How can I apply conditional formatting to cells in VBA?

Authored by

stats writer

June 24, 2024

RECOMMENDED CITATION

stats writer (2024). *How can I apply conditional formatting to cells in VBA?*. PSYCHOLOGICAL SCALES. Retrieved from https://scales.arabpsychology.com/?p=150807

ARABPSYCHOLOGY.COM

Conditional formatting is a feature in Microsoft Excel that allows users to automatically format cells based on specified criteria. This can be applied to cells in VBA (Visual Basic for Applications) by using the "FormatConditions" method. By setting up conditions and corresponding formatting rules, users can create dynamic and visually appealing spreadsheets in VBA. This can be achieved by using various VBA functions and properties such as "If-Then" statements and "ColorIndex". By utilizing conditional formatting in VBA, users can efficiently manage and analyze large sets of data with ease.

VBA: Apply Conditional Formatting to Cells

You can use the following methods in VBA to apply conditional formatting to cells:

Method 1: Apply Conditional Formatting Based on One Condition

```
Sub ConditionalFormatOne()
Dim rg As Range
Dim cond As FormatCondition

'specify range to apply conditional formatting
Set rg = Range("B2:B11")

'clear any existing conditional formatting
rg.FormatConditions.Delete

'apply conditional formatting to any cell in range B2:B11 with value greater than 30
```

ARABPSYCHOLOGY.COM

```
Set cond = rg.FormatConditions.Add(xlCellValue, xlGreater, "=30")

'define conditional formatting to use
With cond
.Interior.Color = vbGreen
.Font.Color = vbBlack
.Font.Bold = True
End With

End Sub
```

Method 2: Apply Conditional Formatting Based on Multiple Conditions

```
Sub ConditionalFormatMultiple()
Dim rg As Range
Dim cond1 As FormatCondition, cond2 As FormatCondition, cond3 As FormatCondition

'specify range to apply conditional formatting
Set rg = Range("A2:A11")

'clear any existing conditional formatting
rg.FormatConditions.Delete
```

ARABPSYCHOLOGY.COM

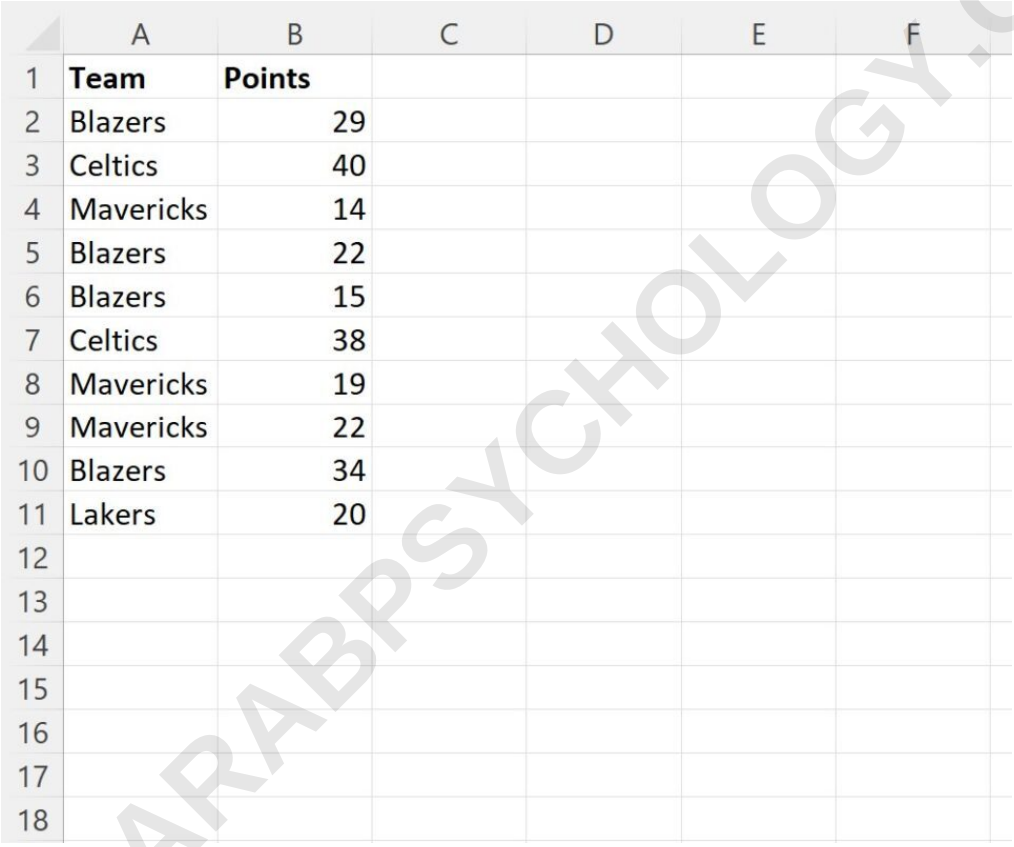
```
'specify rules for conditional formatting
Set cond1 = rg.FormatConditions.Add(xlCellValue, xlEqual, "Mavericks")
Set cond2 = rg.FormatConditions.Add(xlCellValue, xlEqual, "Blazers")
Set cond3 = rg.FormatConditions.Add(xlCellValue, xlEqual, "Celtics")

'define conditional formatting to use
With cond1
.Interior.Color = vbBlue
.Font.Color = vbWhite
.Font.Italic = True
End WithWith cond2
.Interior.Color = vbRed
.Font.Color = vbWhite
.Font.Bold = True
End WithWith cond3
.Interior.Color = vbGreen
.Font.Color = vbBlack
End WithEnd Sub
```

Method 3: Remove All Conditional Formatting Rules from Cells

ARABPSYCHOLOGY.COM

```
Sub RemoveConditionalFormatting()
ActiveSheet.Cells.FormatConditions.Delete
End Sub
```

The following examples shows how to use each method in practice with the following dataset in Excel:



	A	B	C	D	E	F
1	**Team**	**Points**				
2	Blazers	29				
3	Celtics	40				
4	Mavericks	14				
5	Blazers	22				
6	Blazers	15				
7	Celtics	38				
8	Mavericks	19				
9	Mavericks	22				
10	Blazers	34				
11	Lakers	20				
12						
13						
14						
15						
16						
17						
18						

ARABPSYCHOLOGY.COM

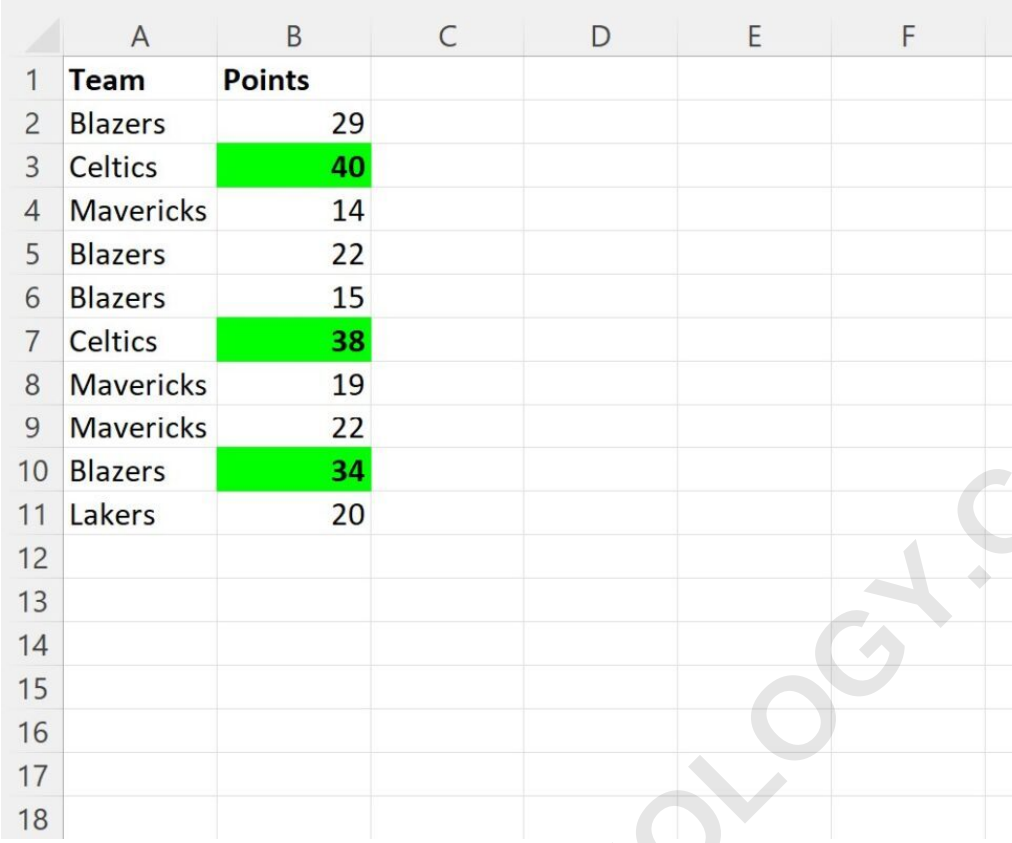
Example 1: Apply Conditional Formatting Based on One Condition

We can use the following macro to fill in cells in the range B2:B11 that have a value greater than 30 with a green background, black font and bold text style:

```
Sub ConditionalFormatOne()
Dim rg As Range
Dim cond As FormatCondition

'specify range to apply conditional formatting
Set rg = Range("B2:B11")

'clear any existing conditional formatting
rg.FormatConditions.Delete

'apply conditional formatting to any cell in range B2:B11
with value greater than 30
Set cond = rg.FormatConditions.Add(xlCellValue,
xlGreater, "=30")

'define conditional formatting to use
With cond
.Interior.Color = vbGreen
.Font.Color = vbBlack
.Font.Bold = True
End WithEnd Sub
```

When we run this macro, we receive the following output:

ARABPSYCHOLOGY.COM



	A	B	C	D	E	F
1	**Team**	**Points**				
2	Blazers	29				
3	Celtics	**40**				
4	Mavericks	14				
5	Blazers	22				
6	Blazers	15				
7	Celtics	**38**				
8	Mavericks	19				
9	Mavericks	22				
10	Blazers	**34**				
11	Lakers	20				
12						
13						
14						
15						
16						
17						
18						

ARABPSYCHOLOGY.COM

Notice that each cell in the range B2:B11 that has a value greater than 30 has conditional formatting applied to it.

Any cell with a value equal to or less than 30 is simply left alone.

Example 2: Apply Conditional Formatting Based on Multiple Conditions

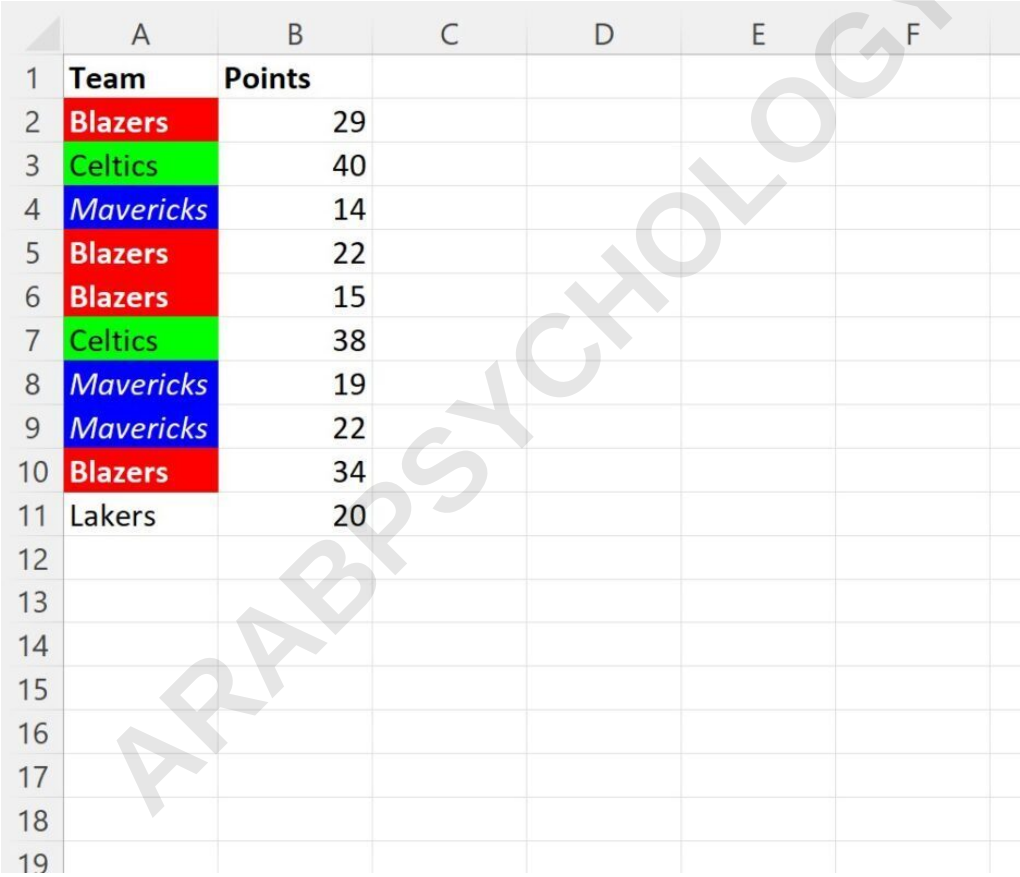
```
Sub ConditionalFormatMultiple()
Dim rg As Range
Dim cond1 As FormatCondition, cond2 As
```

```
FormatCondition, cond3 As FormatCondition

'specify range to apply conditional formatting
Set rg = Range("A2:A11")

'clear any existing conditional formatting
rg.FormatConditions.Delete

'specify rules for conditional formatting
Set cond1 = rg.FormatConditions.Add(xlCellValue,
xlEqual, "Mavericks")
Set cond2 = rg.FormatConditions.Add(xlCellValue,
xlEqual, "Blazers")
Set cond3 = rg.FormatConditions.Add(xlCellValue,
xlEqual, "Celtics")

'define conditional formatting to use
With cond1
.Interior.Color = vbBlue
.Font.Color = vbWhite
.Font.Italic = True
End WithWith cond2
.Interior.Color = vbRed
.Font.Color = vbWhite
.Font.Bold = True
```

ARABPSYCHOLOGY.COM

```
End WithWith cond3
.Interior.Color = vbGreen
.Font.Color = vbBlack
End WithEnd Sub
```

When we run this macro, we receive the following output:



	A	B	C	D	E	F
1	**Team**	**Points**				
2	**Blazers**	29				
3	Celtics	40				
4	*Mavericks*	14				
5	**Blazers**	22				
6	**Blazers**	15				
7	Celtics	38				
8	*Mavericks*	19				
9	*Mavericks*	22				
10	**Blazers**	34				
11	Lakers	20				
12						
13						
14						
15						
16						
17						
18						
19						

ARABPSYCHOLOGY.COM

Notice that cells with the team names "Mavericks", "Blazers" and Celtics" all have specific conditional formatting applied to them.

The one team with the name "Lakers" is left alone since we didn't specify any conditional formatting rules for cells with this team name.

Example 3: Remove All Conditional Formatting Rules from Cells

Lastly, we can use the following macro to remove all conditional formatting rules from cells in the current sheet:

```
Sub RemoveConditionalFormatting()
ActiveSheet.Cells.FormatConditions.Delete
End Sub
```

When we run this macro, we receive the following output:

ARABPSYCHOLOGY.COM

	A	B	C	D	E	F
1	**Team**	**Points**				
2	Blazers	29				
3	Celtics	40				
4	Mavericks	14				
5	Blazers	22				
6	Blazers	15				
7	Celtics	38				
8	Mavericks	19				
9	Mavericks	22				
10	Blazers	34				
11	Lakers	20				
12						
13						
14						
15						
16						
17						
18						

Notice that all conditional formatting has been removed from each of the cells.

ARABPSYCHOLOGY.COM