
HARALD NAHRSTEDT

Excel + VBA

Ergänzungen

Kapitel

Einführung in VBA

Teilergebnisse mit VBA

Erstellt am 01.11.2011

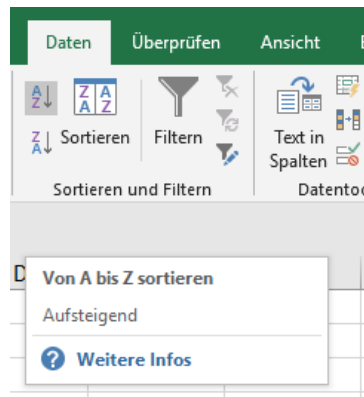
Beschreibung Dialogs sind eine Auflistung aller Dialog-Objekte. Es kann kein Dialog-Objekt gelöscht oder neu hinzugefügt werden.

1 Teilergebnisse manuell erstellen

Um Teilergebnisse in Excel manuell zu erstellen, müssen zwei Methoden aufgerufen werden.

	A	B	C
1			
2		Monat	Betrag
3		Mrz	2,65 €
4		Feb	7,23 €
5		Jan	7,23 €
6		Feb	7,64 €
7		Mrz	18,00 €
8		Mrz	18,54 €
9		Feb	18,65 €
10		Mrz	18,65 €

Im ersten Schritt wird eine beliebige Zelle in der Spalte markiert, für die ein Teilergebnis erstellt werden soll. In diesem Beispiel sollen verschiedene monatliche Kosten des ersten Quartals in einer Summe zusammengefasst werden. Danach wird diese Spalte auf- oder absteigend sortiert. Das geht am einfachsten mit den Schaltflächen in der Symbolleiste. Erst danach liefert die Methode Teilergebnis in der Menügruppe Gliederung die Teilergebnissummen nach der gewählten Spalte.



Mit dem Gliederungsschalter 2 wird nur die Anzeige der Teilergebnisse gewählt. In einem letzten Schritt können uninteressante Spalten ausgeblendet werden.

1	2	3	A	B	C
	1				
	2			Monat	Betrag
	14	+		Feb Ergebnis	428,92 €
	20	+		Jan Ergebnis	144,88 €
	31	+		Mrz Ergebnis	336,74 €
	32	-		Gesamtergebnis	910,54 €

2 Tabelle mit VBA sortieren

In einem ersten Schritt muss eine Zelle der Spalte markiert werden, nach der sortiert werden soll. Dies liefert die Anweisung `Range(Zelladresse).Select`.

Die Methode `Selection.Sort` benötigt einige Parameter, die im Code als Kommentar aufgeführt sind.

```
Sub SortiereTabelle()
    'Markierung einer Zelle in der Tabelle
    Range("B8").Select

    'Syntax
    'Ausdruck.Sort(Key1, Order1, Key2, Type, Order2,
    '   Key3, Order3, Header,
    '   OrderCustom, MatchCase, Orientation, SortMethod)
    '
    '   Key1 As Variant           erstes Sortierfeld (optional)
    '   Order1 As Variant        xlAscending aufsteigend
    '                             xlDescending absteigend
```

```

' Key2 As Variant      zweites Sortierfeld (optional)
' Type As Variant      welche Elemente sortieren (optional)
' Order2 As Variant    wie Order1
' Key3 As Variant      drittes Sortierfeld (optional)
' Order3 As Variant    wie Order1
' Header As Variant    erste Zeile enthält Überschriften:
'                      xlYes / xlNo / xlGuess
' OrderCustom As Variant Offset für Sortierreihenfolge
'                      (default = 1)
' MatchCase As Variant Groß- und Kleinschreibung beachten:
'                      True / False
' Orientation As Variant xlSortRows oder xlSortColumns
' SortMethod As Variant Sortiertyp: xlPinYin / xlStroke

Selection.Sort
Key1:=Range("B8"), _
Order1:=xlAscending, _
Header:=xlGuess, _
OrderCustom:=1, _
MatchCase:=False, _
Orientation:=xlTopToBottom
End Sub

```

Das Ergebnis ist eine sortierte Liste.

	A	B	C
1			
2		Monat	Betrag
3		Feb	7,23 €
4		Feb	7,64 €
5		Feb	18,65 €
6		Feb	24,17 €
7		Feb	33,45 €
8		Feb	37,18 €
9		Feb	43,63 €
10		Feb	43,63 €
11		Feb	47,55 €
12		Feb	78,43 €
13		Feb	87,36 €
14		Jan	7,23 €

3 In einer Tabelle mit VBA Teilergebnisse bilden

Liegt eine sortierte Tabelle vor, dann können wiederum für eine markierte Spalte Teilergebnisse abgerufen werden. Das liefert die Methode *Selection.Subtotal*. Sie benötigt ebenfalls einige Parameter, deren Erläuterung wiederum als Kommentar im Code steht.

```

Sub BildeSummen()
'Markierung einer Zelle in der Tabelle
Range("B8").Select

```

```

'Syntax
'Ausdruck.Subtotal (GroupBy, Function, TotalList,
' Replace, PageBreaks, _
' SummaryBelowData)
' GroupBy As Long      Feld, nach dem gruppiert werden soll
' Function As Long     Funktion für Teilergebnis:
'                      xlAverage / xlCount / xlCountNums /
'                      xlMax /xlMin / xlProduct / xlStDev /
'                      xlStDevP /xlSum / xlVar / xlVarP
' TotalList As Variant Matrix für Teilergebnisse
' Replace As Variant   bestehende Teilergebnisse ersetzen:
'                      True / False
' PageBreaks As Variant  Seitenwechsel einfügen: True / False
' SummaryBelowData As Variant  xlSummaryAbove / xlSummaryBelow

Selection.Subtotal _
  GroupBy:=1, _
  Function:=xlSum, _
  TotalList:=Array(2), _
  Replace:=True, _
  PageBreaks:=False, _
  SummaryBelowData:=True

'Anzeige des Gliederungslevels 2
ActiveSheet.Outline.ShowLevels RowLevels:=2
End Sub

```

Das Ergebnis zeigt die Teilsummen und die Gesamtsumme.

1	2	3	A	B	C
	1				
	2			Monat	Betrag
+	14			Feb Ergebnis	428,92 €
+	20			Jan Ergebnis	144,88 €
+	31			Mrz Ergebnis	336,74 €
-	32			Gesamtergebnis	910,54 €

4 Die zusammengefassten Prozeduren

Die nachfolgende Prozedur ist eine Zusammenfassung der beiden vorhergehenden Prozeduren.

```

Sub Gesamt ()
  Range("B8").Select

  Selection.Sort
  Key1:=Range("B8"), _
  Order1:=xlAscending, _
  Header:=xlGuess, _
  OrderCustom:=1, _
  MatchCase:=False, _

```

```
Orientation:=xlTopToBottom

Selection.Subtotal _
  GroupBy:=1, _
  Function:=xlSum,
  TotalList:=Array(2), _
  Replace:=True, _
  PageBreaks:=False, _
  SummaryBelowData:=True

ActiveSheet.Outline.ShowLevels RowLevels:=2
End Sub
```

5 Die Anzeige verschiedener Level

Die nachfolgenden Prozeduren erlauben die Anzeige verschiedener Level.

```
Sub Level_1()
  ActiveSheet.Outline.ShowLevels RowLevels:=1
End Sub
Sub Level_2()
  ActiveSheet.Outline.ShowLevels RowLevels:=2
End Sub
Sub Level_3()
  ActiveSheet.Outline.ShowLevels RowLevels:=3
End Sub
```