

示」タブの「マクロ」の「▼」から「相対参照で記録」を選ぶ(図15)。なお、この操作は、記録を開始する前に実行してもよい。以下、マクロ「書式設定1」の記録時と全く同じ手順で操作していき、マクロの記録を終了する。

マクロ「書式設定1」と、新たに作成したマクロ「書式設定3」がどのように違うかを、実際に実行して確認しよう。「実行用2」シートを開き、E2セルを選択して、マクロ「書式設定1」を実行する(図16上)。この場合、最初に選択していたセルとは関係なく、B2～C3セルに書式が設定される(図16下)。

一方、同じ内容の「実行用3」シートを開き、やはりE2セルを選択して、マクロ「書式設定3」を実行する(図17上)。今回は、B2～C3セルではなく、選択したE2セルを基準とする2行×2列のセル範囲、つまりE2～F3セルに対して、同様の書式が設定される(図17下)。

相対参照のマクロコードを確認

このマクロ「書式設定3」がどのようなSubプロシージャになっているかを、VBEで確認してみよう(図18)。書式設定の処理自体はSelectionに対して実行されており、「書式設定1」と変わらない。異なるのは、最初に対象のセル(範囲)を選択している部分だ。

「Range("B2:C2")」などの「Range」もプロパティだが、通常は対象オブジェクトを指定せず、引数に文字列で指定されたセル参照が、そのままRangeオブジェクトとして取得される。一方、「Range」の対象に別のRangeオブジェクトを指定



図16 記録機能で作成した2つのマクロの動作を比較してみよう。まず、「実行用2」シートを開き、E2セルを選択して、マクロ「書式設定1」を実行する。この場合、最初に選択していたセルとは関係なく、B2～C3セルに書式が設定される



図17 同じ内容の「実行用3」シートを開き、やはりE2セルを選択して、マクロ「書式設定3」を実行する。相対参照で記録しているため、最初に選択したE2セルを左上端とする2行×2列のセル範囲、すなわちE2～F3セルに書式が設定される

●マクロ「書式設定3」を確認する

```
Sub 書式設定3()
    ' 書式設定3 Macro
    '
    ActiveCell.Range("A1:B1").Select
    Selection.Font.Bold = True
    With Selection
        .HorizontalAlignment = xlCenter
        .VerticalAlignment = xlCenter
        .WrapText = False
        .Orientation = 0
        .AddIndent = False
        .IndentLevel = 0
        .ShrinkToFit = False
        .ReadingOrder = xlContext
        .MergeCells = False
    End With
    ActiveCell.Offset(1, 0).Range("A1:B1").Select
    Selection.NumberFormatLocal = "##,###0";[赤]("##,##0")
End Sub
```

図18 「マクロ」画面でマクロ「書式設定3」を選択して「編集」をクリックし、VBEでそのSubプロシージャを確認する。そのコードでは、処理自体は「書式設定1」と同様だが、最初に対象のセルを選択する部分の記述が異なっている