

HANSER



Leseprobe

Daniel Koch

XML für Webentwickler

Ein praktischer Einstieg

ISBN: 978-3-446-42256-8

Weitere Informationen oder Bestellungen unter

<http://www.hanser.de/978-3-446-42256-8>

sowie im Buchhandel.

8.6 Schriftgestaltung

XSL-FO stellt zahlreiche Attribute für die Gestaltung von Texten zur Verfügung. Diese reichen von Möglichkeiten der unterschiedlichen Ausrichtung über Einrückungen bis hin zur Definition verschiedener Schriftarten. Genau darum geht es auf den folgenden Seiten. Viele der hier vorgestellten Attribute und deren Werte sind verwandt mit CSS. So wird z.B. auch in XSL-FO zum Ausrichten von Text `text-align` verwendet.

8.6.1 Ausrichtung

Für die Ausrichtung von Texten gibt es die beiden Attribute `text-align` und `text-align-last`. Über beide Attribute wird bestimmt, wie der Textinhalt von Blöcken ausgerichtet werden soll. Mit `text-align-last` lässt sich die letzte Zeile eines Blocks gesondert behandeln.

Beide Attribute kennen die folgenden Werte:

- `center` – zentrierte Ausrichtung
- `left` – linksbündig, rechts flatternd
- `right` – rechtsbündig, links flatternd
- `justify` – Blocksatz
- `inside` – linke Seite rechtsbündig, rechte Seite linksbündig
- `outside` – linke Seite linksbündig, rechte Seite rechtsbündig
- `start` – Diese Angabe ist in Verbindung mit einer anderen Schreibrichtung als `lr-tb` wichtig. Hierüber lässt sich nämlich die Bündigkeit mit dem `start`-Bereich definieren.
- `end` – Auch diese Angabe ist in Verbindung mit einer anderen Schreibrichtung als `lr-tb` wichtig. Hierüber lässt sich nämlich die Bündigkeit mit dem `end`-Bereich definieren.

Das folgende Beispiel zeigt einige typische Ausrichtungsvarianten:

Listing 8.43 Verschiedene Ausrichtungsvarianten werden eingesetzt.

```
<fo:block space-before="20mm" padding="5mm 5mm 5mm 5mm" text-align="left" space-after="1em">
  Willkommen
</fo:block>

<fo:block space-before="20mm" padding="5mm 5mm 5mm 5mm" text-align="center" space-after="1em">
  auf meinem
</fo:block>

<fo:block space-before="20mm" padding="5mm 5mm 5mm 5mm" text-align="right" space-after="1em">
  PDF-Dokument!
</fo:block>
```

Das Ergebnis sehen Sie auf **Abbildung 8.20**.

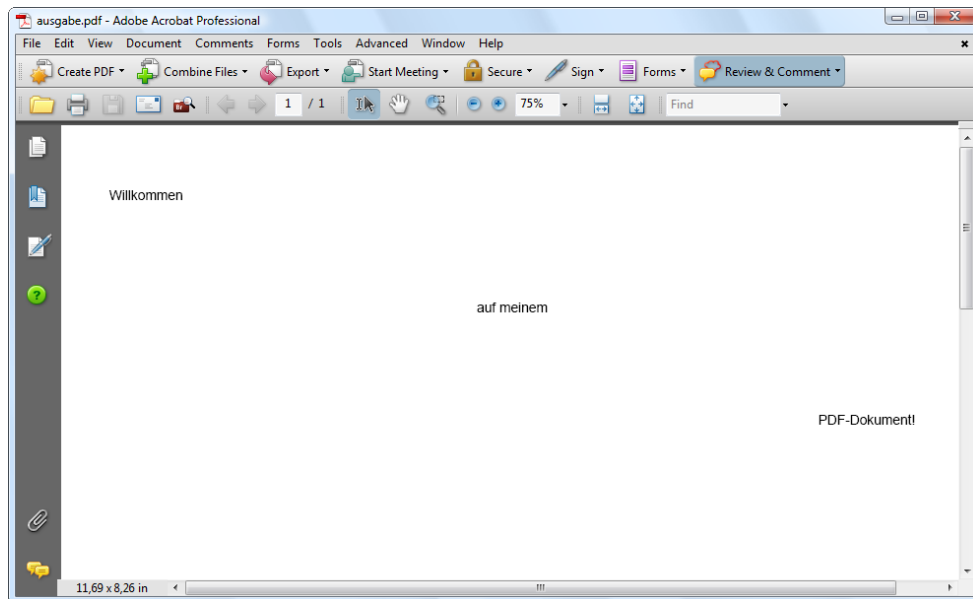


Abbildung 8.20 Alle drei Varianten in Aktion

8.6.2 Mit Einrückungen arbeiten

Für die Realisierung von Einrückungen stellt XSL-FO gleich mehrere Elemente zur Verfügung. Da wären einmal `start-indent` und `end-indent`. Mit beiden Attributen lassen sich ganze Blöcke einrichten. Das bedeutet also, dass alle Zeilen eines Blocks gleichmäßig eingerückt werden.

Listing 8.44 So werden Einrückungen definiert.

```
<fo:block start-indent="20mm">
  Und ich weiß nicht, wohin der Weg geht,
  was am Ende in deinem Buch steht, doch
  ich weiß, wir gehen schon mal vor.
</fo:block>

<fo:block space-before="2cm">
  So was verglüht nicht, so was verbrennt.
  Neid ist hässlich und Dummheit hemmt,
  und ich weiß, wir gehen schon mal vor.
</fo:block>
```

Das Ergebnis dieser Syntax zeigt **Abbildung 8.21**.

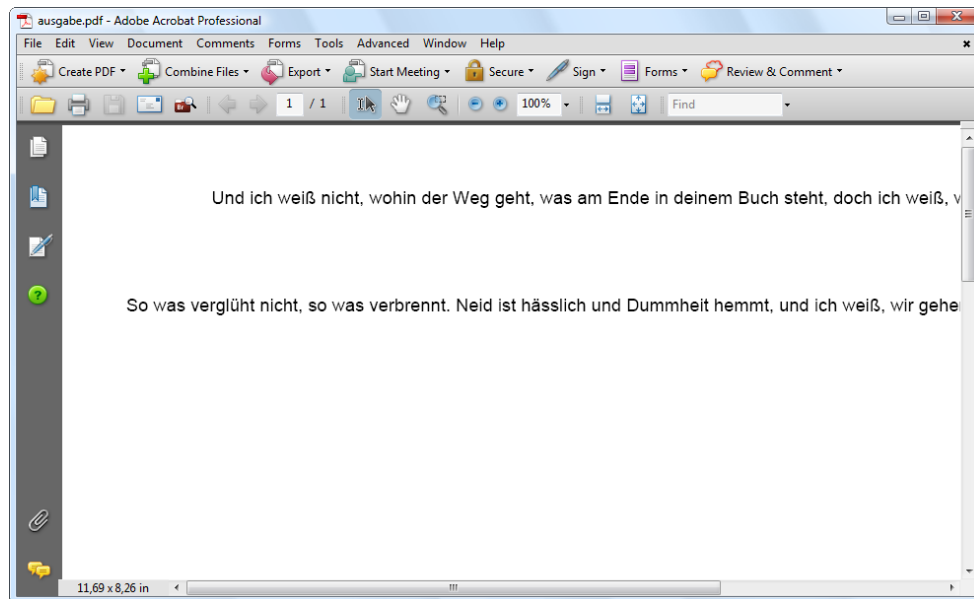


Abbildung 8.21 Hier wurde mit Einrückungen gearbeitet.

Neben `start-indent` und `end-indent` gibt es auch noch `text-indent` und `last-line-end-indent`. Das Attribut `text-indent` greift lediglich auf die erste Zeile im Startbereich zu. `last-line-end-indent` sorgt bei einem positiven Wert dafür, dass die Einrückung des Absatzes im Startbereich stattfindet. Setzt man hingegen einen negativen Wert, bewirkt das die Einrückung des Absatzes im Endbereich.

8.6.3 Schriftart festlegen

Die Schriftfamilie wird über `font-family` gesteuert. Hierbei kommt es immer wieder zu Problemen. Gerade das mag man gar nicht glauben, wenn man sich die sehr einfache Syntax ansieht.

Listing 8.45 Times wurde als Schriftart bestimmt.

```
<fo:block font-family="Times">
  Hallo, Welt!
</fo:block>
```

Dem Attribut `font-family` wird die Schriftfamilie oder eine Liste von Schriftfamilien zugewiesen. Bei einer Liste von Schriftfamilien werden die Schriften der Reihe nach im System gesucht. Ist die erste Schrift nicht vorhanden, wird nach der zweiten gesucht usw.

Die Namen der Schriftfamilien müssen durch Leerzeichen getrennt angegeben werden. Namen, die selbst Leerzeichen enthalten, muss man in Hochkommata einschließen.

```
'Times New Roman'
```

Darüber hinaus können auch generische Schriftfamilien angegeben werden. Die folgenden Schriftfamilien sind fest vordefiniert. Diese Angaben können Sie also neben Schriftarten-namen verwenden.

- `serif` = Eine Schriftart mit Serifen
- `sans-serif` = Eine Schriftart ohne Serifen
- `fantasy` = Schreibschrift/Zierschrift
- `cursive` = Eine Schriftart für Schreibschrift
- `monospace` = Eine Schriftart mit diktengleichen Zeichen

Es ist empfehlenswert, solche generischen Schriftfamilien als letzte Angabe einer `font-family`-Wertzuweisung anzugeben. Somit hat der Formatierer die Chance, eine Schriftart auszuwählen, die wenigstens vom Typ her der gewünschten entspricht, wenn diese nicht auf dem System vorhanden sein sollte.

```
font-family="Arial, sans-serif"
```

8.6.4 Schriftgröße bestimmen

Für die Definition der Schriftgröße wird das Attribut `font-size` verwendet. Es können alle in XSL-FO verfügbaren Größenangaben verwendet werden.

- `cm`
- `mm`
- `in`
- `pt`
- `pc`
- `px`
- `em`
- `%`

Normalerweise wird die Schriftgröße allerdings in Punkt (`pt`) angegeben. Ebenso kann aber auch eine prozentuale Angabe gemacht werden. Dadurch wird die Schrift relativ zur vererbten Größe, also der in der Umgebung vorherrschenden Schriftgröße, angezeigt.

Millimeter-Werte sind ebenso möglich. Dabei wird jedoch lediglich eine Stelle hinter dem Komma verwertet. Gibt man also

```
14,28mm
```

an, so wird lediglich

```
14,2mm
```

interpretiert. Das mag in vielen Fällen nicht dramatisch sein. Wenn es aber exakt werden soll, verwenden Sie besser einen Punktwert.

Listing 8.46 So wird die Schriftgröße definiert.

```
<fo:block font-family="Times" font-size="14pt">
  Hallo, Welt!
</fo:block>
```

In diesem Fall wird der Text in einer Schriftgröße von 14 Punkt angezeigt.

8.6.5 Zeilenhöhe

Über das Attribut `line-height` lassen sich die Zeilenhöhe und somit auch der Zeilenabstand bestimmen. Bei der Definition der Zeilenhöhe können sämtliche XSL-FO-Maßeinheiten angegeben werden.

- cm
- mm
- in
- pt
- pc
- px
- em
- %

Es empfiehlt sich, immer die Zeilenhöhe festzulegen. Denn fehlt sie im Stylesheet, kann es passieren, dass die Voreinstellung des jeweiligen Formatierers greift. Eine echte Kontrolle über das Schriftbild ist dann nicht möglich.

Ein Beispiel, in dem verschiedene Zeilenhöhen verwendet werden:

Listing 8.47 Hier wird mit unterschiedlichen Zeilenhöhen gearbeitet.

```
<fo:block line-height="5.5cm" text-align="justify">
  Du bist mein Leben,
  ein offenes Buch.
  Du, dieser Ort mein Start,
  mein Segen und mein Fluch.
  Und bin ich fern dann Suche
  ich dein Substitut.
  Eine Konstante wie ein Grab
  in der mein Leben ruht.
</fo:block>

<fo:block line-height="3.5cm" text-align="justify">
  Trag mich zurück zu dir
  bringt mich zum Beton.
  Legt mich auf dem Asphalt
  dem Boden von dem ich komm.
  bettet mich in grauen Staub
  lasst mich hier allein.
  Die Lichter aus meiner Stadt
  mein Heiligenschein.
</fo:block>
```

Und das Ergebnisdokument sieht folgendermaßen aus:

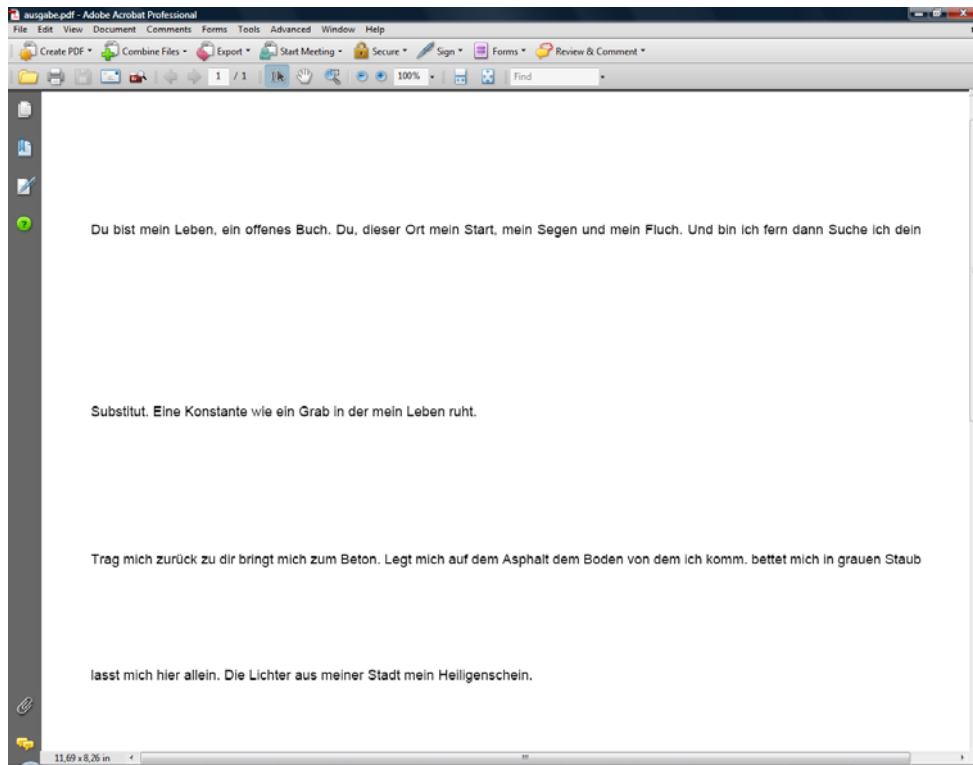


Abbildung 8.22 Hier wurden verschiedene Zeilenhöhen verwendet.

8.6.6 Unterstreichungen & Co.

Auch Unterstriche, Überstriche und durchgestrichene Texte sind möglich. Verwendet wird dafür das Attribut `text-decoration`. Wenn das Attribut auf einen Block angewendet wird, dann gilt der Attributwert für alle untergeordneten Elemente. Definiert man es hingegen für ein einzeliges Element, dann gilt der Attributwert ausschließlich für den Inhalt des betreffenden Elements.

`text-decoration` kennt die folgenden Werte:

- `underline` – Der Text wird einfach unterstrichen.
- `overline` – Der Text bekommt einen einfachen Überstrich.
- `line-through` – Der Text wird durchgestrichen.
- `none` – Der Text wird normal angezeigt. Dabei handelt es sich um die Standardeinstellung.

Beachten Sie, dass es sich bei `text-decoration` nicht um ein Font-Merkmal, sondern um ein Leistungsmerkmal des Formatierers handelt.

Im folgenden Beispiel werden die verfügbaren Werte für `text-decoration` verwendet. (Lediglich `none` wurde nicht eingesetzt.)

Listing 8.48 Drei von vier Varianten

```

<fo:block text-decoration="underline">
  Unterstrichen
</fo:block>

<fo:block text-decoration="line-through">
  Durchgestrichen
</fo:block>

<fo:block text-decoration="overline">
  Überstrichen
</fo:block>

```

Und auch hier natürlich wieder das Ergebnis:

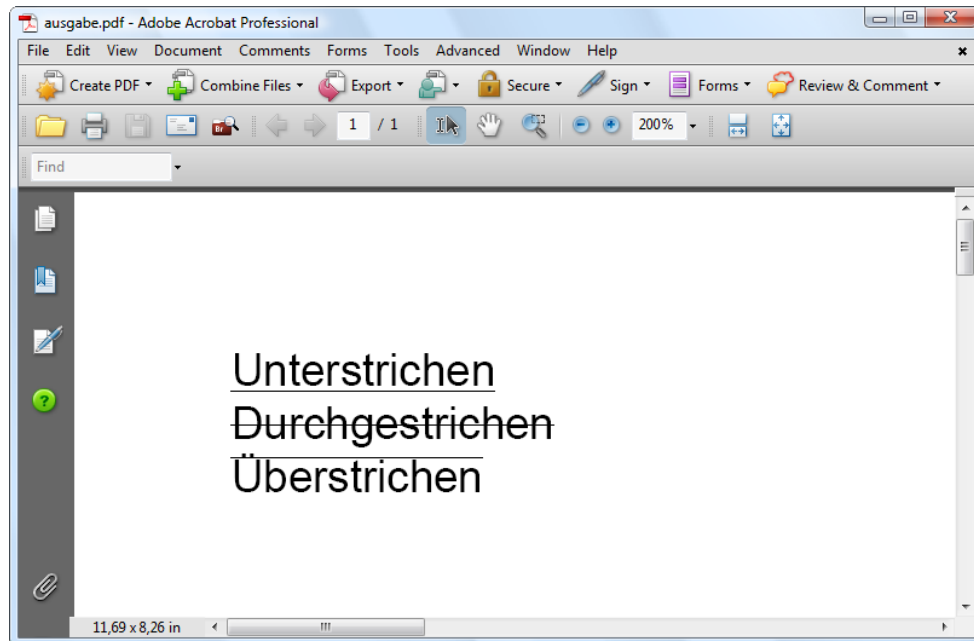


Abbildung 8.23 Die verschiedenen Textdekorationen wurden angewendet.

Sie sollten mit `text-decoration` übrigens sehr vorsichtig umgehen und dieses Attribut nur im begrenzten Maß einsetzen. Texte, bei denen permanent unter- und überstrichen wird, wirken schnell unübersichtlich.

8.6.7 Horizontale Linien innerhalb von Blöcken

Mit dem Element `fo:leader` lassen sich horizontale Linien innerhalb von Zeilen und Blöcken generieren. Bevor ins Detail gegangen wird, hier ein typisches Beispiel, wofür sich solche Linien einsetzen lassen.

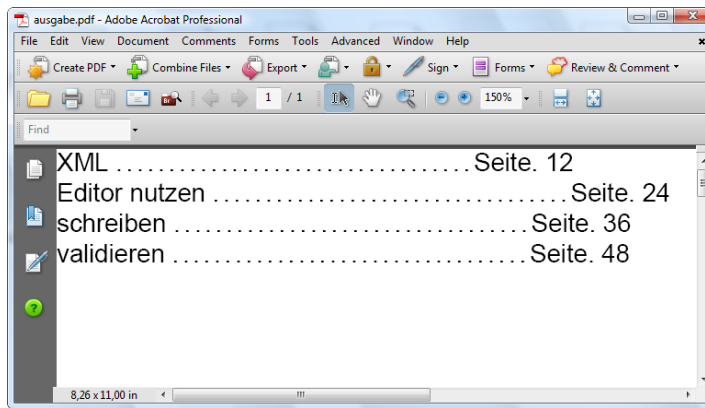


Abbildung 8.24 So könnte ein Index aussehen.

Mit wenigen Zeilen Code kann man so beispielsweise einen Index oder ein Inhaltsverzeichnis generieren. Verwendet wird für das Anlegen von Linien das Element `fo:leader`. Dieses kennt eine Vielzahl an Attributen. Laut der Spezifikation kann aus den folgenden Eigenschaften gewählt werden:

- Spracheigenschaften
- Rahmeneigenschaften
- Abstände
- Hintergründe
- Schrifteigenschaften
- Randabstände
- `alignment-adjust`
- `alignment-baseline`
- `baseline-shift`
- `color`
- `dominant-baseline`
- `text-depth`
- `text-altitude`
- `id`
- `keep-with-next`
- `keep-with-previous`
- `leader-alignment`
- `leader-length`
- `leader-pattern`
- `leader-pattern-width`
- `rule-style`
- `rule-thickness`

- letter-spacing
- line-height
- text-shadow
- visibility
- word-spacing

Die meisten dieser Eigenschaften werden derzeit von den Formatierern allerdings noch nicht unterstützt. Die folgenden Varianten können Sie aber bedenkenlos einsetzen.

- leader-length – Hierüber wird die Linienlänge bestimmt.
- leader-pattern – Damit legt man das Füllzeichenmuster fest. Mögliche Werte sind space (Leerzeichen), rule (Linie), dots (gepunktet) und use-content (ein beliebiges Zeichenmuster, das als Inhalt im fo:leader-Element enthalten ist).
- rule-style – Hierüber wird das Aussehen der Linie bestimmt. Mögliche Werte sind dotted (gepunktet), solid (durchgehend), double (doppelt), groove (3-D) und ridge (3-D).
- rule-thickness – Hierüber wird die Dicke der Linie angegeben. Standardmäßig ist die Linie ein Pixel dick.
- color – Damit kann die Farbe der Linie bestimmt werden. Standardmäßig wird bei den Formatierern Schwarz verwendet.

Das folgende Beispiel zeigt das eingangs dieses Abschnitts auf **Abbildung 8.24** zu sehende Inhaltsverzeichnis.

Listing 8.49 Linien werden definiert.

```
<fo:block text-align="start">XML
  <fo:leader leader-pattern="dots" leader-pattern-width="5pt"
    leader-alignment="reference-area" leader-length="6cm"/>Seite. 12
</fo:block>

<fo:block text-align="start">Editor nutzen
  <fo:leader leader-pattern="dots" leader-pattern-width="5pt"
    leader-alignment="reference-area" leader-length="6cm"/>Seite. 24
</fo:block>

<fo:block text-align="start">schreiben
  <fo:leader leader-pattern="dots"
    leader-pattern-width="5pt"
    leader-alignment="reference-area" leader-length="6cm"/>Seite. 36
</fo:block>

<fo:block text-align="start">validieren
  <fo:leader leader-pattern="dots"
    leader-alignment="reference-area"
    leader-pattern-width="5pt" leader-length="6cm"/>Seite. 48
</fo:block>
```

Diese Syntax zeigt, wie die einzelnen Attribute kombiniert werden. Durch die entsprechenden Längenangaben lässt sich so also z.B. ein Inhaltsverzeichnis oder Index erstellen.

Es ist bereits angeklungen, dass sich durch rule-style die Linienart bestimmen lässt. Auch hierzu wieder ein Beispiel, in dem die verschiedenen Linienvarianten verwendet werden.

Listing 8.50 Das sind die Linienarten in Aktion.

```

<fo:block text-align="center" space-before.optimum="12pt" space-
after.optimum="12pt">
<fo:leader leader-pattern="rule" leader-length="18cm" rule-style="dashed"
rule-thickness="1pt" color="black"/>
</fo:block>

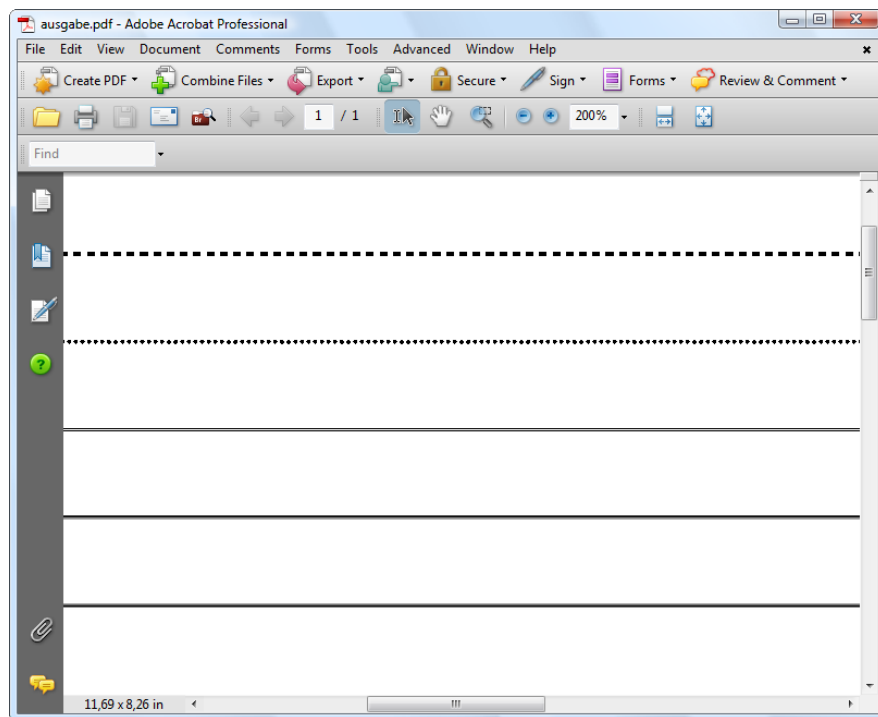
<fo:block text-align="center" space-before.optimum="12pt" space-
after.optimum="12pt">
<fo:leader leader-pattern="rule" leader-length="18cm" rule-style="dotted"
rule-thickness="1pt" color="black"/>
</fo:block>

<fo:block text-align="center" space-before.optimum="12pt" space-
after.optimum="12pt">
<fo:leader leader-pattern="rule" leader-length="18cm" rule-style="double"
rule-thickness="1pt" color="black"/>
</fo:block>

<fo:block text-align="center" space-before.optimum="12pt" space-
after.optimum="12pt">
<fo:leader leader-pattern="rule" leader-length="18cm" rule-style="groove"
rule-thickness="1pt" color="black"/>
</fo:block>

<fo:block text-align="center" space-before.optimum="12pt" space-
after.optimum="12pt">
<fo:leader leader-pattern="rule" leader-length="18cm" rule-style="ridge"
rule-thickness="1pt" color="black"/>
</fo:block>

```

Abbildung 8.25 zeigt die Linienarten im Ergebnisdokument.**Abbildung 8.25** Das sind die verfügbaren Linienarten.

8.6.8 Silbentrennung

Standardmäßig werden Texte in XSL-FO-Dokumenten nicht getrennt. Allerdings steht eine Vielzahl an Elementen bereit, mit der sich eine Silbentrennung realisieren lässt. Bei der Silbentrennung handelt es sich zunächst einmal nicht direkt um ein typografisches Merkmal. Allerdings lässt sich damit durchaus das Schriftbild beeinflussen. (Und das sowohl im positiven wie auch im negativen Sinn.)

- `hyphenate` – Bestimmt, ob eine Silbentrennung stattfinden soll oder nicht. Mögliche Werte sind `true` (findet statt) und `false` (findet nicht statt).
- `hyphenation-character` – Hierüber wird das Trennzeichen festgelegt. Als Standardzeichen wird der herkömmliche Trennstrich verwendet.
- `hyphenation-keep` – Die Silbentrennung wird am Ende einer Spalte oder einer Seite durchgeführt. Durch die Werte `column` und `page` kann die Silbentrennung am Ende einer Spalte oder einer Seite verhindert werden.
- `hyphenation-ladder-count` – Damit wird die Anzahl der maximal aufeinanderfolgenden Zeilen angegeben, für die die Silbentrennung erlaubt sein soll.
- `hyphenation-push-character-count` – Gibt für ein Wort die Mindest-Buchstabenanzahl an, die nach dem Trennzeichen in der nächsten Zeile stehen muss. Der Standardwert ist 2.
- `hyphenation-remain-character-count` – Gibt für ein Wort die Mindest-Buchstabenanzahl an, die vor dem Trennzeichen stehen muss. Der Standardwert ist 2.
- `country` – Hier wird das Land angegeben, dessen Regeln bei der Silbentrennung berücksichtigt werden sollen. Für Deutschland geben Sie `de` an.
- `language` – Darüber gibt man die verwendete Sprache an. Für Deutschland notieren Sie `de`.

Ein Beispiel, wie sich die verschiedenen Attribute kombinieren lassen.

Listing 8.51 Die Silbentrennung wird festgelegt.

```
<fo:block hyphenate="true"
  hyphenation-character="!"
  hyphenation-push-character-count="2"
  hyphenation-remain-character-count="3"
  language="de">
  Bitte nur seeeeeeeeeeeeeeeeehr lange Wörter trennen
</fo:block>
```

Die Trennung funktioniert nur, wenn die Sprache des Textes angegeben wurde. Das geschieht entweder im `fo:block`-Element selbst, in einem ihm hierarchisch übergeordneten `fo:block`-Element oder innerhalb von `fo:page-sequence`.

8.6.9 Groß- und Kleinschreibung

Über das Attribut `text-transform` kann die Umwandlung von Groß- und Kleinbuchstaben erreicht werden.

8.10 Grafiken einbinden

Selbstverständlich lassen sich auch Grafiken einbinden. Verwendet wird dafür das Element `fo:external-graphic`. Als Attribut muss zumindest `src` angegeben werden. Diesem Attribut weist man den Pfad zur und den Namen der Grafik zu, die angezeigt werden soll. Hier gelten die gleichen Regeln, die vom Einbinden von Grafiken aus HTML her bekannt sind.

Gibt man nur `src` an, wird das Bild in seiner Originalgröße angezeigt. Anhand zweier Attribute können Sie Einfluss auf die Bildergröße nehmen.

- `content-width` – Hierüber gibt man die Breite der Grafik an. Als Standardwert wird `auto` angenommen. Dadurch wird die Grafik in ihrer Originalbreite angezeigt. Durch `scale-to-fit` wird die Grafik an die Größe eines vorgegebenen Rechtecks angepasst. Ebenso sind auch prozentuale und absolute Breitenangaben möglich.
- `content-height` – Durch dieses Attribut legt man die Höhe der Grafik fest. Ansonsten gilt das Gleiche, was zuvor im Zusammenhang mit `content-width` beschrieben wurde.

Wenn man lediglich für eine Dimension eine Größenangabe macht, wird die andere Dimension proportional angepasst.

Listing 8.59 Die gleiche Grafik wird drei Mal eingebunden.

```
<fo:block>
  <fo:inline>
    <fo:external-graphic src="image.jpg"/>
  </fo:inline>
  <fo:inline>
    <fo:external-graphic src="image.jpg" content-width="40mm"/>
  </fo:inline>
  <fo:inline>
    <fo:external-graphic src="image.jpg" content-width="40mm"
      content-height="40mm"/>
  </fo:inline>
</fo:block>
```

Abbildung 8.34 zeigt, wie die Grafiken jeweils eingebunden werden.

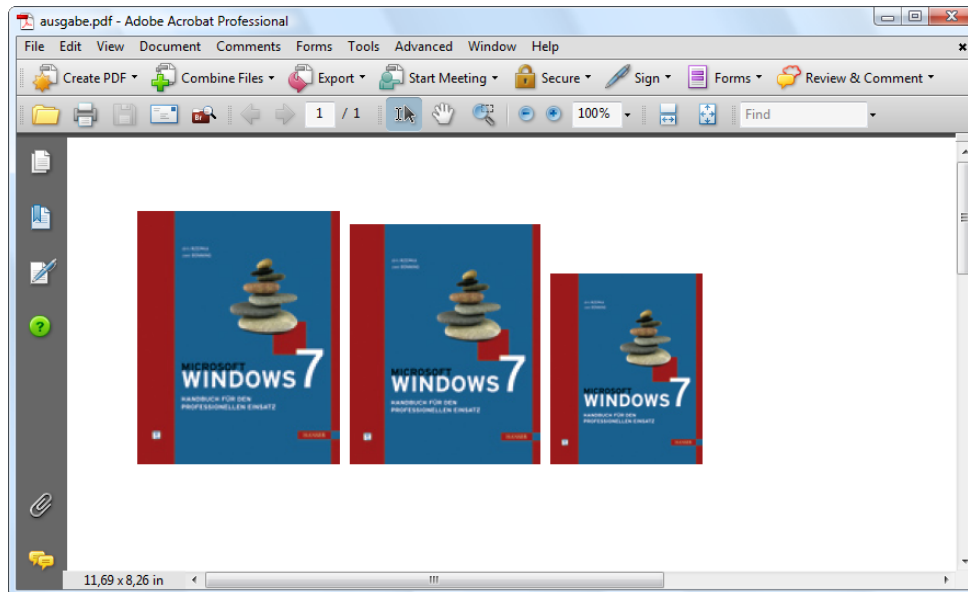


Abbildung 8.34 Die gleiche Grafik, aber in unterschiedlichen Größen

Um Verzerrungen zu vermeiden, sollte man in jedem Fall die Maße exakt angeben oder nur die Breite oder nur die Höhe des Bildes definieren.

Die Grafik muss korrekt referenziert werden, da der Formatierer ansonsten die Generierung des Dokuments mit einer Fehlermeldung abbricht.

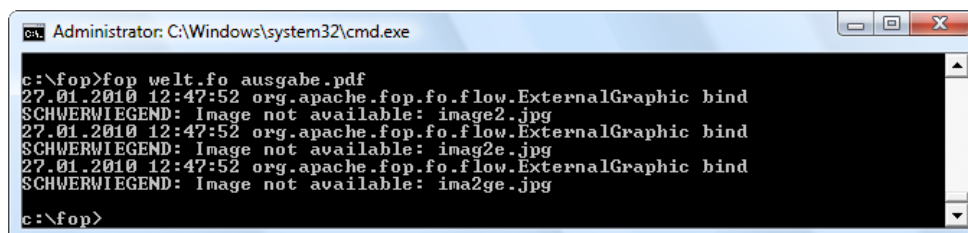


Abbildung 8.35 Die Grafiken wurden nicht gefunden.

Hier konnte die Grafik nicht eingebunden werden.

8.10.1 Hintergrundbilder definieren

Auch Hintergrundbilder lassen sich einbinden. Für diesen Zweck stellt die Spezifikation zahlreiche Attribute zur Verfügung, die Ihnen so vielleicht bereits von CSS her bekannt sind. Bevor die Attribute aber im Einzelnen vorgestellt werden, ein einfaches Beispiel, in dem eine Hintergrundgrafik eingebunden wird.

Listing 8.60 Die Hintergrundgrafik wird eingebunden.

```
<fo:block background-image="background.jpg"
background-repeat="x-repeat" font-size="40pt">
  Herzlich Willkommen
</fo:block>
```

Und jetzt die verfügbaren Attribute.

- `background-image` – Dieses Attribut erwartet als Wert den Pfad zur und den Namen der Grafik.
- `background-repeat` – Hierüber wird festgelegt, ob und wie die Hintergrundgrafik wiederholt werden soll. Der Vorgabewert ist `repeat`. Dadurch wird die Grafik vertikal und horizontal so oft wiederholt, bis der gesamte Platz, für den die Grafik gelten soll, ausgefüllt ist. Durch `x-repeat` erfolgt, wenn es nötig ist, eine horizontale Wiederholung der Grafik. Setzt man hingegen `y-repeat` ein, wird die Grafik vertikal wiederholt. Will man die Wiederholung sowohl vertikal wie auch horizontal unterbinden, setzt man `no-repeat` ein. Beachten Sie, dass FOP die beiden Werte `x-repeat` und `y-repeat` nicht korrekt interpretiert.
- `background-position-vertical` – Über dieses Attribut wird die vertikale Positionierung innerhalb des Block-Rechtecks ggf. inklusive der `padding`-Fläche angegeben. Der Vorgabewert ist `0%`, der am oberen Rand des Rechtecks beginnt. Durch andere Werte kann man das Hintergrundbild innerhalb des Rechtecks verschieben. Neben numerischen Werten sind auch noch `top` (entspricht `0%`), `center` (entspricht `50%`) und `bottom` (entspricht `100%`) möglich.
- `background-position-horizontal` – Das ist das Gegenstück zu `background-position-vertical`. Allerdings wird bei `background-position-horizontal` die horizontale Positionierung innerhalb des gegebenen Block-Rechtecks bestimmt. Auch hier sind wieder numerische Angaben oder `top`, `center` und `bottom` möglich.
- `background-position` – Dabei handelt es sich um eine verkürzte Syntax für die Positionierung der Hintergrundgrafik. Der erste Wert entspricht `background-position-vertical`, der zweite `background-position-horizontal`.
- `background-color` – Hiermit geben Sie die Hintergrundfarbe an. Interessant ist diese z.B., wenn die definierte Hintergrundgrafik – aus welchem Grund auch immer – nicht angezeigt werden kann.

Mit dem Einsatz von Hintergrundgrafiken sollte man sehr sparsam umgehen. Das gilt insbesondere, wenn die Grafik hinter Text gelegt werden soll. Oftmals ist dann der eigentliche Text nämlich nur noch schwer oder überhaupt nicht mehr zu lesen.

Abbildung 8.36 zeigt eindrucksvoll, wie man es nicht machen sollte.



Abbildung 8.36 Hier kann man den Text kaum lesen.

8.10.2 SVG einbinden

Die Formatting Objects wurden für die Anzeige von Textdokumenten entwickelt. Es besteht allerdings auch die Möglichkeit, Grafiken, die im Binärformat vorliegen, einzubinden. Der entsprechende Elementtyp lautet `external-graphic`. Wie dieser einzusetzen ist, wurde auf den vorherigen Seiten beschrieben.

Anders sieht es aber aus, wenn Grafiken eingebunden werden sollen, die in XML-Formaten vorliegen. Ein typisches Anwendungsbeispiel dafür ist SVG. Aber auch solche Grafiken können eingebunden werden. Dabei wird allerdings der entsprechende XML-Quelltext in das XSL-FO-Dokument übernommen. Zum Einsatz kommt hier das Element `instream-foreign-object`. Die Kinder eines solchen Elements kommen aus anderen Namensräumen als dem XSL-Namensraum. Welche Formate letztendlich unterstützt werden, hängt vom verwendeten XSL-Prozessor ab.

Integrieren lassen sich z.B. SVG-Grafiken. Wie das funktioniert, zeigt das folgende Beispiel. Zum Einbinden der XML- bzw. SVG-Syntax wird `fo:instream-foreign-object` verwendet.

Listing 8.61 SVG wird integriert.

```
<?xml version="1.0" encoding="UTF-8"?>
<fo:root xmlns:fo="http://www.w3.org/1999/XSL/Format" >
  <fo:layout-master-set>
    <fo:simple-page-master master-name="simple">
      <fo:region-body/>
    </fo:simple-page-master>
  </fo:layout-master-set>
```

```

<fo:page-sequence master-reference="simple">
  <fo:flow flow-name="xsl-region-body">
    <fo:block>

      <fo:instream-foreign-object>
        <svg:svg width="200pt"
          height="100pt"
          xmlns:svg="http://www.w3.org/2000/svg" >

          <svg:circle cx="60pt"
            cy="50pt"
            r="50pt"
            style="fill:red;"/>
          </svg:svg>

        </fo:instream-foreign-object>
      </fo:block>
    </fo:flow>
  </fo:page-sequence>
</fo:root>

```

Innerhalb des `fo:instream-foreign-object`-Elements steht herkömmliche SVG-Syntax. Im aktuellen Beispiel wird einfach ein roter Kreis generiert. Das Ergebnis sieht folgendermaßen aus:

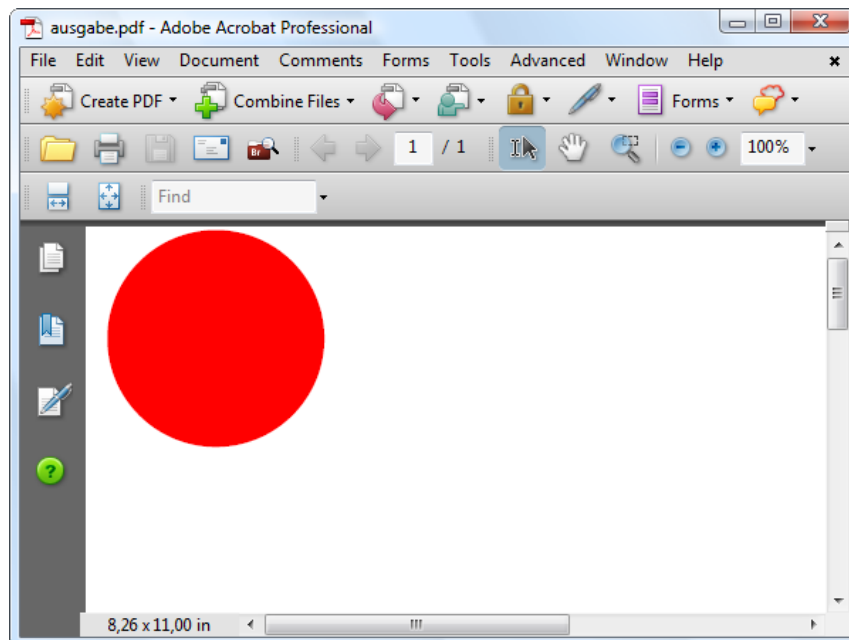


Abbildung 8.37 Ein SVG-Element wurde integriert.

Die Sache hat natürlich einen entscheidenden Haken. Animationen, wie sie sonst mittels SVG möglich sind, lassen sich in PDF- oder anderen „statischen“ Elementen leider nicht realisieren.

FOP erlaubt es übrigens, dass Sie Ihre normalen XSL-FO-Dokumente in das SVG-Format umwandeln. Dafür muss bei FOP lediglich das Ausgabeformat SVG angegeben werden.

8.11 Mit Tabellen arbeiten

Tabellen kennen Sie aus Textverarbeitungsprogrammen und z.B. aus HTML. Auch in XSL-FO lassen sich Tabellen definieren. Die Tabellenstruktur in XSL-FO ist allerdings nicht mit der aus HTML vergleichbar.

Listing 8.62 Eine einfache Tabelle wird angelegt.

```
<fo:table>
  <fo:table-body>
    <fo:table-row>

      <fo:table-cell>
        <fo:block>Zelle 1</fo:block>
      </fo:table-cell>

      <fo:table-cell>
        <fo:block>Zelle 2</fo:block>
      </fo:table-cell>

      <fo:table-cell>
        <fo:block>Zelle 3</fo:block>
      </fo:table-cell>

      <fo:table-cell>
        <fo:block>Zelle 4</fo:block>
      </fo:table-cell>

    </fo:table-row>
  </fo:table-body>
</fo:table>
```

Für die Definition von Tabellen sind mehrere Elemente zuständig.

- `fo:table` – Hierüber wird die Tabelle erzeugt.
- `fo:table-header` – Erzeugt den Kopfbereich der Tabelle. Dieser Kopfbereich wird, wenn sich die Tabelle über mehrere Seiten erstreckt, wiederholt.
- `fo:table-body` – Darin wird der eigentliche Inhalt der Tabelle definiert.
- `fo:table-footer` – Hierüber wird der Fußbereich der Tabelle erzeugt.
- `fo:table-and-caption` – Wenn die Tabelle eine Überschrift oder Legende besitzt, handelt es sich bei diesem Element um das oberste und obligatorische Tabellenelement.
- `fo:table-caption` – Nimmt die Überschrift oder Legende auf, die als oberstes Element `fo:table-and-caption` besitzt.
- `fo:table-row` – Darüber werden Tabellenzeilen definiert.
- `fo:table-cell` – Hierüber bestimmt man die Tabellenzellen.

Abbildung 8.38 zeigt das Ergebnis der zuvor definierten Syntax.

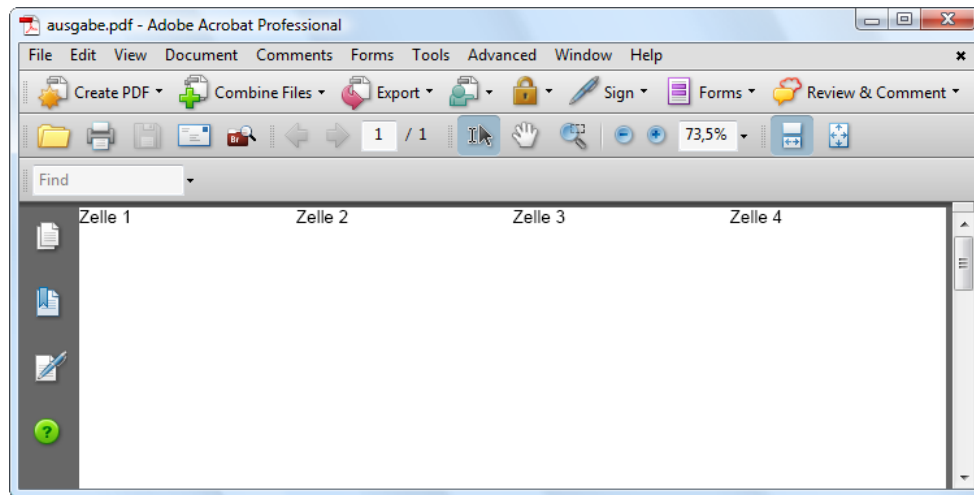


Abbildung 8.38 Eine sehr einfache Tabelle wurde angelegt.

Es handelt sich hier um eine sehr einfache Tabelle. Tabellendefinitionen lassen sich aber durch zahlreiche Attribute erweitern. So kann man u.a. Rahmen, Abstände und Farben einsetzen.

Listing 8.63 Die Tabelle wurde erweitert.

```
<fo:block>
<fo:table width="160mm" border-style="ridge" border-width="5pt">
  <fo:table-body>
    <fo:table-row>

      <fo:table-cell width="40mm" border-style="solid"
border-width="2pt">
        <fo:block>Zelle 1</fo:block>
      </fo:table-cell>

      <fo:table-cell width="40mm" border-style="solid"
border-width="2pt">
        <fo:block>Zelle 2</fo:block>
      </fo:table-cell>

      <fo:table-cell width="40mm" border-style="solid"
border-width="2pt">
        <fo:block>Zelle 3</fo:block>
      </fo:table-cell>

      <fo:table-cell width="40mm" border-style="solid"
border-width="2pt">
        <fo:block>Zelle 4</fo:block>
      </fo:table-cell>
    </fo:table-row>
  </fo:table-body>
</fo:table>

<fo:table-cell border-style="solid" border-width="1pt">
  <fo:block>Zelle 5</fo:block>
</fo:table-cell>

<fo:table-cell border-style="solid" border-width="1pt">
  <fo:block>Zelle 6</fo:block>
</fo:table-cell>
</fo:block>
```

```

<fo:table-cell border-style="solid" border-width="1pt">
  <fo:block>Zelle 7</fo:block>
</fo:table-cell>

<fo:table-cell border-style="solid" border-width="1pt">
  <fo:block>Zelle 8</fo:block>
</fo:table-cell>

</fo:table-row>
</fo:table-body>
</fo:table>
</fo:block>

```

Wie das Ergebnis dieser Syntax aussieht, zeigt **Abbildung 8.39**.

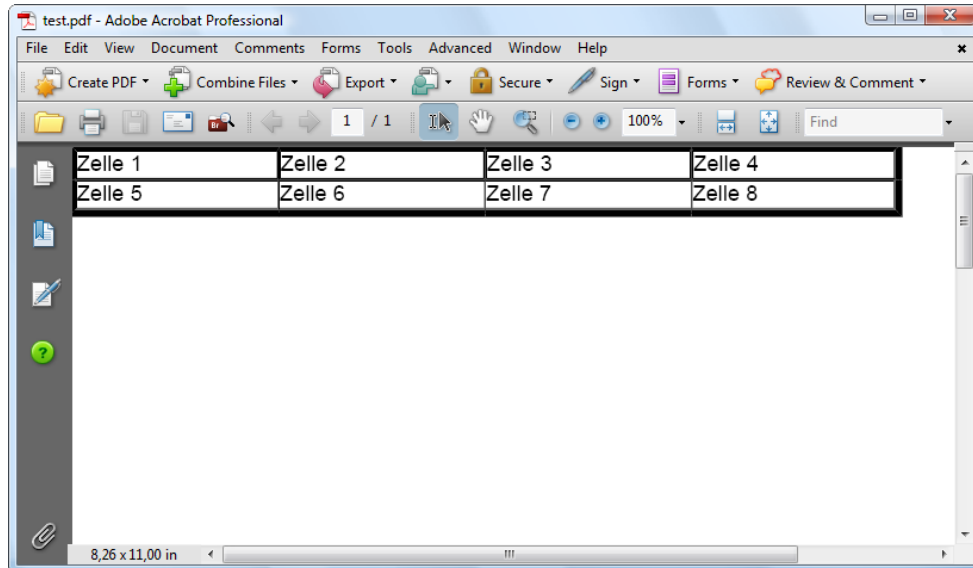


Abbildung 8.39 Das sieht schon eher nach einer Tabelle aus.

8.11.1 Zellen und Zeilen überspannen

In den bisherigen Beispielen wurde noch nicht die Möglichkeit des Überspannens von Spalten und Zeilen genutzt.

Mit dem Attribut `number-columns-spanned` bei `fo:table-column` und bei `fo:table-cell` wird das Überspannen von Spalten realisiert. Will man hingegen Zeilen überspannen, verwendet man das Attribut `number-rows-spanned` bei `fo:table-cell`.

Das folgende Beispiel zeigt, wie sich diese Elemente einsetzen lassen.

Listing 8.64 Eine etwas aufwendigere Tabelle

```

<fo:block>
  <fo:table width="110mm" border-style="outset" border-width="1pt">
    <fo:table-column column-number="1" column-width="25%"
      border-style="solid" border-width="1pt"/>
    <fo:table-column column-number="2" column-width="25%"
      border-style="solid" border-width="1pt"/>
    <fo:table-column column-number="3" column-width="25%"

```

```
border-style="solid" border-width="1pt"/>
<fo:table-column column-number="4" column-width="25%"
border-style="solid" border-width="1pt"/>

<fo:table-header>
  <fo:table-row>

    <fo:table-cell number-columns-spanned="4">
      <fo:block>Kopfzeile</fo:block>
    </fo:table-cell>

  </fo:table-row>
</fo:table-header>

<fo:table-footer>
  <fo:table-row>

    <fo:table-cell number-columns-spanned="4">
      <fo:block> Fusszeile</fo:block>
    </fo:table-cell>

  </fo:table-row>
</fo:table-footer>

<fo:table-body>
  <fo:table-row>

    <fo:table-cell column-number="1">
      <fo:block>1. Zelle</fo:block>
    </fo:table-cell>

    <fo:table-cell column-number="2">
      <fo:block>2. Zelle</fo:block>
    </fo:table-cell>

    <fo:table-cell column-number="3" number-columns-spanned="2">
      <fo:block>3. Zelle</fo:block>
    </fo:table-cell>

  </fo:table-row>

  <fo:table-row>

    <fo:table-cell column-number="1" number-columns-spanned="4">
      <fo:block>Mittelteil</fo:block>
    </fo:table-cell>

  </fo:table-row>

  <fo:table-row>

    <fo:table-cell column-number="1" number-columns-spanned="2">
      <fo:block>1. Zelle</fo:block>
    </fo:table-cell>

    <fo:table-cell column-number="3">
      <fo:block>2. Zelle</fo:block>
    </fo:table-cell>

    <fo:table-cell column-number="4">
      <fo:block>3. Zelle</fo:block>
    </fo:table-cell>

  </fo:table-row>
</fo:table-body>
</fo:table>
</fo:block>
```

Und auch hier natürlich wieder das Ergebnisdokument.