

How do you compare strings in VBA, and what are some examples of how to do so?

Authored by
stats writer

June 23, 2024

RECOMMENDED CITATION

stats writer (2024). *How do you compare strings in VBA, and what are some examples of how to do so?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=148669>

In VBA, strings can be compared using logical operators such as "equal to" (=), "not equal to" (<>), "less than" (<), "less than or equal to" (<=). These operators can be used to compare two strings character by character, starting from the leftmost character. The comparison stops when a mismatch is found, and the result is based on the value of the mismatched characters.

For example, if we have two strings "apple" and "banana", using the "less than" operator will result in "apple" being less than "banana" because "a" comes before "b" in alphabetical order.

Another way to compare strings in VBA is by using the StrComp function. This function compares two strings and returns an integer value based on the result. If the strings are equal, it returns 0. If the first string is less than the second, it returns a negative number. If the first string is greater than the second, it returns a positive number.

For instance, using the StrComp function to compare "cat" and "dog" will result in a negative number because "cat" comes before "dog" in alphabetical order.

In conclusion, there are various ways to compare strings in VBA, and the method used will depend on the specific requirements of the program. Whether it is using logical operators or the StrComp function, string comparison is essential in VBA for sorting, searching, and other operations involving strings.

Compare Strings in VBA (With Examples)

You can use the following methods in VBA to compare strings:

Method 1: Case-Sensitive String Comparison

Sub CompareStrings()

Dim i As Integer

For i = 2 To 10

Range("C" & i) = StrComp(Range("A" & i), Range("B" & i)) = 0

```
Next i  
End Sub
```

This macro will perform a case-sensitive string comparison between the strings in the corresponding cells in the ranges A2:A10 and B2:B10 and return TRUE or FALSE in the range C2:C10 to indicate whether or not the strings are equal.

Method 2: Case-Insensitive String Comparison

```
Sub CompareStrings()  
Dim i As Integer
```

```
For i = 2 To 10  
Range("C" & i) = StrComp(Range("A" & i), Range("B" &  
i), vbTextCompare) = 0  
Next i  
End Sub
```

This macro will perform a case-insensitive string comparison between the strings in the corresponding cells in the ranges A2:A10 and B2:B10.

The following examples show how to use each method

in practice with the following lists of strings in Excel:

	A	B	C	D	E
1	String 1	String 2			
2	Duck	Duck			
3	rooster	Rooster			
4	Turtle	Turtle			
5	elephant	elephant			
6	pig	PIG			
7	horse	Horse			
8	COW	cow			
9	Ant	Ant			
10	Chicken	Human			
11					
12					
13					
14					
15					
16					
17					
18					

Example 1: Case-Sensitive String Comparison in VBA

We can create the following macro to perform a case-sensitive string comparison between each corresponding string in columns A and B:

```
Sub CompareStrings()
```

```
Dim i As Integer
```

```
For i = 2 To 10
```

```
Range("C" & i) = StrComp(Range("A" & i), Range("B" & i)) = 0
Next i
End Sub
```

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

	A	B	C	D	E
1	String 1	String 2	Strings Are Equal?		
2	Duck	Duck	TRUE		
3	rooster	Rooster	FALSE		
4	Turtle	Turtle	TRUE		
5	elephant	elephant	TRUE		
6	pig	PIG	FALSE		
7	horse	Horse	FALSE		
8	COW	cow	FALSE		
9	Ant	Ant	TRUE		
10	Chicken	Human	FALSE		
11					
12					
13					
14					
15					
16					
17					

Column C returns TRUE if the strings are equal and have the same case.

Otherwise, column C returns FALSE.

Example 2: Case-Insensitive String Comparison in VBA

```
Sub CompareStrings()
```

```
Dim i As Integer
```

```
For i = 2 To 10
```

```
Range("C" & i) = StrComp(Range("A" & i), Range("B" & i), vbTextCompare) = 0
```

```
Next i
```

```
End Sub
```

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

	A	B	C	D	E
1	String 1	String 2	Strings Are Equal?		
2	Duck	Duck	TRUE		
3	rooster	Rooster	TRUE		
4	Turtle	Turtle	TRUE		
5	elephant	elephant	TRUE		
6	pig	PIG	TRUE		
7	horse	Horse	TRUE		
8	COW	cow	TRUE		
9	Ant	Ant	TRUE		
10	Chicken	Human	FALSE		
11					
12					
13					
14					
15					
16					
17					
18					
19					

Column C returns TRUE if the strings are equal, regardless of the case.

Column C only returns FALSE if the strings are not equal.

Note: You can find the complete documentation for the StrComp function in VBA .