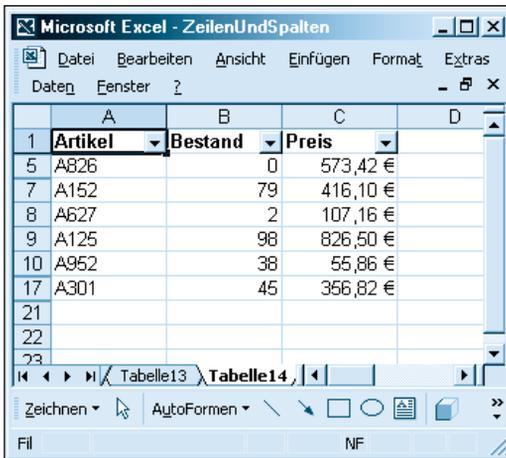


Kapitel 3

Der Zeilen/Spalten-Workshop



The screenshot shows a Microsoft Excel window titled "Microsoft Excel - ZeilenUndSpalten". The spreadsheet contains a table with three columns: "Artikel", "Bestand", and "Preis". The data is as follows:

	A	B	C	D
1	Artikel	Bestand	Preis	
5	A826	0	573,42 €	
7	A152	79	416,10 €	
8	A627	2	107,16 €	
9	A125	98	826,50 €	
10	A952	38	55,86 €	
17	A301	45	356,82 €	
21				
22				
23				

The interface includes a menu bar with "Datei", "Bearbeiten", "Ansicht", "Einfügen", "Format", and "Extras". The status bar at the bottom shows "Zeichnen", "AutoFormen", and "NF".

Nachdem wir bereits gelernt haben, wie wir mit Zellen in der Programmierung umgehen, lernen wir in diesem Kapitel, was man alles mit Zeilen und Spalten anstellen kann. Dazu gehört es, Zeilen und Spalten einzufügen, zu löschen oder ein- bzw. auszublenden. Des Weiteren können Sie in Spalten auch mit Filtern arbeiten, die es Ihnen erleichtern, umfangreiche Listen schnell auszuwerten.

Das können Sie schon:

Erste Schritte mit VBA	10
Programmieren mit Zellen und Bereichen	32

**Das lernen Sie neu:**

Zeilen und Spalten markieren	96
Zeilenhöhe und Spaltenbreite einstellen	109
Zeilen(n) einfügen bzw. löschen	112
Spalten einfügen und löschen	119
Zeilen aus- und einblenden	123
Spalten aus- und einblenden	125
Zeilen filtern	127

Zeilen und Spalten markieren

Insgesamt stehen Ihnen in Excel pro Tabelle 65.536 Zeilen zur Verfügung. Diese können Sie mit einem Klick markieren, indem Sie links oben auf den Schnittpunkt der Zeilen- und Spaltenbeschriftung klicken.

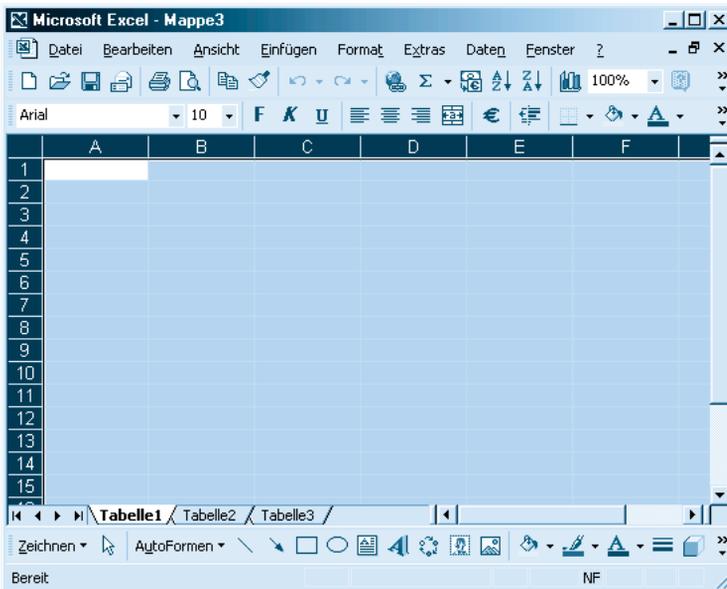
Alle Zellen markieren

Diesen manuellen Vorgang können Sie im Übrigen auch mit dem Makro aus Listing 3.1 durchführen.

```
Sub AlleZellenMarkieren()  
  Sheets("Tabelle1").Activate  
  Cells.Select  
End Sub
```

Listing 3.1: Alle Zellen der Tabelle markieren

Das Objekt `Cells` enthält alle Zellen einer Tabelle. Indem Sie die Methode `Select` auf dieses Objekt anwenden, markieren Sie alle Zellen der Tabelle. Das sind dann genau 256 Spalten x 65536 Zeilen = 16.777.216 Zellen.



Alle Zellen sind markiert



Verwendete Zellen markieren

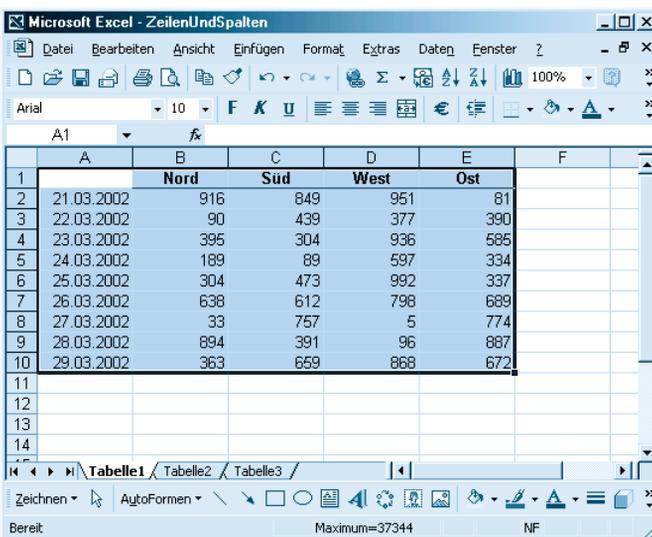
Für eine effektive Bearbeitung sollten Sie aber darauf achten, dass Sie gerade nicht alle Zeilen einer Tabelle bearbeiten, sondern besser nur die, die Sie wirklich anschließend bearbeiten möchten.

Um diese Aufgabenstellung zu lösen, setzen Sie das Makro aus Listing 3.2 ein.

```
Sub AlleBenötigtenZellenMarkieren()
    Sheets("Tabelle1").Activate
    ActiveSheet.UsedRange.Select
End Sub
```

Listing 3.2: Den verwendeten Bereich markieren

Über die Eigenschaft `UsedRange` erhalten Sie Auskunft darüber, welcher Bereich einer Tabelle mit Daten gefüllt ist. Mit der Methode `Select` markieren Sie diesen Bereich direkt im Anschluss.



Nur die benutzten Zeilen werden markiert

Angrenzende Zellen markieren

Im folgenden Beispiel werden ausgehend von einer aktiven Zelle alle umliegenden Zellen markiert. Dazu setzen Sie in der folgenden Abbildung den Mauszeiger auf die Zelle G4 und starten das Makro aus Listing 3.3.

```

Sub UmliegendeZellenMarkieren()
    Sheets("Tabelle2").Activate
    Range("G4").Select
    ActiveCell.CurrentRegion.Select
End Sub

```

Listing 3.3: Die umliegenden Zellen ermitteln und markieren

Die Eigenschaft `CurrentRegion` gibt ein `Range`-Objekt zurück, das den aktuellen Bereich darstellt. Der aktuelle Bereich wird von einer beliebigen Kombination leerer Zeilen und Spalten umschlossen. Über die Methode `Select` markieren Sie diesen Bereich.

	A	B	C	D	E	F	G	H
1		Nord	Süd	West	Ost		Gesamt	
2	21.03.2002	916	849	951	81		2.797	
3	22.03.2002	90	439	377	390		1.296	
4	23.03.2002	395	304	936	585		2.220	
5	24.03.2002	189	89	597	334		1.209	
6	25.03.2002	304	473	992	337		2.106	
7	26.03.2002	638	612	798	689		2.737	
8	27.03.2002	33	757	5	774		1.589	
9	28.03.2002	894	391	96	887		2.268	
10	29.03.2002	363	659	868	672		2.562	
11								
12								
13								
14								
15								

Umliegende Zellen markieren

Einzelne Zeile markieren

Die Markierung einer einzelnen Zeile ist kein Problem. Eine einzelne Zeile markieren Sie, indem Sie die Eigenschaft `Rows` auf das Objekt `Range` anwenden. So markieren Sie mit dem Makro aus Listing 3.4 die fünfte Zeile in der TABELLE2.

```

Sub BestimmteZeileMarkieren()
    Sheets("Tabelle2").Activate
    Rows("5:5").Select
End Sub

```

Listing 3.4: Eine bestimmte Zeile markieren



Geben Sie der Eigenschaft `Rows` bekannt, welche Zeile Sie meinen, und markieren diese über die Methode `Select`.

	A	B	C	D	E	F	G	H
1		Nord	Süd	West	Ost		Gesamt	
2	21.03.2002	916	849	951	81		2.797	
3	22.03.2002	90	439	377	390		1.296	
4	23.03.2002	395	304	936	585		2.220	
5	24.03.2002	189	89	597	334		1.209	
6	25.03.2002	304	473	992	337		2.106	
7	26.03.2002	638	612	798	689		2.737	
8	27.03.2002	33	757	5	774		1.569	
9	28.03.2002	894	391	96	887		2.268	
10	29.03.2002	363	659	868	672		2.562	
11								
12								
13								
14								
15								

Die Zeile 5 wird komplett markiert

Mehrere Zeilen auf dem Tabellenblatt markieren

Sicher wissen Sie, dass Sie manuell mehrere Zeilen auf dem Tabellenblatt markieren können. Mit dem Einsatz der `[Strg]`-Taste im Zusammenspiel mit der linken Maustaste, mit welcher Sie auf die Zeilenköpfe klicken, markieren Sie mehrere, auch nicht zusammenliegende Zeilen. Wie funktioniert dies aber mit VBA?

Im nächsten Beispiel sollen die Zeilen 2, 3, 4 und 10 auf der TABELLE2 markiert werden.

```
Sub MehrfachauswahlZeilen()
    Sheets("Tabelle2").Activate
    Range("2:2,3:3,4:4,10:10").Select
End Sub
```

Listing 3.5: Mehrere Zeilen markieren

Wenden Sie in diesem Fall die Eigenschaft `Range` an und übergeben ihr die gewünschten Zeilen, die markiert werden sollen. Die Markierung selbst wird durch die Methode `Select` durchgeführt.

	A	B	C	D	E	F	G	H
1		Nord	Süd	West	Ost		Gesamt	
2	21.03.2002	916	849	951	81		2.797	
3	22.03.2002	90	439	377	390		1.296	
4	23.03.2002	395	304	936	585		2.220	
5	24.03.2002	189	89	597	334		1.209	
6	25.03.2002	304	473	992	337		2.106	
7	26.03.2002	638	612	798	689		2.737	
8	27.03.2002	33	757	5	774		1.569	
9	28.03.2002	894	391	96	887		2.268	
10	29.03.2002	363	659	868	672		2.562	
11								
12								
13								
14								
15								

Mehrere Zeilen markieren

Einzelne Spalte markieren

Ähnlich wie schon bei der Markierung von Zeilen funktioniert es auch bei den Spalten. Eine Spalte markieren Sie normalerweise, indem Sie mit der linken Maustaste auf die Spaltenüberschrift klicken oder die Tastenkombination **Strg** + **Leer** drücken.

Im nächsten Beispiel soll die komplette Spalte G der TABELLE2 markiert werden. Setzen Sie zu diesem Zweck das Makro aus Listing 3.6 ein.

```
Sub EinzelneSpalteMarkieren()
    Sheets("Tabelle2").Activate
    Columns("G:G").Select
End Sub
```

Listing 3.6: Einzelne Spalte markieren

Eine einzelne Spalte markieren Sie, indem Sie die Eigenschaft `Columns` auf das `Range`-Objekt anwenden. Geben Sie bekannt, welche Spalte Sie genau bearbeiten möchten und markieren diese über die Methode `Select`.



	A	B	C	D	E	F	G	H
1		Nord	Süd	West	Ost		Gesamt	
2	21.03.2002	916	849	951	81		2.797	
3	22.03.2002	90	439	377	390		1.296	
4	23.03.2002	395	304	936	585		2.220	
5	24.03.2002	189	89	597	334		1.209	
6	25.03.2002	304	473	992	337		2.106	
7	26.03.2002	638	612	798	689		2.737	
8	27.03.2002	33	757	5	774		1.589	
9	28.03.2002	894	391	96	887		2.268	
10	29.03.2002	363	659	868	672		2.562	
11								
12								
13								
14								
15								

Einzelne Spalte markieren

Mehrere Spalten markieren

Wenn Sie mehrere, auch nicht zusammenhängende Spalten markieren möchten, sehen Sie sich das nächste Makro in Listing 3.7 an.

```
Sub MehrereSpaltenMarkieren()
    Sheets("Tabelle2").Activate
    Range("A:A,B:B,E:E").Select
End Sub
```

Listing 3.7: Mehrere Spalten markieren

Wenden Sie in diesem Fall die Eigenschaft `Range` an und übergeben ihr die gewünschten Spalten, die markiert werden sollen. Die Markierung selbst wird durch die Methode `Select` durchgeführt.

	A	B	C	D	E	F	G	H
1		Nord	Süd	West	Ost		Gesamt	
2	21.03.2002	916	849	951	81		2.797	
3	22.03.2002	90	439	377	390		1.296	
4	23.03.2002	395	304	936	585		2.220	
5	24.03.2002	189	89	597	334		1.209	
6	25.03.2002	304	473	992	337		2.106	
7	26.03.2002	638	612	798	689		2.737	
8	27.03.2002	33	757	5	774		1.569	
9	28.03.2002	894	391	96	887		2.268	
10	29.03.2002	363	659	868	672		2.562	
11								
12								
13								
14								
15								

Die Spalten A, B und E wurden markiert

Hinweis

Übrigens haben Sie auch die Möglichkeit, Spalten über Indices zu markieren. Wissen müssen Sie dazu, dass die Spalte A den Index 1 und die letzte Spalte IV den Index 256 hat.

Im nächsten Beispiel soll die Spalte A bis Spalte E in der TABELLE2 markiert werden. Das Makro für diese Aufgabe sehen Sie in Listing 3.8.

```
Sub MehrereSpaltenMarkiernIndex()
  Sheets("Tabelle2").Activate
  Range(Columns(1), Columns(5)).Select
End Sub
```

Listing 3.8: Mehrere Spalten über Index markieren

Wenden Sie in diesem Fall die Eigenschaft Range an und übergeben ihr die gewünschten Spaltenindizes, die markiert werden sollen. Die Markierung selbst wird durch die Methode Select durchgeführt.



	A	B	C	D	E	F	G	H
1		Nord	Sud	West	Ost		Gesamt	
2	21.03.2002	916	849	951	81		2.797	
3	22.03.2002	90	439	377	390		1.296	
4	23.03.2002	395	304	936	585		2.220	
5	24.03.2002	189	89	597	334		1.209	
6	25.03.2002	304	473	992	337		2.106	
7	26.03.2002	638	612	798	689		2.737	
8	27.03.2002	33	757	5	774		1.589	
9	28.03.2002	894	391	96	867		2.268	
10	29.03.2002	363	659	868	672		2.562	
11								
12								
13								
14								
15								

Mehrere Spalten über Index markieren

Zeilen und Spalten markieren

Um jeweils eine Spalte sowie eine Zeile in einem Arbeitsgang zu markieren, setzen Sie das Makro aus Listing 3.9 ein.

```
Sub BereicheMarkieren()
    Dim Bereich1 As Range
    Dim Bereich2 As Range
    Dim Gesamtb As Range

    Sheets("Tabelle2").Activate
    Set Bereich1 = Rows(7)
    Set Bereich2 = Columns("C:C")
    Set Gesamtb = Union(Bereich1, Bereich2)
    Gesamtb.Select
End Sub
```

Listing 3.9: Zeile und Spalte markieren

Legen Sie mit Hilfe der Anweisung Set erst einmal die Bereiche fest, die Sie gemeinsam ansprechen möchten. Danach fassen Sie die einzelnen Bereiche über die Methode Union zusammen. Danach markieren Sie den Gesamtbereich durch die Methode Select.

	A	B	C	D	E	F	G	H
1		Nord	Süd	West	Ost		Gesamt	
2	21.03.2002	916	849	951	81		2.797	
3	22.03.2002	90	439	377	390		1.296	
4	23.03.2002	395	304	936	585		2.220	
5	24.03.2002	189	89	597	334		1.209	
6	25.03.2002	304	473	992	337		2.106	
7	26.03.2002	638	612	798	689		2.737	
8	27.03.2002	33	757	5	774		1.569	
9	28.03.2002	894	391	96	887		2.268	
10	29.03.2002	363	659	868	672		2.562	
11								
12								
13								
14								
15								

Spalte und Zeile markieren

Anzahl der verwendeten Zeilen ermitteln

Die Anzahl der verwendeten Zeilen auf einem Tabellenblatt können Sie ermitteln, indem Sie die Eigenschaft `UsedRange` einsetzen, um den verwendeten Bereich auf dem Tabellenblatt herauszubekommen, sowie die Eigenschaften `Rows` und `Count`, um die Zeilen im verwendeten Bereich zu zählen.

```
Sub AnzahlVerwendeteZeilen()
Dim l As Long

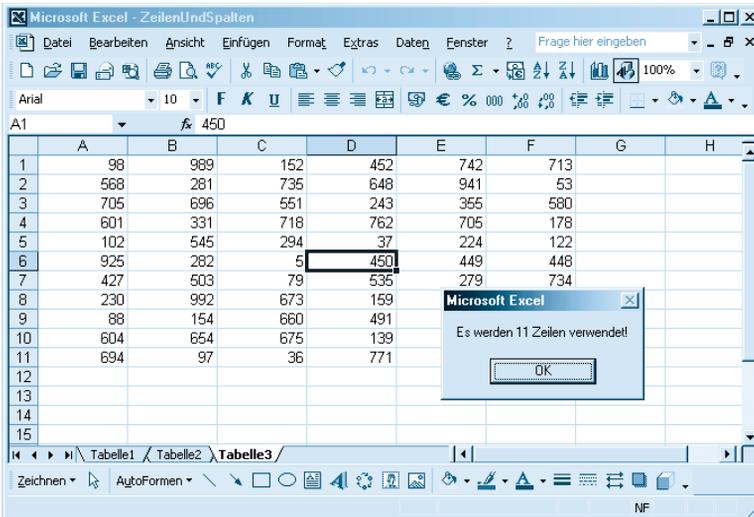
    Sheets("Tabelle3").Activate
    l = ActiveSheet.UsedRange.Rows.Count
    MsgBox "Es werden " & l & " Zeilen verwendet!"
End Sub
```

Listing 3.10: Anzahl der verwendeten Zeilen ermitteln

Achtung

Achten Sie darauf, dass Sie bei der Programmierung von Zeilen zur Sicherheit eine Variable vom Typ `Long` definieren, wenn Sie sehr viele Zeilen abarbeiten möchten. Der Datentyp `Integer` kann nur Zahlenwerte bis 32767 speichern, d.h. wenn die TABELLE3 über 32767 Zeilen hätte, dann würde es bei einem `Integer`-Typen zu einer Fehlermeldung kommen.





Die verwendeten Zeilen in der Tabelle zählen

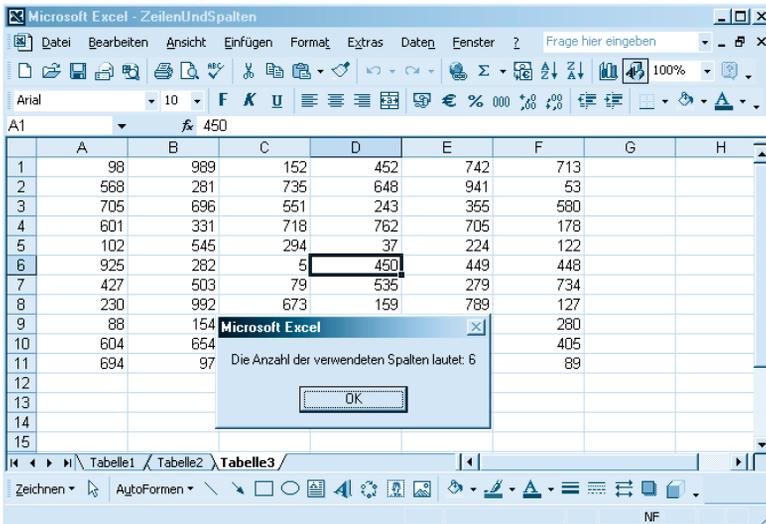
Anzahl der verwendeten Spalten ermitteln

Bei der Ermittlung der Anzahl der belegten Spalten brauchen Sie sich um die Variablendefinition keine Sorgen zu machen. Da es in einem Excel-Tabelleblatt so oder so nur 256 Spalten gibt, reicht eine Variable vom Typ Integer bei weitem aus.

```
Sub AnzahlVerwendeteSpalten()
Dim i As Integer

    Sheets("Tabelle3").Activate
    i = ActiveSheet.UsedRange.Columns.Count
    MsgBox _
    "Die Anzahl der verwendeten Spalten lautet: " & i
End Sub
```

Listing 3.11: Verwendete Spalten zählen



Die Anzahl der verwendeten Spalten ermitteln

Markieren von Zeilen ab bestimmter Position

Stellen Sie sich vor, Sie haben eine Tabelle, bei der die relevanten Daten erst ab Zeile 3 beginnen. Ihre Aufgabe besteht nun darin, alle Daten ab der Zeile 3 abwärts zu markieren. Das Makro für diese Aufgabe lautet:

```
Sub MarkierenAbZeile5()
    Sheets("Tabelle4").Activate
    Range(Range("A3"), _
    Range("A3").End(xlDown)).EntireRow.Select
End Sub
```

Listing 3.12: Zeilen markieren ab Zeile 3

Markieren Sie den datenrelevanten Bereich mit Hilfe der Methode `Select`, indem Sie im ersten Argument die Startadresse der Zelle angeben und für das zweite Argument die Eigenschaft `End` mit der Richtungskonstante `xlDown` einsetzen. Über die Eigenschaft `EntireRow` markieren Sie diese Zeilen komplett.



Nr	Bezeichnung	Preis
5647	PC	1999
5648	Drucker	500
5649	Scanner	350,95
5650	Web-Cam	159,5
5651	Monitor	567
5652	Boxen	59,99
5653	Software	159,99

Zeilen markieren ab bestimmter Zeile

Ähnlich gelagert ist auch das folgende Beispiel. In einer Tabelle sollen lediglich die letzten drei Zeilen eines Datenbereichs markiert werden. Das Makro für diese Aufgabe können Sie dem Listing 3.13 entnehmen.

```
Sub Letzte3ZeilenInSpalteMarkieren()
  Sheets("Tabelle4").Activate
  Range("A65536").End(xlUp).Select
  Range(Selection, _
  ActiveCell.Offset(-2, 0)).EntireRow.Select
End Sub
```

Listing 3.13: Die letzten Zeilen einer Liste markieren

Über die Anweisung `Range("A65536").End(xlUp).Select` ermitteln Sie die letzte gefüllte Zelle in Spalte A. Bilden Sie nun eine Markierung, die ausgehend von dieser Zelle weitere zwei Zeilen oberhalb der Zelle markiert.

	A	B	C	D	E	F	G
1	Artikelliste vom 21.03.2002						
2							
3	Nr	Bezeichnung	Preis				
4	5647	PC	1999				
5	5648	Drucker	500				
6	5649	Scanner	350,95				
7	5650	Web-Cam	159,5				
8	5651	Monitor	567				
9	5652	Boxen	59,99				
10	5653	Software	159,99				
11							
12							
13							
14							
15							

Die letzten drei Artikelzugänge in der Liste markieren

Letzte verwendete Zeile einfärben

Im nächsten Beispiel färben Sie die letzte verwendete Zeile mit der Hintergrundfarbe HELLGELB. Das Makro für diese Aufgabe entnehmen Sie dem Listing 3.14.

```
Sub LetzteZeileEinfärben()
    Sheets("Tabelle4").Activate
    Range("A65536").End(xlUp).Select
    ActiveCell.EntireRow.Select
    With Selection.Interior
        .ColorIndex = 36
        .Pattern = xlSolid
    End With
End Sub
```

Listing 3.14: Die letzte Zeile in Tabelle ermitteln und einfärben

Ermitteln Sie über die Eigenschaft `End` sowie dem Richtungsoperator `xlUp` die zuletzt verwendete Zeile. Danach markieren Sie die gesamte Zeile mit Hilfe der Eigenschaft `EntireRow`. Führen Sie daraufhin die Formatierung der Zeile durch, indem Sie den `ColorIndex` auf den Wert 36 setzen. Über die Eigenschaft `Pattern` legen Sie das Hintergrundmuster des Bereichs fest.



	A	B	C	D	E	F	G
1	Artikelliste vom 21.03.2002						
2							
3	Nr	Bezeichnung	Preis				
4	5647	PC	1999				
5	5648	Drucker	500				
6	5649	Scanner	350,95				
7	5650	Web-Cam	159,5				
8	5651	Monitor	567				
9	5652	Boxen	59,99				
10	5653	Software	159,99				
11							
12							
13							
14							
15							

Die letzte Zeile in der Liste einfärben

Zeilenhöhe und Spaltenbreite einstellen

Bei der Einstellung für die richtige Spaltenbreite bzw. die korrekte Zeilenhöhe können Sie bestimmen, ob die Einstellung für Ihr ganzes Tabellenblatt oder nur für bestimmte Spalten und Zeilen gelten soll.

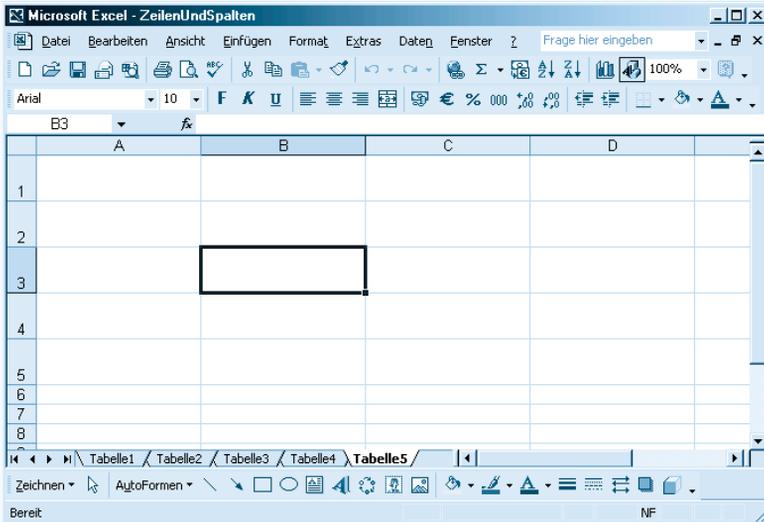
Im ersten Beispiel werden auf dem Tabellenblatt TABELLE5 die Spalten A bis C sowie die Zeilen 1 bis 5 angepasst. Dazu setzen Sie im ersten Fall die Eigenschaften `EntireColumn` und `ColumnWidth` ein, um die Spaltenbreite festzulegen. Im zweiten Fall wenden Sie die Eigenschaften `EntireRow` und `RowHeight` auf das Objekt `Range` an, um die Zeilenhöhe festzulegen.

```
Sub SpaltenZeilenEinstellen()
    Sheets("Tabelle5").Activate
    Range("A:C").EntireColumn.ColumnWidth = 20
    Range("1:5").EntireRow.RowHeight = 30
End Sub
```

Listing 3.15: Spaltenbreite und Zeilenhöhen anpassen

Hinweis

Die Einheit für die Spaltenbreite bzw. die Zeilenhöhe entspricht der Breite eines Zeichens in der Formatvorlage STANDARD. Bei Proportionalchriftarten wird die Breite des Zeichens 0 (Null) verwendet.

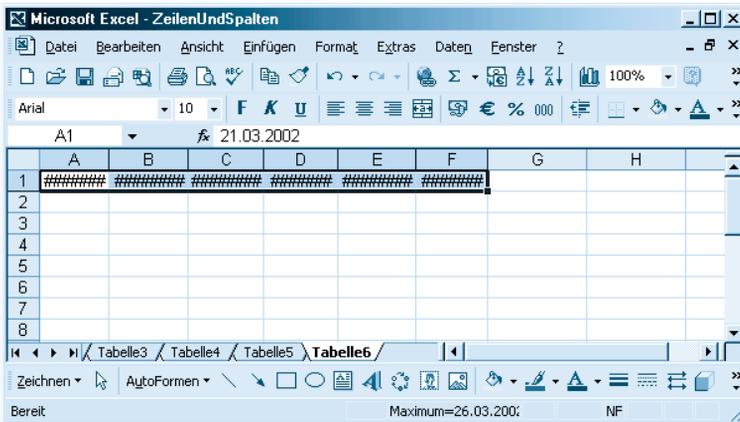


Zeilenhöhe und Spaltenbreite einstellen

Tipp

Gerade haben Sie erfahren, wie Sie in Excel Zeilen bzw. Spalten eine feste Größe geben können. Ebenso haben Sie die Möglichkeit, beispielsweise Ihren Spalten genau die Breite zu geben, die benötigt wird. Dazu verwenden Sie die Methode AutoFit, welche die angegebenen Spalten automatisch in die richtige Breite einstellt. Sicher kennen Sie die Reaktion von Excel, wenn eine Spalte zu klein ist und der eingegebene Zahlenwert nicht mehr angezeigt werden kann: Excel füllt die Zelle dann scheinbar mit dem Zeichen # auf. Erst wenn Sie die Spalte mit einem Doppelklick auf die Begrenzung des entsprechenden Spaltenkopfes anpassen, wird die Spalte so vergrößert, dass der komplette Zelleninhalt angezeigt werden kann.



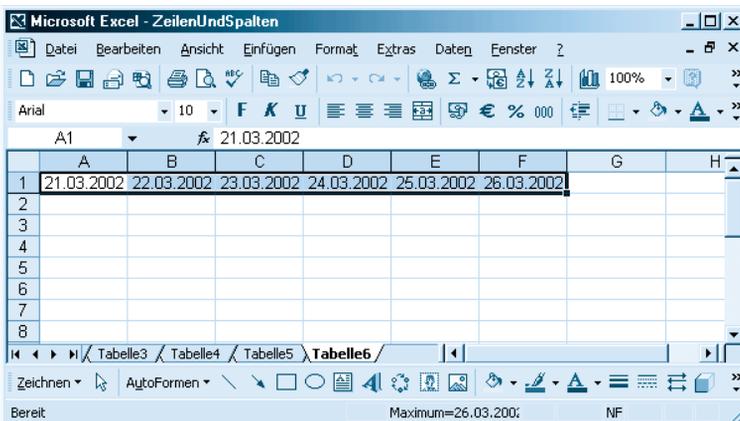


Die einzelnen Zellen sind zu klein

Um diese zu kleinen Spalten automatisch an den tatsächlichen Platzbedarf anzupassen, starten Sie das Makro aus Listing 3.16.

```
Sub SpaltenbreitenNachBedarfEinstellen()  
Worksheets("Tabelle6").Columns("A:F").AutoFit  
End Sub
```

Listing 3.16: Die Spalten A bis G werden automatisch richtig eingestellt



Alle Spalten sind automatisch angepasst worden

Zeile(n) einfügen bzw. löschen

Wenn Sie in Ihrem Tabellenblatt Zeilen hinzufügen oder auch löschen möchten, setzen Sie die Methode `Insert` bzw. die Methode `Delete` ein. Vorher muss die gewünschte Zeile komplett markiert werden. Dies erreichen Sie mit Hilfe der Anweisung `Selection.EntireRow`.

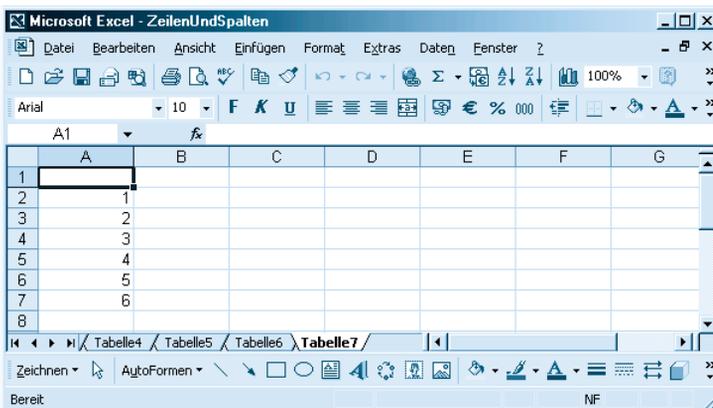
Zeile(n) einfügen

Im nächsten Beispiel legen Sie eine neue Tabelle an und geben in Spalte A einige Werte untereinander an. Ihre Aufgabe besteht nun darin, eine neue, leere Zeile vor die erste gefüllte Zeile einzufügen. Diese Aufgabenstellung lösen Sie mit dem Makro aus Listing 3.17.

```
Sub ZeileEinfügen()  
  Sheets("Tabelle7").Activate  
  Range("A1").Select  
  Selection.EntireRow.Insert  
End Sub
```

Listing 3.17: Eine einzelne Zeile einfügen

Positionieren Sie über die Methode `Select` den Mauszeiger auf die gewünschte Zelle und wenden direkt im Anschluss daran die Methode `Insert` an, um eine neue Zeile einzufügen. Damit wirklich eine ganze Zeile und nicht nur eine Zelle eingefügt wird, hängen Sie vor die Methode `Insert` die Eigenschaft `EntireRow`.



Eine leere Zeile wurde zu Beginn der Tabelle eingefügt

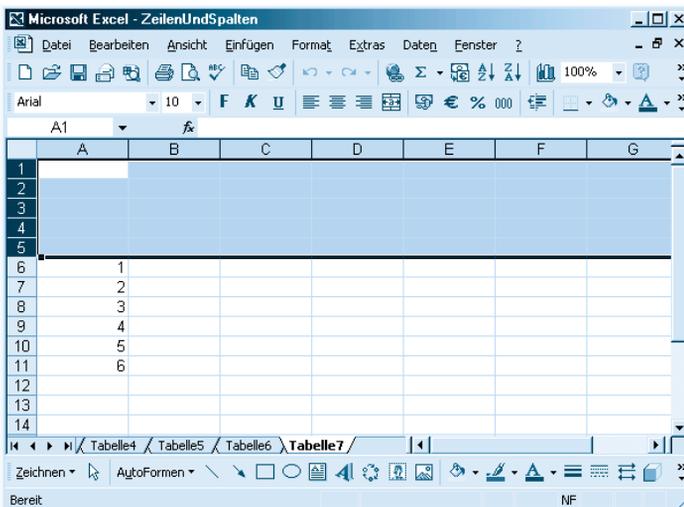


Sollen es gleich mehrere Zeilen sein, die eingefügt werden sollen, dann starten Sie das Makro aus Listing 3.18.

```
Sub ZeilenEinfügen()  
    Sheets("Tabelle7").Activate  
    Rows("1:5").Select  
    Selection.Insert Shift:=xlDown  
End Sub
```

Listing 3.18: Mehrere Zeilen einfügen

Hier werden genau fünf Zeilen oberhalb der ersten Zeile eingefügt, d.h. die übrigen Zeilen werden um fünf Zeilen nach unten geschoben. Die Eigenschaft Rows markiert die Zeilen 1 bis 5. Danach werden vier neue Zeilen über die Methode Insert eingefügt.



Fünf Zeilen zu Beginn der Tabelle einfügen

Zeile(n) löschen

Um eine Zeile in Ihrer TABELLE7 wieder zu löschen, starten Sie das Makro aus Listing 3.19.

```
Sub ZeileLöschen()  
    Sheets("Tabelle7").Activate  
    Range("A1").Select  
    ActiveCell.EntireRow.Delete  
End Sub
```

Listing 3.19: Die erste Zeile der Tabelle löschen

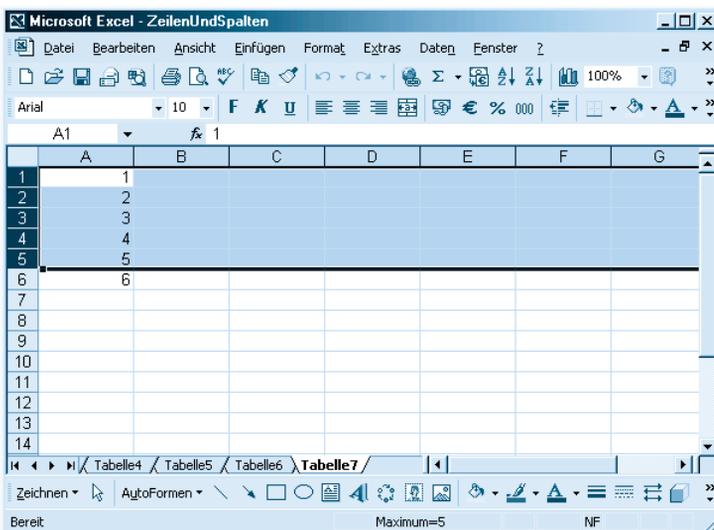
Aktivieren Sie im ersten Schritt die Tabelle und die Zelle, deren Zeile Sie löschen möchten. Wenden Sie danach die Methode `Delete` an, um die Zeile zu löschen. Damit auch wirklich die gesamte Zeile und nicht nur die aktive Zelle gelöscht wird, hängen Sie vor die Methode `Delete` noch die Eigenschaft `EntireRow`.

Um gleich mehrere Zeilen zu löschen, können Sie das Makro aus Listing 3.20 einsetzen.

```
Sub ZeilenLöschen()  
  Sheets("Tabelle7").Activate  
  Rows("1:5").Select  
  Selection.Delete Shift:=xlUp  
End Sub
```

Listing 3.20: Die ersten fünf Zeilen werden gelöscht

Markieren Sie zunächst per VBA die Zeilen, die Sie löschen möchten und wenden danach die Methode `Delete` an.



Die ersten fünf Zeilen wurden gelöscht

Bedingtes Löschen

1 Gehen Sie jetzt einen Schritt weiter und löschen Sie bestimmte Zeilen, die einer bestimmten Bedingung entsprechen.

2 Geben Sie jetzt in eine leere Tabelle in Spalte A einige Werte untereinander ein.



3 Zwischen den einzelnen Zeilen lassen Sie ruhig ein paar leere Zellen. Die Tabelle könnte dann in etwa wie folgt aussehen:

	A	B	C	D	E	F	G
1	4711						
2	4712						
3	4713						
4	4714						
5	4715						
6							
7	4801						
8	4802						
9							
10	4977						
11	4978						
12	4979						
13							
14	5096						
15	5099						
16							
17							
18							
19							

Die Ausgangssituation – eine Tabelle mit Leerzeilen

4 Wie Sie sehen sind in der Abbildung oben die Zeilen 6, 9 und 13 leer.

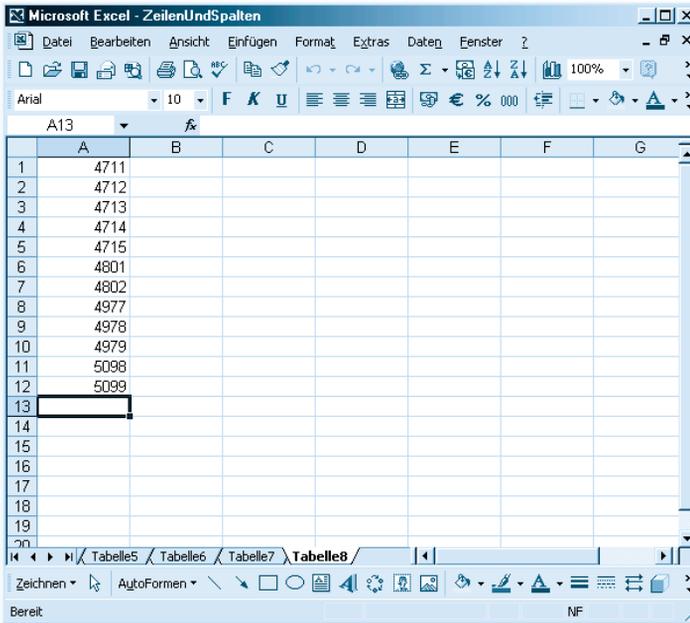
5 Erfassen Sie jetzt folgendes Makro aus Listing 3.21.

```
Sub LeerzeilenLöschen()
Dim l As Long
Dim Zl As Long

Sheets("Tabelle8").Activate
Zl = ActiveSheet.UsedRange.Rows.Count
Range("A1").Select
For l = 1 To Zl
    If Len(ActiveCell.Value) = 0 _
    Then Selection.EntireRow.Delete _
    Else ActiveCell.Offset(1, 0).Select
Next l
End Sub
```

Listing 3.21: Leerzeilen löschen

Im ersten Schritt ermitteln Sie die Anzahl der belegten Zeilen in Ihrer Tabelle. Die ermittelte Anzahl speichern Sie in der Variablen `z1` vom Typ `long`. Danach setzen Sie eine `For Next`-Schleife auf, die so lange ausgeführt wird, bis die letzte Zeile im verwendeten Bereich erreicht wird. Innerhalb der Schleife ermitteln Sie mit Hilfe der Funktion `Len` die Anzahl der eingegebenen Zeichen der Zelle. Beträgt die Zeichenlänge der Zelle null Zeichen, wird die ganze Zeile gelöscht. Im anderen Fall wird zur nächsten Zeile positioniert.

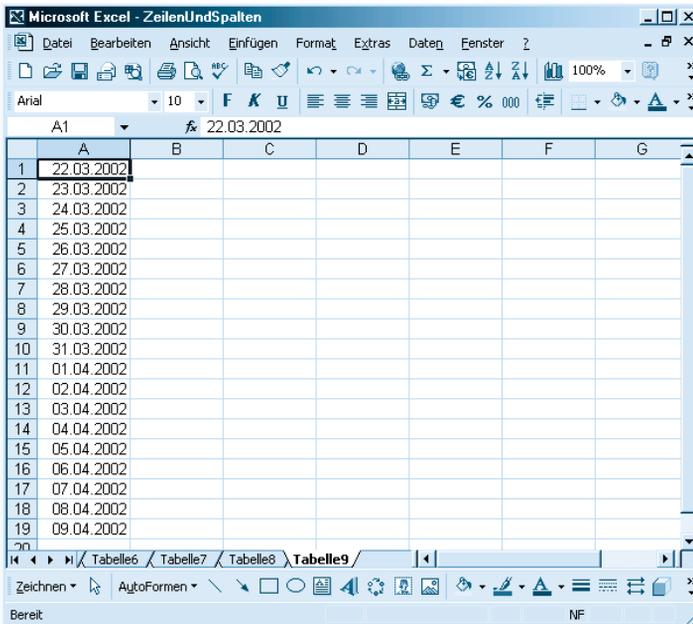


Alle Leerzeilen wurden gelöscht

Im folgenden Beispiel werden in einer Tabelle mit Datumsangaben alle Wochenendzeilen herausgelöscht. Führen Sie dazu folgende vorbereitende Schritte durch:

- 1 Setzen Sie den Mauszeiger auf Zelle A1.
- 2 Drücken Sie die Tastenkombination `Strg` + `.`.
- 3 Ziehen Sie das Ausfüllkästchen der Zelle A1 nach unten bis in Zelle A19. Sie haben jetzt eine Datumsleiste in der Spalte A.





Aus der Datumsleiste sollen die Wochenenden gelöscht werden

Da Sie auf den ersten Blick nicht erkennen können, ob es sich bei der jeweiligen Zelle um einen Wochentag oder um einen Tag am Wochenende handelt, können Sie die Löschaktion durch ein Makro durchführen lassen.

Führen Sie jedoch vorher folgenden Schritt durch, um zu testen, ob das nachfolgende Makro aus Listing 3.22 auch richtig funktioniert.

- 1** Kopieren Sie den Datenbereich A1:A19.
- 2** Fügen Sie diesen Datenbereich ab Zelle B1 ein.
- 3** Markieren Sie den Bereich B1:B19.
- 4** Weisen Sie diesem Bereich das benutzerdefinierte Format TTT zu.

	A	B	C	D	E	F	G
1	22.03.2002	Fr					
2	23.03.2002	Sa					
3	24.03.2002	So					
4	25.03.2002	Mo					
5	26.03.2002	Di					
6	27.03.2002	Mi					
7	28.03.2002	Do					
8	29.03.2002	Fr					
9	30.03.2002	Sa					
10	31.03.2002	So					
11	01.04.2002	Mo					
12	02.04.2002	Di					
13	03.04.2002	Mi					
14	04.04.2002	Do					
15	05.04.2002	Fr					
16	06.04.2002	Sa					
17	07.04.2002	So					
18	08.04.2002	Mo					
19	09.04.2002	Di					

Die Datumsleiste mit Tagesanzeige

Diese Darstellung ist schon besser und für Sie ist es auch eine Kontrolle, ob das nachfolgende Makro aus Listing 3.22 korrekt arbeitet. Erfassen Sie nun folgendes Makro:

```

Sub WochenendeLöschen()
Dim i As Long
Sheets("Tabelle9").Activate
Range("A1").Select
For i = 1 To ActiveSheet.UsedRange.Rows.Count
If Application.WorksheetFunction.Weekday _
(ActiveCell) = 1 _
Or Application.WorksheetFunction.Weekday _
(ActiveCell) = 7 _
Then ActiveCell.EntireRow.Delete _
Else ActiveCell.Offset(1, 0).Select
Next i
End Sub

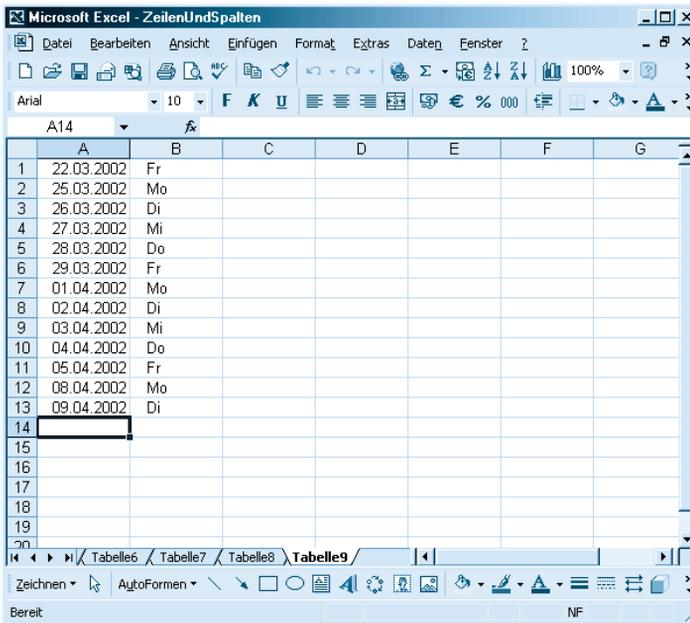
```

Listing 3.22: Alle Wochenendzeilen entfernen

Setzen Sie eine Schleife auf, die alle verwendeten Zeilen einer Tabelle durchläuft. Über die Funktion `Weekday` können Sie überprüfen, ob eine Datumzelle ein Wochenende enthält. Da Sie einer Datumsangabe schließlich



nicht ansehen können, ob dieses Datum auf ein Wochenende fällt, meldet Ihnen die Funktion `Weekday` den Wert 7 für einen Samstag und den Wert 1 für einen Sonntag zurück. In beiden Fällen löschen Sie die komplette Zeile durch die Anweisung `ActiveCell.EntireRow.Delete`.



Alle Wochenendzeilen wurden gelöscht

Spalte(n) einfügen und löschen

Das Einfügen von Spalten funktioniert in Excel analog zum Einfügen von Zeilen. Geben Sie bekannt, wo Sie eine Spalte einfügen möchten, nutzen Sie dann die Eigenschaft `EntireColumn` und fügen mit der Methode `Insert` eine Spalte ein.

Spalte(n) einfügen

Im nächsten Beispiel fügen Sie eine neue Spalte direkt vor die erste Spalte A ein. Sehen Sie sich dazu folgende Ausgangssituation an:

	A	B	C	D	E	F	G
1	Nord	Süd	West	Ost			
2	360	181	335	958			
3	356	572	721	797			
4	678	183	552	531			
5	796	659	570	803			
6	97	651	736	77			
7	445	706	831	152			
8	69	655	180	986			
9	834	221	539	197			
10	552	165	604	15			
11	648	80	169	217			
12	551	156	637	316			
13	383	222	170	945			
14	3	92	527	690			
15	152	956	600	18			

Die Ausgangssituation

Fügen Sie jetzt eine neue leere Spalte vor Spalte A ein und verwenden für diese Aufgabe das Makro aus Listing 3.23.

```
Sub SpalteEinfügen()
    Sheets("Tabelle10").Select
    Range("A1").Select
    Selection.EntireColumn.Insert
End Sub
```

Listing 3.23: Spalte einfügen

Aktivieren Sie zuerst die Tabelle, auf der Sie eine neue Spalte einfügen möchten. Danach positionieren Sie den Mauszeiger auf die Zelle, vor der Sie eine Leerspalte einfügen möchten. Diese Arbeit übernimmt dann die Methode Insert, die Sie auf die komplette Spalte (EntireColumn) anwenden.



	A	B	C	D	E	F	G
1							
2							
3			Nord	Sud	West	Ost	
4			360	181	335	958	
5			396	572	721	797	
6			678	183	552	531	
7			796	659	570	803	
8			97	651	736	77	
9			445	706	831	152	
10			69	655	180	986	
11			834	221	539	197	
12			552	165	604	15	
13			648	80	169	217	
14			551	156	637	316	
15			383	222	170	345	
16			3	92	527	690	
17			152	956	600	18	

Eine leere Spalte wurde eingefügt

Im nächsten Beispiel möchten Sie auf der Tabelle11 zwei Spalten zwischen den Spalten B und C einfügen. Sehen Sie sich vorher einmal die Ausgangssituation an.

	A	B	C	D	E	F	G
1	Datum	Standort	Umsatz				
2	22.03.2002	Köln	174				
3	23.03.2002	Bonn	943				
4	24.03.2002	München	861				
5	25.03.2002	Hannover	848				
6	26.03.2002	Düsseldorf	629				
7	27.03.2002	Düsseldorf	641				
8	28.03.2002	Düsseldorf	81				
9	29.03.2002	München	253				
10	30.03.2002	München	55				
11	31.03.2002	München	286				
12	01.04.2002	Köln	337				
13	02.04.2002	Bonn	969				
14	03.04.2002	Bonn	18				
15	04.04.2002	Bonn	297				

Zwischen Standort und Umsatz sollen zwei Spalten eingefügt werden

Erfassen Sie jetzt das Makro aus Listing 3.24.

```
Sub SpaltenEinfügen()
    Sheets("Tabelle11").Select
    Columns("C:D").Select
    Selection.Insert Shift:=xlToRight
End Sub
```

Listing 3.24: Mehrere Spalten einfügen

Denken Sie daran, dass Sie vor dem Einfügen neuer Spalten die Spalten über die Methode `Select` markieren. Da beim Makro aus Listing 3.24 die Spalten C und D markiert wurden, bedeutet dies automatisch, dass genau zwei Spalten vor Spalte C eingefügt werden. Das Einfügen der neuen Spalten wird durch die Methode `Insert` realisiert, der Sie als Konstante die Verschieberichtung `xlToLeft` (nach rechts) bekannt geben.

	A	B	C	D	E	F	G
1	Datum	Standort			Umsatz		
2	22.03.2002	Köln			174		
3	23.03.2002	Bonn			943		
4	24.03.2002	München			861		
5	25.03.2002	Hannover			848		
6	26.03.2002	Düsseldorf			628		
7	27.03.2002	Düsseldorf			641		
8	28.03.2002	Düsseldorf			81		
9	29.03.2002	München			253		
10	30.03.2002	München			55		
11	31.03.2002	München			286		
12	01.04.2002	Köln			337		
13	02.04.2002	Bonn			989		
14	03.04.2002	Bonn			18		
15	04.04.2002	Bonn			297		
16							
17							
18							
19							
20							

Es wurden zwei leere Spalten eingefügt

Spalte(n) löschen

Um eine Spalte wieder zu löschen, verwenden Sie die Methode `Delete`. Um dies an einem Beispiel zu demonstrieren, greifen Sie auf das Beispiel zurück, bei dem Sie vorher in `TABELLE10` eine Spalte eingefügt haben. Entfernen Sie diese jetzt wieder und setzen dafür das Makro aus Listing 3.25 ein.

```
Sub SpalteLöschen()
    Sheets("Tabelle10").Activate
    Range("A1").Select
    ActiveCell.EntireColumn.Delete
End Sub
```

Listing 3.25: Spalte löschen



Sollen es gleich mehrere Spalten sein, die gelöscht werden sollen, dann starten Sie das Makro aus Listing 3.26. Greifen Sie dazu auf das Beispiel zurück, bei dem Sie vorher in TABELLE11 zwei zusätzliche Spalten eingefügt haben. Entfernen Sie diese jetzt wieder.

```
Sub SpaltenLöschen()
  Sheets("Tabelle11").Activate
  Columns("C:D").Select
  Selection.Delete Shift:=xlToLeft
End Sub
```

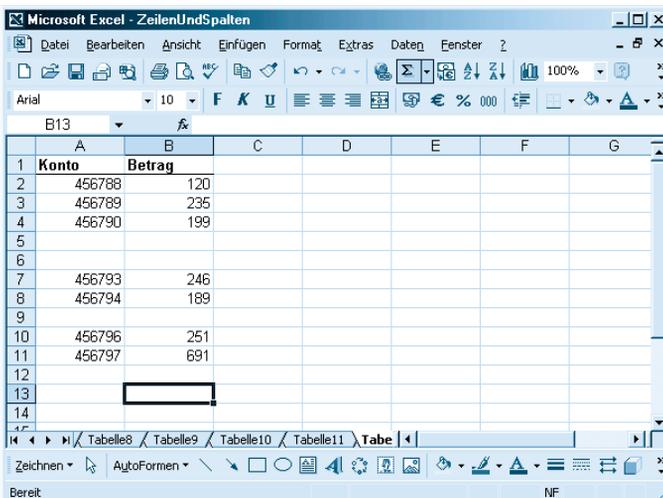
Listing 3.26: Mehrere Spalten löschen

Zeilen aus- und einblenden

Wenn Sie Informationen in Ihrer Tabelle nicht anzeigen möchten, können Sie die Informationen zeitweise ausblenden und bei Bedarf wieder einblenden.

Leere Zeilen ausblenden

Im folgenden Beispiel werden alle Leerzeilen ausblendet. Eine Zeile gilt genau dann als leer, wenn in der betreffenden Zeile in Spalte A kein Eintrag vorhanden ist. Sehen Sie sich dabei folgende Ausgangssituation an:



Die Leerzeilen sollen ausgeblendet werden

Das Makro für diese Ausgabe lautet:

```

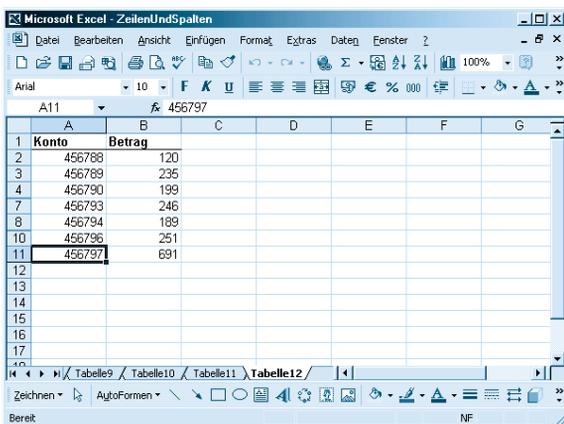
Sub AusblendenLeereZeilen()
Dim i As Long

Sheets("Tabelle12").Activate
For i = 1 To ActiveSheet.UsedRange.Rows.Count
Range("A" & i).Select
If ActiveCell.Value = "" Then
    ActiveCell.EntireRow.Hidden = True
Else
End If
Next I
End Sub

```

Listing 3.27: Jede leere Zelle wird ausgeblendet

Über die Anweisung `ActiveSheet.UsedRange.Rows.Count` ermitteln Sie die Anzahl der gefüllten Zeilen der Tabelle. Diese ermittelte Anzahl bildet die Endbedingung für die folgende Schleife. Durchlaufen Sie jetzt alle verwendeten Zellen der Spalte A und prüfen, ob diese leer sind. Wenn ja, dann blenden Sie diese Zeilen über die Anweisung `ActiveCell.EntireRow.Hidden = True` aus.



Die leeren Zeilen wurden ausgeblendet

Hinweis

Wenn Sie genau auf die Zeilenköpfe links hinsehen, dann sehen Sie, dass die Nummerierung durchbrochen ist. Nach Zeile 4 kommt direkt die Zeile 7, was bedeutet, dass die Zeilen 5 und 6 ausgeblendet wurden.



Zeilen einblenden

Um alle versteckten Zeilen einer Tabelle einzublenden, starten Sie das Makro aus Listing 3.28.

```
Sub AlleAusgeblendetenZeilenEinblenden()
Dim Zeile As Range

Sheets("Tabelle12").Activate
For Each Zeile In ActiveSheet.UsedRange.Rows
    Zeile.Hidden = False
Next
End Sub
```

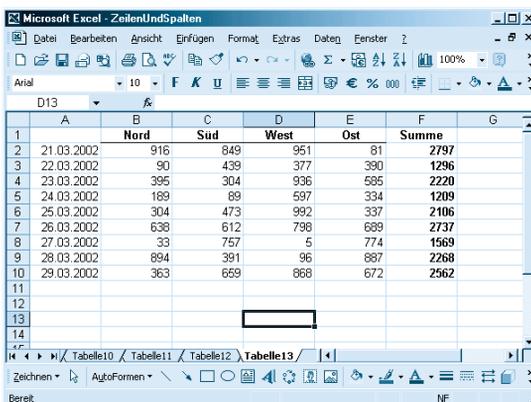
Listing 3.28: Alle ausgeblendeten Zeilen wieder einblenden

Definieren Sie zuerst eine Objektvariable, die die Zeile darstellen soll. Danach durchlaufen Sie den benutzten Bereich in einer For Each-Schleife und setzen die Eigenschaft `Hidden` jeder Zeile auf den Wert `True`. Damit blenden Sie alle versteckten Zeilen automatisch wieder ein.

Spalten ein- und ausblenden

Wenn Sie sehr umfangreiche Tabellen besitzen, bei denen Sie zwar alle Spalten benötigen, aber je nach Aufgabe jeweils nur eine Teilmenge davon brauchen, dann können Sie die momentan nicht benötigten Informationen ausblenden.

Im nächsten Beispiel werden in der TABELLE13 die Spalten B bis E nicht benötigt. Sehen Sie sich dazu einmal folgende Ausgangssituation an:



	A	B	C	D	E	F	G
		Nord	Süd	West	Ost	Summe	
1							
2	21.03.2002	916	849	951	81	2797	
3	22.03.2002	90	439	377	390	1296	
4	23.03.2002	395	304	936	595	2220	
5	24.03.2002	189	89	597	334	1209	
6	25.03.2002	304	473	992	337	2106	
7	26.03.2002	638	612	798	689	2737	
8	27.03.2002	33	757	5	774	1569	
9	28.03.2002	894	391	96	887	2268	
10	29.03.2002	363	659	868	672	2562	
11							
12							
13							
14							

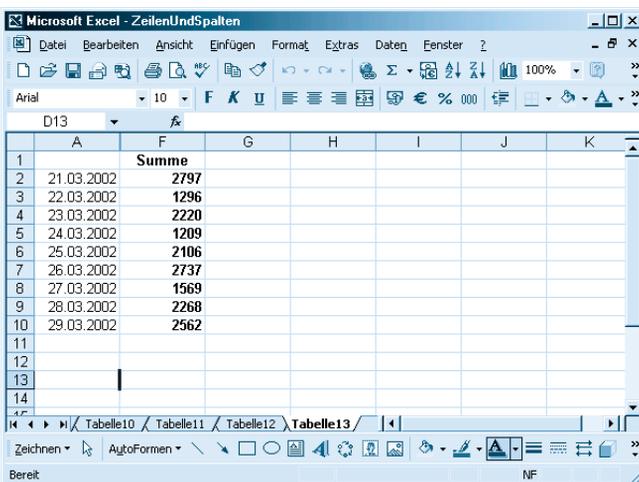
Die Spalten B bis E sollen ausgeblendet werden

Blenden Sie jetzt diese Spalten aus, indem Sie das Makro aus Listing 3.29 starten.

```
Sub SpaltenAusblenden()  
  Sheets("Tabelle13").Activate  
  Columns("B:E").EntireColumn.Hidden = True  
End Sub
```

Listing 3.29: Nicht benötigte Spalten ausblenden

Mit Hilfe der Eigenschaft `EntireColumn` sprechen Sie die Spalten B:E an. Auf diese Spalten wenden Sie dann die Eigenschaft `Hidden` an, indem Sie diese auf den Wert `True` setzen. Damit werden diese Spalten ausgeblendet.



	A	F	G	H	I	J	K
1		Summe					
2	21.03.2002	2797					
3	22.03.2002	1296					
4	23.03.2002	2220					
5	24.03.2002	1209					
6	25.03.2002	2106					
7	26.03.2002	2737					
8	27.03.2002	1569					
9	28.03.2002	2268					
10	29.03.2002	2562					
11							
12							
13							
14							

Mehrere Spalten wurden ausgeblendet

Hinweis

Wenn Sie genau in die Spaltenkopfleiste hinsehen, dann werden Sie jetzt feststellen, dass zwischen dem Buchstaben A und F eine Lücke klafft. Die Spalten B, C, D, und E sind nicht sichtbar.

Spalten einblenden

Um alle versteckten Zeilen einer Tabelle einzublenden, starten Sie das Makro aus Listing 3.30.

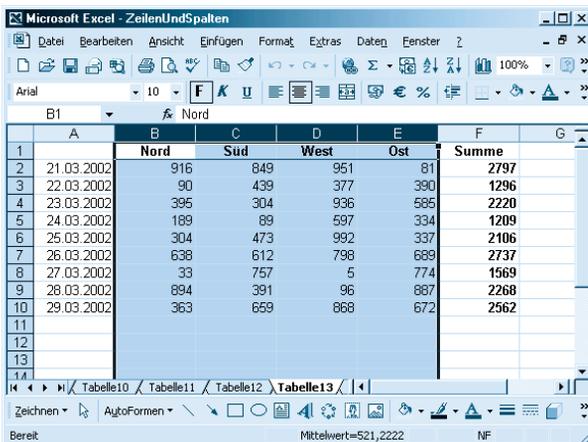


```
Sub AlleAusgeblendetenSpaltenEinblenden()  
Dim Spalte As Range
```

```
    Sheets("Tabelle13").Activate  
    For Each Spalte In ActiveSheet.UsedRange.Columns  
        Spalte.Hidden = False  
    Next  
End Sub
```

Listing 3.30: Alle versteckten Spalten einer Tabelle einblenden

Definieren Sie zuerst eine Objektvariable, die die Spalte darstellen soll. Danach durchlaufen Sie den benutzten Bereich in einer For Each-Schleife und setzen die Eigenschaft `Hidden` jeder Spalte auf den Wert `True`. Damit blenden Sie alle versteckten Spalten der Tabelle automatisch wieder ein.

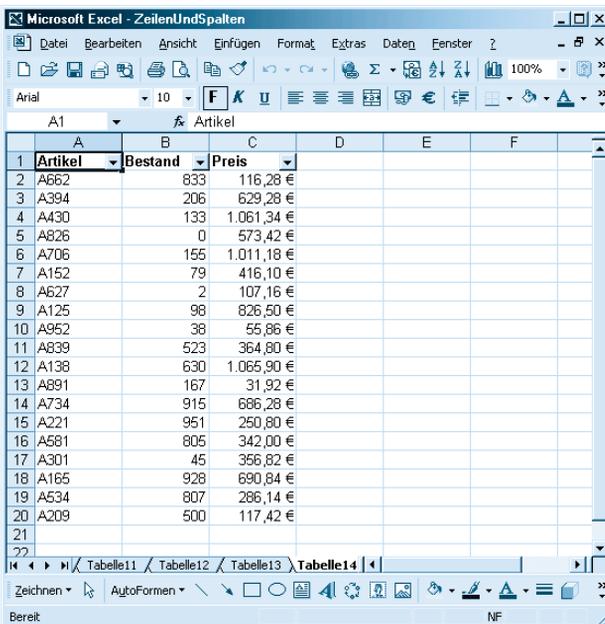


Zeilen filtern

In Excel können Sie Daten filtern. In einer gefilterten Liste werden nur die Zeilen angezeigt, die den Kriterien entsprechen, die Sie für eine Spalte angeben. Andersherum gesehen, werden beim Filtern alle Zeilen ausgeblendet, die nicht den eingestellten Kriterien entsprechen. Excel stellt zwei Befehle zum Filtern von Listen zur Verfügung: die AutoFilter mit der Möglichkeit nach der ausgewählten Zelle zu filtern für einfache Kriterien sowie die Spezialfilter für komplexere Aufgaben. Beim Filtern von Zeilen wird die Liste nicht neu angeordnet, es werden nur vorübergehend Zeilen ausgeblendet, die nicht angezeigt werden sollen.

AutoFilter aktivieren bzw. deaktivieren

Den AutoFilter können Sie auf Ihrer Tabelle aktivieren, indem Sie aus dem Menü DATEN den Befehl FILTER/AUTOFILTER wählen. Danach blendet Excel für jede Spalte im benutzten Bereich einen AutoFilter ein. Zunächst befinden sie sich im Ruhezustand, d.h. es wurde noch keine Filterung vorgenommen, was Sie auch am schwarzen Pfeil der AutoFilter erkennen können. AutoFilter, bei denen schon Filterkriterien eingestellt sind, haben einen blauen Pfeil. Beim folgenden Makro aus Listing 3.31 wird der AutoFilter, sofern er noch nicht verfügbar ist, eingeblendet.



Artikel	Bestand	Preis
A662	833	116,28 €
A394	206	629,28 €
A430	133	1.061,34 €
A826	0	573,42 €
A706	155	1.011,18 €
A152	79	416,10 €
A627	2	107,16 €
A125	96	826,50 €
A952	38	55,86 €
A839	523	364,80 €
A138	630	1.065,90 €
A891	167	31,92 €
A734	915	686,28 €
A221	951	250,80 €
A581	805	342,00 €
A301	45	356,82 €
A165	928	690,84 €
A534	807	286,14 €
A209	500	117,42 €

Der AutoFilter wurde aktiviert

```
Sub AutoFilterEinschalten()  
  Sheets("Tabelle14").Activate  
  If Not ActiveSheet.AutoFilterMode = True _  
  Then Range("A1").AutoFilter  
End Sub
```

Listing 3.31: Den AutoFilter einschalten

Die Eigenschaft `AutoFilterMode` liefert den Wert `True`, wenn die Drop-down-Pfeile für AutoFilter momentan auf dem Tabellenblatt sichtbar sind.



Liefert diese Eigenschaft den Wert `False` zurück, dann blenden Sie die AutoFilter mit Hilfe der Methode `AutoFilter` ein.

Hinweis

Wenn Sie die Filter wieder aus Ihrem Tabellenblatt verbannen möchten, setzen Sie diese Eigenschaft `AutoFilterMode` auf den Wert `False`.

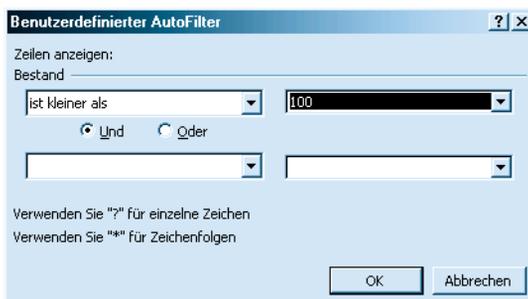
Daten filtern

Ist der AutoFilter in der Tabelle verfügbar, können einzelne Kriterien für die Spalten festgelegt werden, nach denen dann bestimmte Zeilen gefiltert werden können. Setzen Sie jetzt auf dem vorherigen Beispiel aus der Abbildung auf Seite 128 auf.

Im nächsten Beispiel sollen alle Zeilen angezeigt werden, die einen Bestand < 100 haben und daher bald nachbestellt werden müssen. Das Makro für diese Bestandsprüfung sehen Sie in Listing 3.32.

```
Sub BestandKleiner100()
    Sheets("Tabelle14").Activate
    Selection.AutoFilter _
        Field:=2, Criteria1:="<100", Operator:=xlAnd
End Sub
```

Listing 3.32: Einzelne Zeilen herausfiltern



Der benutzerdefinierte AutoFilter

Die komplette Syntax der Methode `AutoFilter` lautet:

```
Ausdruck.AutoFilter(Field, Criteria1, Operator, Criteria2, VisibleDropDown)
```

Mit der `AutoFilter`-Methode haben Sie der Spalte B (`Field:=2`) das Anzeigekriterium für Zeilen bekannt gegeben (`Criteria1:="<100"`). Bei dem Argument `Operator` haben Sie die Auswahl aus mehreren Konstanten, die Sie aus nachfolgender Tabelle entnehmen können.

Konstante	Erklärung
<code>XlAnd</code>	Es müssen zwei Kriterien erfüllt sein (<code>Criteria1</code> und <code>Criteria2</code>).
<code>XlOr</code>	Es muss entweder das eine oder das andere Kriterium erfüllt sein (<code>Criteria1</code> oder <code>Criteria2</code>).
<code>xlTop10Items</code>	Die n-höchsten Einträge aus der Liste werden angezeigt (Absolutbetrachtung).
<code>xlTop10Percent</code>	Die n-höchsten Einträge aus der Liste werden angezeigt (prozentuale Betrachtung).
<code>XlBottom10Items</code>	Die n-niedrigsten Einträge aus der Liste werden angezeigt (Absolutbetrachtung).
<code>XlBottom10Percent</code>	Die n-niedrigsten Einträge aus der Liste werden angezeigt (prozentuale Betrachtung).

Tabelle 3.1: Die Operatoren der Methode `AutoFilter`

Das Argument `Criteria2` stellt ein mögliches zweites Kriterium dar. Es wird zusammen mit `Criteria1` und dem Argument `Operator` zum Erstellen von zusammengesetzten Kriterien verwendet.

Das letzte Argument `VisibleDropDown` ist standardmäßig mit dem Wert `True` voreingestellt, was bedeutet, dass die Dropdown-Pfeile für das gefilterte Feld angezeigt werden. Wenn das Argument auf den Wert `False` gesetzt wird, werden die Dropdown-Pfeile des `AutoFilters` für das gefilterte Feld ausgeblendet.



	A	B	C	D	E	F
1	Artikel	Bestand	Preis			
5	A826	0	573,42 €			
7	A152	79	416,10 €			
8	A627	2	107,16 €			
9	A125	98	826,50 €			
10	A952	38	55,86 €			
17	A301	45	356,82 €			
21						
22						
23						
24						

Abbildung 3.35: Alle Artikel mit Bestand < 100 werden angezeigt

Die kleine Erfolgskontrolle

Beantworten Sie am Ende dieses Kapitels bitte folgenden Verständnisfragen:

1. Wie können Sie alle Zellen einer Tabelle markieren?
2. Welche Funktion hat die Anweisung `ActiveSheet.Usedrange.Select`?
3. Über welche Anweisung können Sie die letzte verwendete Zelle in Spalte B ermitteln?
4. Wie können Sie die Funktion `AutoFilter` für Ihre Tabelle per VBA einschalten?