

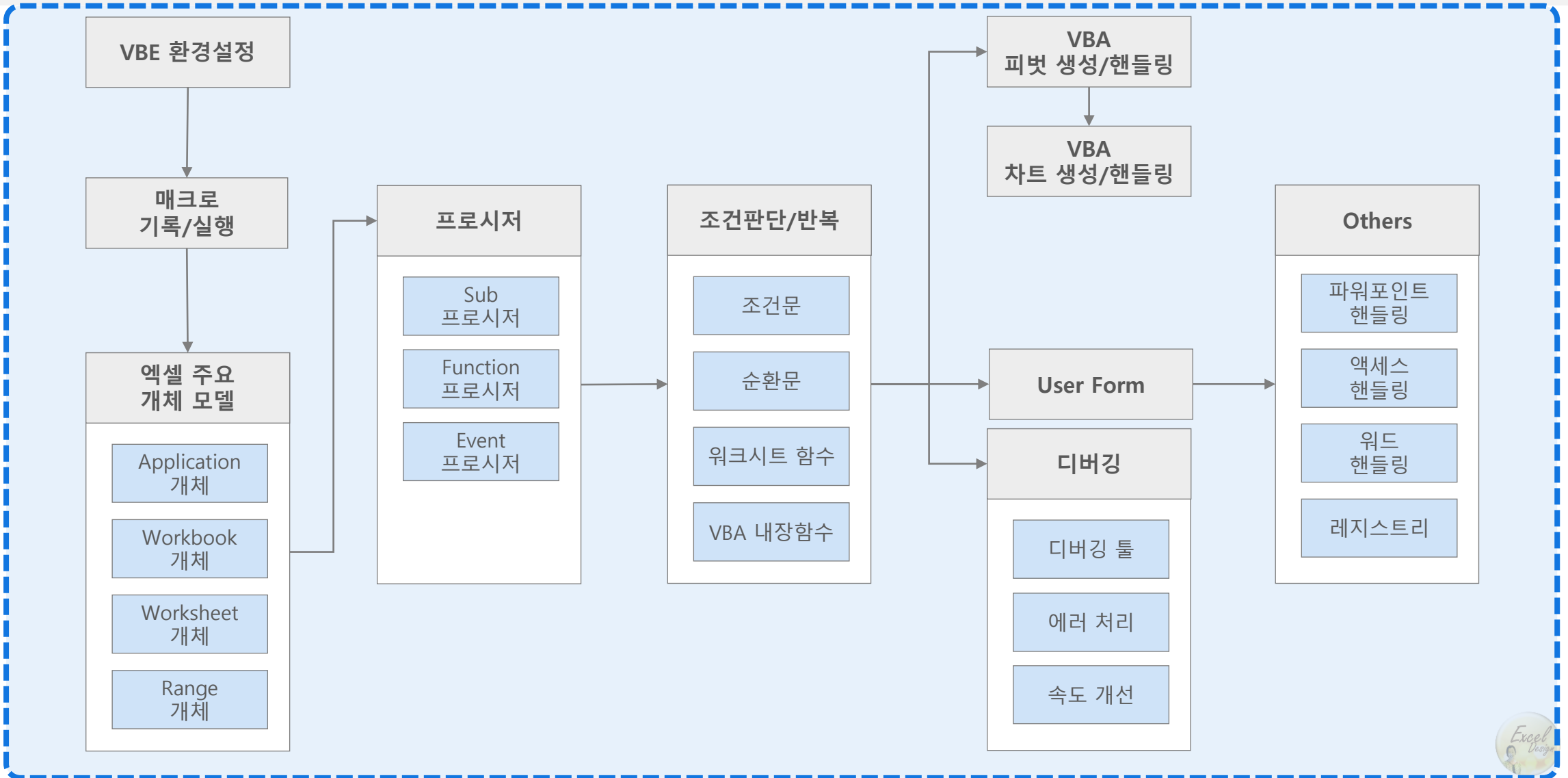
엑셀 VBA를 활용한 자동화 기초



I. VBA 필수 문법



1. VBA Skill Map



2. 용어 정리

1) 매크로(macro)

- 반복되는 작업을 기록해 두었다가 필요시 단축키 또는 버튼만 누르면 해당 작업을 자동으로 실행할 수 있는 기능
- 매크로 기능이 들어있는 파일을 ‘매크로 파일’이라고 부름

2) VBA(Visual Basic for Application)

- MS사의 윈도 오피스 응용 프로그램용 프로그래밍 언어

3) VBE(Visual Basic Editor)

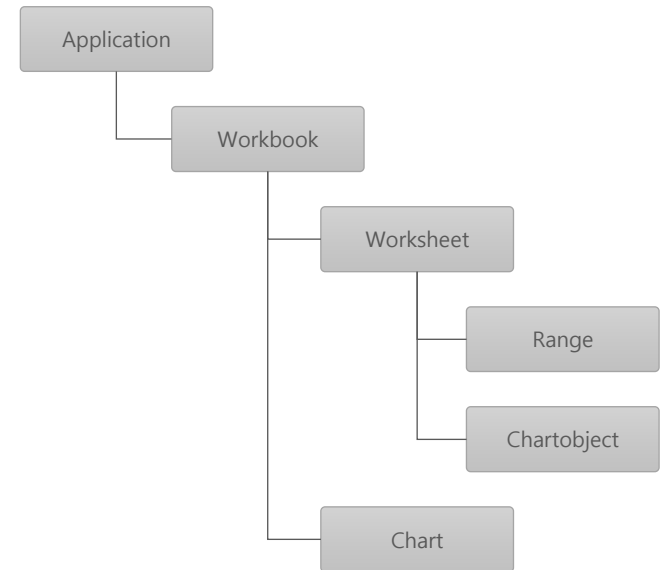
- VBA로 코딩하기 위한 편집창



4. 개체 모델 이해 1

예제파일: 개체모델>개체모델.xlsm

- 1) 개체간의 관계를 설명하는 족보
- 2) Workbooks 컬렉션 개체 핸들링
 - Application.Workbooks("1.xlsm").Activate
 - Workbooks("2.xlsm").Activate
- 3) Thisworkbook과 Activeworkbook 차이
 - Thisworkbook : 현재 VBA 코딩이 이루어지고 있는 파일
 - Activeworkbook : 현재 활성화된 파일로, 키보드로 타이핑을 하면 입력되는 파일
- 4) 워크시트를 표현하는 다양한 방법
 - Worksheets("취합")
 - Worksheets(1)
 - Sheets("취합")
 - Sheet1 : 시트의 이름이 바뀌어도 영향을 받지 않는 표현법



5. 개체 모델 이해2



예제파일: 개체모델>Range 개체.xlsm

1) 다양한 Range 접근 방법

- 기본
Sheet1.Range("a1") = "그랜저"
Sheet1.Cells(1, 1) = "쏘나타"
Sheet1.Cells(2, "a") = "i40"
- 마지막: **마지막 셀의 행 번호, 열 번호 체크**
Sheet2.Range("a1").End(xlDown).Select
Sheet2.Range("a1").End(xlToRight).Select
Sheet2.Range("d10").End(xlUp).Select
Sheet2.Range("d10").End(xlToLeft).Select
- 이동: **데이터를 아래로 아래로 누적시킬 때**
Sheet2.Range("b5").Offset(1, 1).Select
Sheet2.Range("b5").Offset(-2, 1).Select
Sheet2.Range("b5").Offset(3).Select
- 응용: **필드에 빈 셀이 존재할 때 마지막 셀 체크**
Sheet2.Range("c1").End(xlDown).Select
Sheet2.Cells(Rows.Count, "c").Select
Sheet2.Cells(Rows.Count, "c").End(xlUp).Select
- 범위: **연속된 범위 설정**
Sheet2.Range("a1").CurrentRegion.Select
Sheet2.Range("a1", Sheet2.Range("a1").End(xlDown)).Select
Sheet2.Range("c1", Sheet2.Cells(Rows.Count, "c").End(3)).Select

2) 메서드(Method) : 실행 명령

- **개체.메서드**
Sheet1.Range("a8:a9").Delete
Sheet1.Range("a5:a7").ClearContents
Sheet1.Range("a1:a4").Clear

3) 속성 : 개체의 속성 표현

- **개체.속성=속성값**
Sheet2.Range("a1").Interior.ColorIndex=23
Sheet2.Range("a2").Interior.Color = rgb(123,25,11)
Sheet2.Range("a3").Interior.Color=vbyellow

1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	32	33	34	35
36	37	38	39	40	41	42
43	44	45	46	47	48	49
50	51	52	53	54	55	56

6. With문 / 변수

예제파일: 개체모델>with_변수.xlsm

1) With문으로 코드 효율화(처리속도 ↑)

```
With Sheet1
    .Range("a1").CurrentRegion.Select
    .Range("a1", .Range("a1").End(xlDown)).Select
    .Range("c1", .Cells(Rows.Count, "c").End(3)).Select
End With
```

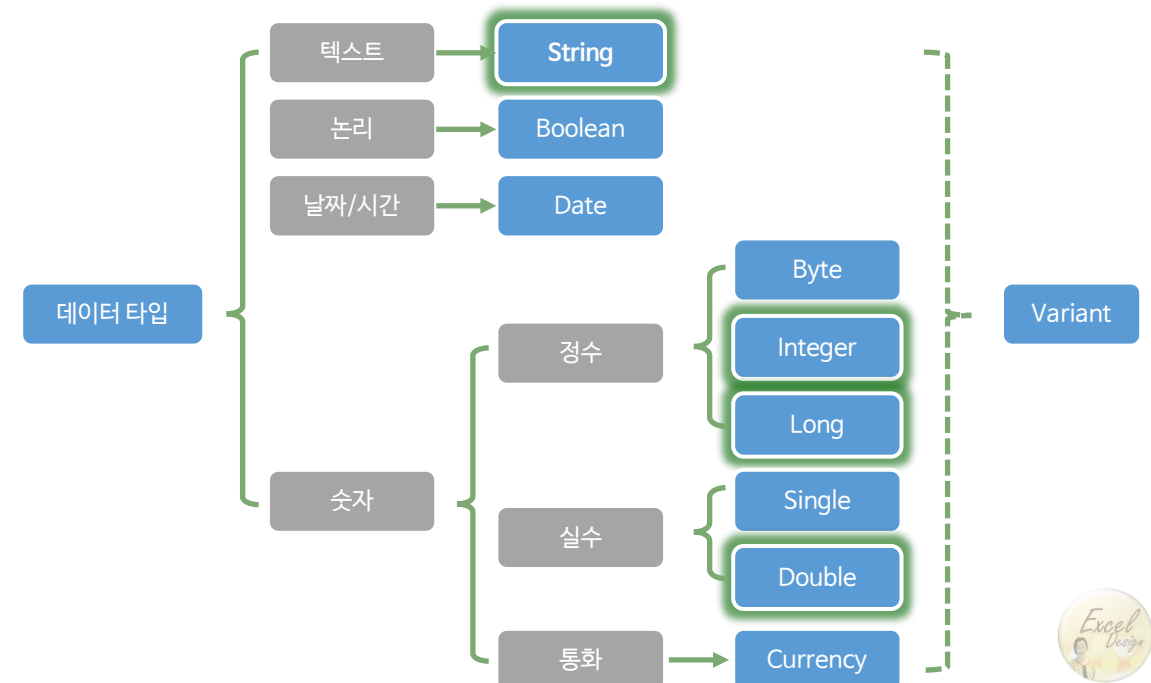
```
With Sheet1.Range("a1")
    .Font.Bold = True
    .Font.Name = "돋움체"
    .Interior.ColorIndex = 32
End With
```

2) 변수 : 값을 저장하는 방

일반변수
개체변수
배열변수

지역변수
전역변수

3) 변수는 Dim문을 사용해 선언: Dim 변수명 As 데이터타입



7. 다양한 디버깅 툴을 활용한 변수 검증

예제파일: 변수 검증.xlsm

- 1) 선언한 변수는 검증을 통해 바로바로 정확한 값이 들어오는지 체크하기
 - 아래의 디버깅 툴 + 필요 시 '중단점' 같이 접목 시키기
- 2) 일반변수에 들어온 값 확인하기(4가지)
 - F8 + 양 손
 - MsgBox
 - Debug.Print + 직접실행창(Ctrl+G)
 - 지역창 or 조사식창
- 3) 개체변수에 들어온 값 확인하기(2가지)
 - 개체변수.Select
 - **지역창** or 조사식창



8. 조건문, 반복문 기본

1) 조건문 종류

- IF문
- Select Case문
- IF문 종류
 - If 조건 Then 실행

```
If 조건 Then  
    실행  
Else  
    실행  
End If
```

```
If      조건      Then  
    실행  
Elseif 조건 Then  
    실행  
Elseif 조건 Then  
    실행  
Else  
    실행  
End If
```

2) 반복문 종류

- For ~ Next문
- For each ~ Next문
- Do ~ Loop문

- For ~ Next문

```
For i = 시작값 To 종료값 Step 증가값
```

```
    실행문
```

```
Next
```

II. 기초 매크로 파일 개발



9. 기초 매크로 파일 개발

예제파일: 평가표.xlsx

1) 평가표

Product Name	Min	Result
BEA123	325	
BEA124	166	
BEA125	487	
BEA126	529	
BEA127	136	
BEA128	502	
BEA129	281	
BEA130	338	
BEA131	171	

Run

2) 구구단

1×1 = 1	2×1 = 2	3×1 = 3	4×1 = 4	5×1 = 5	6×1 = 6	7×1 = 7	8×1 = 8	9×1 = 9
1×2 = 2	2×2 = 4	3×2 = 6	4×2 = 8	5×2 = 10	6×2 = 12	7×2 = 14	8×2 = 16	9×2 = 18
1×3 = 3	2×3 = 6	3×3 = 9	4×3 = 12	5×3 = 15	6×3 = 18	7×3 = 21	8×3 = 24	9×3 = 27
1×4 = 4	2×4 = 8	3×4 = 12	4×4 = 16	5×4 = 20	6×4 = 24	7×4 = 28	8×4 = 32	9×4 = 36
1×5 = 5	2×5 = 10	3×5 = 15	4×5 = 20	5×5 = 25	6×5 = 30	7×5 = 35	8×5 = 40	9×5 = 45
1×6 = 6	2×6 = 12	3×6 = 18	4×6 = 24	5×6 = 30	6×6 = 36	7×6 = 42	8×6 = 48	9×6 = 54
1×7 = 7	2×7 = 14	3×7 = 21	4×7 = 28	5×7 = 35	6×7 = 42	7×7 = 49	8×7 = 56	9×7 = 63
1×8 = 8	2×8 = 16	3×8 = 24	4×8 = 32	5×8 = 40	6×8 = 48	7×8 = 56	8×8 = 64	9×8 = 72
1×9 = 9	2×9 = 18	3×9 = 27	4×9 = 36	5×9 = 45	6×9 = 54	7×9 = 63	8×9 = 72	9×9 = 81

3) VBA 기본 색상표

	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	32	33	34	35
36	37	38	39	40	41	42
43	44	45	46	47	48	49
50	51	52	53	54	55	56

III. 자동 매크로



10. 자동매크로 & 구글링 활용



예제파일: 자동매크로_구글링 활용.xlsx

- 1) 텍스트나누기를 VBA로 어떻게 코딩하는지 모른다
- 2) 개발도구 > 매크로 기록...클릭 : 워크시트에서 작업하는 모든 것이 vba로 자동 코딩됨
- 3) 필요한 작업, '텍스트나누기'가 끝나면, 개발도구 > 기록 중지...클릭
- 4) VBE > 새로운 모듈 or 프로시저가 생성되고 > 여기에 자동 기록된 코드 보임
- 5) 자동매크로 코드를 그대로 사용하는 것은 절대 바람직하지 못하다
 - 자동 매크로가 기록 못해주는 작업도 많다
 - 불필요한 코드가 많아서 처리속도가 현저히 떨어진다
- 6) 그러면 어떻게 해야 하는가?
- 7) 자동매크로에서 중요 키워드 추출
- 8) 구글 검색창에 'vba 중요키워드(영어로)'로 검색하면 영어 커뮤니티에서 잘 정제된 코드들을 얻을 수 있다
- 9) 정제된 코드 복사 > 내 프로시저에 적용 및 수정



IV. 통계분석 자동화



11. 회귀분석 자동화

- 1) 리본메뉴 > 개발도구 > Excel 추가기능 > '분석도구', '분석도구-VBA' 체크 > 확인
- 2) 리본메뉴 > 개발도구 > 매크로 기록 > 확인
- 3) 리본메뉴 > 데이터 > 데이터 분석 > 회귀분석 > 확인
- 4) 회귀분석 대화상자에서 필요 항목 입력 > 확인
- 5) 리본메뉴 > 개발도구 > 기록중지
- 6) VBE > 자동매크로...로 기록된 코드 수정
- 7) 매크로 버튼에 수정된 회귀분석 프로시저 연결

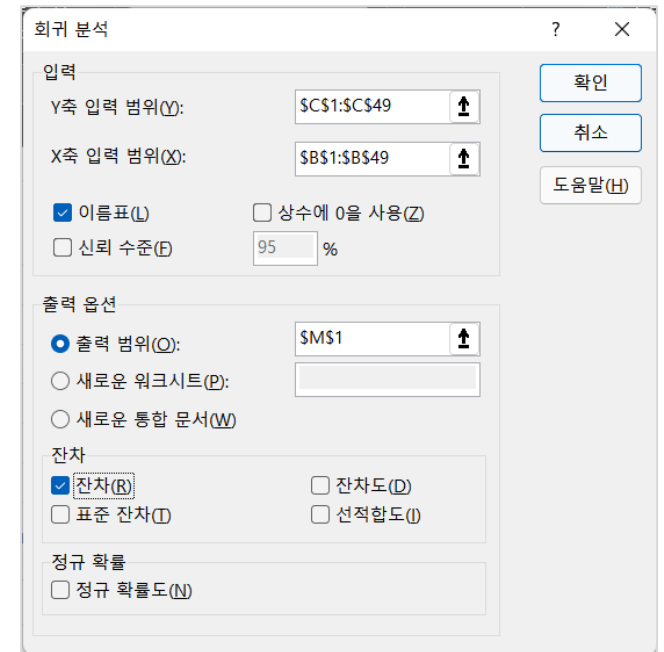
```
Sub RunRegression()  
    Dim rngX As Range, rngY As Range  
  
    With Sheet1  
        Set rngX = .Range("b1", .Cells(Rows.Count, "b").End(3))  
        Set rngY = .Range("c1", .Cells(Rows.Count, "c").End(3))  
  
        Application.Run "ATPVBAEN.XLAM!Regress", rngY,  
            rngX, False, True, , Range("e22"),  
            , True, False, False, False, , False  
  
    End With  
End Sub
```

종속변수 영역

독립변수 영역

회귀분석 결과가 뿌려질 빈 셀 주소

회귀분석 대화상자에서 여러 옵션들의 선택 여부



The image shows the '회귀 분석' (Regression Analysis) dialog box in Excel. It is divided into two main sections: '입력' (Input) and '출력 옵션' (Output Options). In the '입력' section, the 'Y축 입력 범위(Y):' is set to '\$C\$1:\$C\$49' and the 'X축 입력 범위(X):' is set to '\$B\$1:\$B\$49'. The '이름표(L)' checkbox is checked, and the '신뢰 수준(E)' is set to 95%. In the '출력 옵션' section, the '출력 범위(O):' is set to '\$M\$1'. The '잔차(R)' checkbox is checked, and the '잔차도(D)' and '선택합도(I)' checkboxes are unchecked. The '정규 확률' section has the '정규 확률도(N)' checkbox unchecked. On the right side of the dialog, there are three buttons: '확인' (OK), '취소' (Cancel), and '도움말(H)' (Help).



V. 실무 매크로 파일 개발



12. 조건 만족하는 행 삭제 매크로

예제파일: 행삭제.xlsx

- 1) A열에서 셀 값이 'Bentley'로 시작하는 셀은 삭제하기
- 2) 기존의 For ~ Next 구문으로 코딩하게 되면
 - 'Bentley'가 연속되어 있는 셀은 다 지우질 못하는 결과 발생
 - 위 →아래 방향으로 순차적으로 지우면서 행번호가 바뀌기 때문이다
 - What should we do?
 - 아래 →위 방향으로 역순으로 지우면 해결
 - For ~Next 구문의 Step 속성 활용: For i = 1R to 2 Step -1
- 3) 'Bentley'와 같으면
 - If cells(i, "a") = "Bentley" then
- 4) 'Bentley'로 시작하면
 - If cells(i, "a") Like "Bentley*" then



13. 숫자 추출 매크로

예제파일: 숫자추출.xlsx

- If 셀 값 Like “[ABC]*” then : 셀 값이 A, B, C 중 하나로 시작
- If 셀 값 Like “[A-C]*” then : 셀 값이 A, B, C 중 하나로 시작
- If 셀 값 Like “[가-힣]” then : 셀 값이 한글이면
- If 셀 값 Like “[A-Z]” then : 셀 값이 영어이면
- If 셀 값 Like “[0-9]” then : 셀 값이 숫자이면
- If 셀 값 Like “[!가-힣]” then : 셀 값이 한글이 아니면
- If 셀 값 Like “[!ABC]*” then : 셀 값이 A, B, C로 시작하지 않으면
- VBA 코딩 시, 워크시트 함수 목록에 없는 함수들은 사용 못하는가?
 - 그럴 때는
 - 코딩 창에 ‘vba.함수 첫글자’를 입력하면 사용할 수 있는 함수 목록이 뜬다
 - 필요한 함수를 선택하여 적용
 - 맨 앞의 ‘vba’는 생략 가능



VI. 패턴코드 활용 매크로 개발

시각적 분석



14. 시각적 분석 1_데이터 가져오기

예제파일: Superstore KR v2019.xlsx
예제파일: 패턴코드_특정 파일 가져오기.txt

```
Sub GetData()  
'데이터 가져오기  
Dim FN As String  
Dim wb As Workbook  
Dim ws As Worksheet  
  
Application.ScreenUpdating = False  
  
FN = Environ("userprofile") & "\OneDrive\바탕 화면\Superstore KR v2019.xlsx"  
  
If IsFileExist(FN) = False Then  
    MsgBox "파일이 존재하지 않습니다"  
    Exit Sub  
End If  
  
If IsFileOpen(FN) = True Then  
    MsgBox "파일이 이미 열려 있습니다. 닫고 다시 시작하세요"  
    End  
End If  
  
Set wb = Workbooks.Open(Filename:=FN, UpdateLinks:=0)  
Set ws = wb.Sheets(1)  
  
'to do  
Sheet1.Cells.Clear  
If Sheet1.Name <> "주문" Then Sheet1.Name = "주문"  
  
ws.Range("a1").CurrentRegion.Copy Sheet1.Range("a1")  
  
wb.Close False  
  
Application.ScreenUpdating = True  
  
Set wb = Nothing  
Set ws = Nothing  
  
End Sub
```

```
Function IsFileExist(FN As String) As Boolean  
    IsFileExist = (Dir(FN) <> "")  
  
End Function
```

```
Function IsFileOpen(FN As String) As Boolean  
    Dim OpenFName As Workbook  
  
    On Error Resume Next  
    Set OpenFName = Workbooks(Dir(FN))  
    IsFileOpen = (Err.Number = 0)  
  
End Function
```

-----> 원본 데이터가 있는 경로의 데이터 가져오기

-----> 해당 파일 존재여부 파악 프로시저

-----> 해당 파일 오픈 여부 파악 프로시저



15. 시각적 분석2_피벗테이블+피벗차트

예제파일 : 패턴코드_피벗테이블_피벗차트.txt

※ 피벗테이블

```
Application.ScreenUpdating = False
Application.DisplayAlerts = False

'기존 Dashboard 시트 삭제
On Error Resume Next
    Sheets("Dashboard").Delete
On Error GoTo 0

Sheets.Add.Name = "Dashboard"
Range("a3").Select

Set pc = ThisWorkbook.PivotCaches.Create(xlDatabase,
Sheet1.Range("a1").CurrentRegion)
Set pt = pc.CreatePivotTable(ActiveCell, "SalesPivot")

'pivot Table
With pt
    .AddFields "주문 일자"
    .AddDataField .PivotFields("매출")
    .RowAxisLayout xlTabularRow

    Set pf = .PivotFields("주문 일자")
    pf.LabelRange.Group Start:=True, End:=True,
    Periods:=Array(False, False, False, _
    False, True, False, False)

    .DataFields(1).NumberFormat = "#,#"
End With
```

※ 피벗차트

```
Set ws = Sheets("Dashboard")

'기존 차트 삭제
On Error Resume Next
    ws.ChartObjects.Delete
On Error GoTo 0

Set sh = ws.Shapes.AddChart2(, xlChartType.xlLine)
Set ch = sh.Chart

'차트 요소
With ch
    .SetSourceData pt.TableRange1
    .ChartTitle.Text = "월별 매출추이"

    .ShowAllFieldButtons = False
    .SetElement (msoElementLegendNone)
    .SetElement (msoElementPrimaryValueGridLinesNone)
    .SetElement (msoElementPrimaryValueAxisNone)
    .FullSeriesCollection(1).ApplyDataLabels
End With

'차트 위치, 사이즈
With sh
    .Top = pt.TableRange1.Top
    .Left = pt.TableRange1.Left + pt.TableRange1.Width + 10
    .Height = pt.TableRange1.Height
    .Width = 600
End With

ActiveWindow.DisplayGridlines = False
```



16. 시각적 분석3_프로시저 연결

- 1) Call문으로 여러 프로시저를 연결하는 프로시저 생성
- 2) 워크시트에 매크로 버튼 생성
- 3) 매크로 버튼에 call문이 입력된 프로시저를 연결하여 완성

```
Sub main()
```

```
Call GetData  
Call Visualization
```

```
End Sub
```

