBox, 300, 317  
     OFFSET, 252  
     RGB, 310  
     Rnd, 290  
     Switch, 295  
     INDEX, 268  
     VLOOKUP, 256  
     감마함수, 129  
     논리 함수, 74  
     삼각 함수, 111  
     수학 함수, 111  
     행렬, 105  
     함수 마법사, 71, 73  
     합계 함수, 119  
     해 찾기, 200  
     행렬, 105  
     혼합 차트, 181  
     혼합 참조, 60  
     화면표시단추, 6  
     확대/축소 슬라이더, 7  
     확인란, 229, 263, 281  
     히스토그램, 177