

How can I use mm/dd/yyyy as the date format in VBA?

Authored by
stats writer

June 21, 2024

RECOMMENDED CITATION

stats writer (2024). *How can I use mm/dd/yyyy as the date format in VBA?*.

PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=146194>

In order to use the mm/dd/yyyy date format in VBA, you can utilize the "Format" function. This function allows you to specify the desired format for a date in your VBA code. By using the format "mm/dd/yyyy", you can ensure that any dates displayed or manipulated in your VBA program will follow this specific format. Additionally, you can also change the default date format for your computer's regional settings to mm/dd/yyyy, which will automatically apply to your VBA code. This method can be useful when working with date variables or when displaying dates in user interfaces.

VBA: Use mm/dd/yyyy as Date Format

You can use the NumberFormat property in VBA to format date values using a mm/dd/yyyy format.

Here is one common way to do so in practice:

```
Sub FormatDates()
```

```
Dim i As Integer
```

```
For i = 2 To 11
```

```
Range("A" & i).NumberFormat = "mm/dd/yyyy"
```

```
Next i
```

```
End Sub
```

This particular macro will format each of the dates in the range A2:A11 to have a mm/dd/yyyy format.

The following example shows how to use this syntax in

practice.

Example: How to Use mm/dd/yyyy as Date Format in VBA

Suppose we have the following list of dates in Excel:

	A	B	C	D	E	F
1	Date					
2	12-Jan-22					
3	15-Jan-22					
4	20-Feb-23					
5	1-Mar-23					
6	5-Apr-23					
7	10-May-23					
8	1-Jun-23					
9	5-Jun-23					
10	10-Oct-23					
11	2-Jan-24					
12						
13						
14						
15						
16						
17						

Suppose we would like to format each of the dates in column A using a mm/dd/yyyy format.

We can create the following macro to do so:

Sub FormatDates()

Dim i As Integer

```
For i = 2 To 11
```

```
Range("A" & i).NumberFormat = "mm/dd/yyyy"
```

```
Next i
```

```
End Sub
```

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

	A	B	C	D	E	F
1	Date					
2	01/12/2022					
3	01/15/2022					
4	02/20/2023					
5	03/01/2023					
6	04/05/2023					
7	05/10/2023					
8	06/01/2023					
9	06/05/2023					
10	10/10/2023					
11	01/02/2024					
12						
13						
14						
15						
16						

Notice that each of the dates in column A now uses a mm/dd/yyyy format.

Note that by using mm and dd in the format, we force all months and days to be shown with two digits.

For example, the date of March 3rd, 2023 is shown as 03/01/2023.

If you'd like, you could instead use m/d/yyyy as the format to only show one digit if the day or month happens to only be one digit:

```
Sub FormatDates()
```

```
Dim i As Integer
```

```
For i = 2 To 11
```

```
Range("A" & i).NumberFormat = "m/d/yyyy"
```

```
Next i
```

```
End Sub
```

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

	A	B	C	D	E
1	Date				
2	1/12/2022				
3	1/15/2022				
4	2/20/2023				
5	3/1/2023				
6	4/5/2023				
7	5/10/2023				
8	6/1/2023				
9	6/5/2023				
10	10/10/2023				
11	1/2/2024				
12					
13					
14					
15					
16					
17					

The dates are now formatted to only show the minimum number of digits needed to accurately represent the days and months.

For example, the date of March 3rd, 2023 is shown as 3/1/2023.

Note: You can find the complete documentation for the `NumberFormat` property in VBA [here](#).