How do I center text using VBA (with example)?

Authored by

stats writer

November 18, 2025

Achieving precise text placement within Excel cells is fundamental for creating professional and readable spreadsheets. While manual formatting is possible, using VBA (Visual Basic for Applications) offers powerful, automated control over cell styling. Specifically, you can leverage the built-in **HorizontalAlignment** and **VerticalAlignment** properties within your VBA macros to programmatically center text in designated cells, controlling both horizontal and vertical positioning, respectively.

Understanding VBA Alignment Properties

When working with large datasets or developing tools for other users, relying on manual clicks for formatting is inefficient and prone to error. VBA provides the mechanism to apply consistent formatting instantly across thousands of cells. The core object we interact with is the Range object, which represents one or more cells in a worksheet. The appearance of the text within this Range is dictated by its properties, most notably the alignment settings.

The **HorizontalAlignment** property dictates how text is positioned from left to right within the cell boundaries. Common settings include aligning text to the left (`xlLeft`), aligning text to the right (`xlRight`), or, as we focus on here, centering the text (`xlCenter`). Similarly, the **VerticalAlignment** property controls the text position relative to the top and bottom of the cell. This is especially important when row heights are increased, allowing you to position text at the top (`xlTop`), bottom (`xlBottom`), or, ideally, in the middle (`xlCenter`).

To set the center alignment using either of these properties, we utilize the built-in xlCenter constant. This constant is a crucial part of the VBA enumeration system, representing the value required to trigger the centering behavior. Understanding how these properties interact with the Range object is the foundation for automating complex formatting tasks in your spreadsheets.

Three Essential VBA Alignment Methods

Depending on your specific formatting requirements, you may need to center text horizontally, vertically, or both simultaneously. Below are the three fundamental VBA methods demonstrating how to manipulate the alignment properties to achieve these precise centering outcomes.

Method 1: Center Text Horizontally Using VBA

```
Sub CenterText()
Range("A2:A11").HorizontalAlignment = xlCenter
End Sub
```

This macro specifically targets the horizontal axis. It tells VBA to find the specified Range and

apply the **HorizontalAlignment** property, setting its value to the xlCenter constant. This is the simplest approach when vertical alignment adjustments are not required or when the default vertical placement is sufficient.

Method 2: Center Text Vertically Using VBA

```
Sub CenterText()
Range("A2:A11").VerticalAlignment = xlCenter
End Sub
```

This code snippet focuses solely on vertical centering. It uses the **VerticalAlignment** property, ensuring the content is perfectly situated between the top and bottom borders of the cell. This method is crucial when dealing with wrapped text or manually adjusted row heights, preventing the text from clinging to the top or bottom edges.

Method 3: Center Text Both Horizontally & Vertically Using VBA

```
Sub CenterText()
Range("A2:A11").HorizontalAlignment = xlCenter
Range("A2:A11").VerticalAlignment = xlCenter
End Sub
```

For achieving perfect, symmetrical centering, combining both properties is the standard practice. This macro executes two distinct commands on the target Range: first setting the horizontal placement, and then setting the vertical placement. Although these commands could potentially be condensed using a `With` block for efficiency in more complex code, listing them separately clearly illustrates the manipulation of the two distinct alignment properties.

Practical Implementation: Setting Up the Scenario

To illustrate the effects of these three methods, we will apply them to a simple, unformatted dataset. This dataset consists of names and corresponding values, currently left-aligned horizontally and top-aligned vertically (the default settings in Excel). Our goal is to use VBA to adjust the alignment of the text contained within the range **A2:A11**.

The initial appearance of the data before any VBA macro is executed is shown below. Note how the text is skewed to the left and slightly higher within the cells, especially if the row height is increased.

	A	B	C	D	E	F
1	**Team**	**Points**				
2	Mavs	22				
3	Spurs	15				
4	Rockets	19				
5	Kings	24				
6	Warriors	23				
7	Nets	28				
8	Lakers	10				
9	Thunder	12				
10	Blazers	30				
11	Jazz	31				
12						
13						
14						
15						

This visual context is important, as it provides a clear benchmark against which to compare the results of the alignment macros. We will now proceed through each of the three centering methods sequentially, observing the specific change each property makes to the presentation of the data.

Example 1: Centering Text Using VBA: Horizontal Alignment

In this first example, we utilize the **HorizontalAlignment** property to reposition the text within the selected Range. This macro focuses purely on the side-to-side placement, leaving the vertical positioning unchanged from its default state (typically top alignment). This is often preferred for readability in columns where data is dense and space is limited.

```
Sub CenterText()
Range("A2:A11").HorizontalAlignment = xlCenter
End Sub
```

Upon execution of this simple procedure, the Range **A2:A11** is updated. The names are no longer hugging the left edge of the cells but are instead placed centrally along the horizontal axis. Notice that while the text is now centered horizontally, it remains vertically aligned to the top of the cell.

	A	B	C	D	E	F
1	**Team**	**Points**				
2	Mavs	22				
3	Spurs	15				
4	Rockets	19				
5	Kings	24				
6	Warriors	23				
7	Nets	28				
8	Lakers	10				
9	Thunder	12				
10	Blazers	30				
11	Jazz	31				
12						
13						
14						
15						
16						

The result clearly shows that the text in each cell within the range **A2:A11** has been successfully centered horizontally, demonstrating the effective use of the **HorizontalAlignment** property in isolating and modifying this specific formatting aspect.

Example 2: Centering Text Using VBA: Vertical Alignment

The second example addresses the vertical positioning of the cell content. This technique is particularly valuable when standardizing the appearance of tables where row heights are variable or intentionally large to accommodate graphical elements or multi-line headers. By default, Excel often top-aligns data, which can look unbalanced in taller rows.

```
Sub CenterText()
Range("A2:A11").VerticalAlignment = xlCenter
End Sub
```

Running this specific macro applies the **VerticalAlignment** property to the chosen cells. It compels the text to move precisely to the midpoint between the top and bottom borders. Crucially, this operation does not affect the existing horizontal alignment (which remains left-aligned from the

initial state unless previously modified).

	A	B	C	D	E	F
1	**Team**	**Points**				
2	Mavs	22				
3	Spurs	15				
4	Rockets	19				
5	Kings	24				
6	Warriors	23				
7	Nets	28				
8	Lakers	10				
9	Thunder	12				
10	Blazers	30				
11	Jazz	31				
12						
13						
14						
15						
16						

As visualized in the output, the text in each cell in the range **A2:A11** has been centered vertically. This adjustment significantly improves the visual balance of the data within the cells, especially where rows have greater height, confirming the successful use of the **VerticalAlignment** property.

Example 3: Centering Text Using VBA: Combined Horizontal & Vertical Alignment

For achieving the most aesthetically pleasing and symmetrical result, it is best practice to combine both alignment properties. This ensures the text is floating perfectly in the dead center of the cell, regardless of column width or row height. This combination is particularly essential for titles, headers, and key summary data points that need to stand out. We can create the following macro to center the text in each cell in the Range **A2:A11** both horizontally and vertically:

```
Sub CenterText()
Range("A2:A11").HorizontalAlignment = xlCenter
Range("A2:A11").VerticalAlignment = xlCenter
```

End Sub

The efficiency of VBA allows these two formatting commands to be executed near-simultaneously. By calling both the **HorizontalAlignment** and **VerticalAlignment** properties and assigning them the xlCenter value, we achieve comprehensive alignment correction across the entire specified Range.

	A	B	C	D	E	F
1	**Team**	**Points**				
2	Mavs	22				
3	Spurs	15				
4	Rockets	19				
5	Kings	24				
6	Warriors	23				
7	Nets	28				
8	Lakers	10				
9	Thunder	12				
10	Blazers	30				
11	Jazz	31				
12						
13						
14						
15						
16						

The final output demonstrates the successful centralization of the text. Notice that the text in each cell in the range **A2:A11** has been centered both horizontally and vertically, resulting in a cleaner, more organized, and highly professional appearance for the data within the spreadsheet.

Advanced Alignment Considerations and Best Practices

While `xlCenter` is the most common use case, VBA offers several other constants for fine-tuning alignment. For horizontal alignment, you might use `xlJustify` for text wrapping, or `xlFill` to repeat characters across the cell width. For vertical alignment, besides `xlCenter`, you have `xlTop`, `xlBottom`, and `xlJustify` (which vertically distributes text across the height of the row). Selecting the correct constant depends entirely on the specific formatting requirement of the data presentation.

For efficiency, when applying multiple properties (like both horizontal and vertical alignment) to the same Range object, it is highly recommended to use a `With` block. This streamlines the code, making it easier to read and faster to execute, as VBA does not have to resolve the `Range("A2:A11")` reference repeatedly. A refined version of the combined macro using a `With` block would look like this:

```
Sub CenterTextOptimized()
With Range("A2:A11")
.HorizontalAlignment = xlCenter
.VerticalAlignment = xlCenter
End With
End Sub
```

By mastering the **HorizontalAlignment** and **VerticalAlignment** properties, you gain robust control over the presentation of data in Excel. This programmatic approach ensures consistency, saves significant time in manual formatting, and is a cornerstone skill for any user developing custom solutions using VBA.