



Meine erste Website

Sehen und Können

CHRISTIAN WENZ TOBIAS HAUSER



Meine erste Website





Meine erste Website

Sehen und Können

CHRISTIAN WENZ TOBIAS HAUSER



Markt+Technik

Bibliografische Information der Deutschen Nationalbibliothek
Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der
Deutschen Nationalbibliografie; detaillierte bibliografische Daten
sind im Internet über <http://dnb.d-nb.de> abrufbar.

Die Informationen in diesem Produkt werden ohne Rücksicht auf einen
eventuellen Patentschutz veröffentlicht.
Warennamen werden ohne Gewährleistung der freien Verwendbarkeit benutzt.
Bei der Zusammenstellung von Text und Abbildungen wurde mit größter
Sorgfalt vorgegangen.
Trotzdem können Fehler nicht vollständig ausgeschlossen werden.
Verlag, Herausgeber und Autoren können für fehlerhafte Angaben
und deren Folgen weder eine juristische Verantwortung noch
irgendeine Haftung übernehmen.
Für Verbesserungsvorschläge und Hinweise auf Fehler sind Verlag und
Herausgeber dankbar.

Alle Rechte vorbehalten, auch die der fotomechanischen Wiedergabe und der
Speicherung in elektronischen Medien.
Die gewerbliche Nutzung der in diesem Produkt gezeigten Modelle und Arbeiten
ist nicht zulässig.

Fast alle Hardware- und Softwarebezeichnungen und weitere Stichworte und sonstige
Angaben, die in diesem Buch verwendet werden, sind als eingetragene Marken geschützt.
Da es nicht möglich ist, in allen Fällen zeitnah zu ermitteln, ob ein Markenschutz besteht,
wird das Symbol ® in diesem Buch nicht verwendet.

10 9 8 7 6 5 4 3 2 1

14 13 12

ISBN 978-3-8272-4727-8

© 2012 by Markt+Technik Verlag,
ein Imprint der Pearson Deutschland GmbH,
Martin-Kollar-Straße 10–12, D-81829 München/Germany
Alle Rechte vorbehalten
Umschlaggestaltung: Marco Lindenbeck, webwo GmbH, mlindenbeck@webwo.de
Lektorat: Birgit Ellissen, bellissen@pearson.de
Korrektorat: Marita Böhm, München
Herstellung: Martha Kürzl-Harrison, mkuerzl@pearson.de
Satz: text&form GbR, Fürstenfeldbruck
Druck und Verarbeitung: Firmengruppe APPL, aprinta druck, Wemding
Printed in Germany

Liebe Leserin, lieber Leser,

heutzutage reicht es fast nicht mehr aus, „nur“ online zu sein – eine eigene Website gehört fast schon zum „guten Ton“. Unser Buch zeigt Ihnen Bild für Bild, wie Sie Ihre ersten Schritte in Richtung eigener Webauftritt unternehmen können. Wir zeigen Ihnen zunächst einen Homepage-Baukasten und führen Sie dann Schritt für Schritt in die Technologien ein, mit denen Sie selbst eine Website von Grund auf erstellen, sie mit Text und Bildern füllen und am Ende ins Netz stellen und bei Suchmaschinen anmelden.

Wir sind uns sicher, dass Sie in null Komma nichts erste Ergebnisse erzielen und so Schritt für Schritt Ihre eigene Website aufbauen werden. Viel Vergnügen dabei!

Ihre Autoren

Christian Wenz und Tobias Hauser



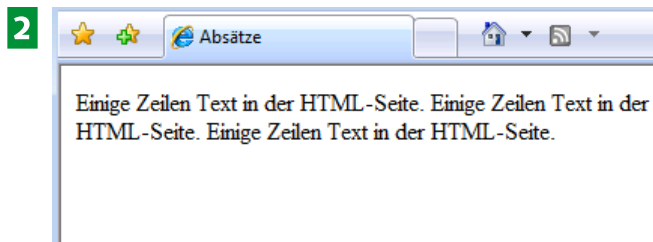
Text erfassen und formatieren

Start

1

```
<html>
<head>
  <title>Abs&auml;tze</title>
</head>
<body>
  Einige Zeilen Text in der HTML-Seite.
  Einige Zeilen Text in der HTML-Seite.
  Einige Zeilen Text in der HTML-Seite.
</body>
</html>
```

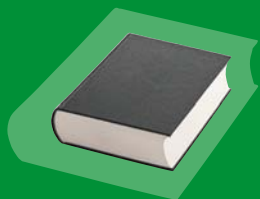
Die kryptische Zeichenfolge steht für „ä“.



3

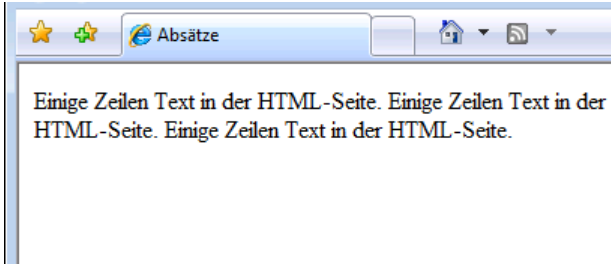
```
<body>
  <p>Einige Zeilen Text in der HTML-Seite.
  Einige Zeilen Text in der HTML-Seite.
  Einige Zeilen Text in der HTML-Seite.</p>
</body>
```

- 1 Tippen Sie ein paar Zeilen Text in das `<body>`-Tag des Browsers ein.
- 2 Testen Sie dies im Browser.
- 3 Schließen Sie nun den Text in ein `<p>`-Tag ein.



Korrekterweise sollte Text in HTML nur innerhalb von Blockelementen stehen. Eine der wichtigsten Arten von Blockelementen sind Absätze. Sie trennen Textteile voneinander ab. Innerhalb von Absätzen können Sie außerdem Zeilenumbrüche einfügen.

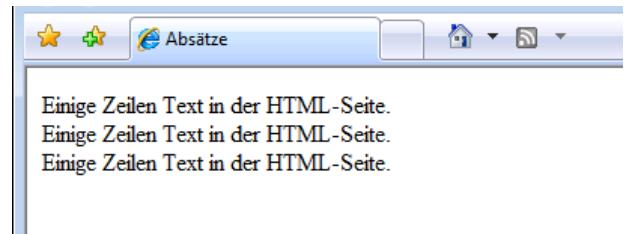
4



5

```
<body>
  <p>Einige Zeilen Text in der HTML-Seite.<br />
  Einige Zeilen Text in der HTML-Seite.<br />
  Einige Zeilen Text in der HTML-Seite.</p>
</body>
```

6



- 4 Testen Sie erneut.
- 5 Brechen Sie einzelne Textpassagen innerhalb des Absatzes mit `<br />`-Tags um.
- 6 Prüfen Sie auch hier, wie der Browser Umbrüche darstellt.

Ende

Absätze haben standardmäßig einen Abstand nach oben und einen Abstand nach unten. Sie lassen sich über den CSS-Befehl *margin* steuern (siehe Kapitel 6).

Blockelement bezeichnet ein HTML-Element, das einen Zeilenumbruch nach sich zieht. Dazu zählen Absätze und Überschriften.

Das `p` vom `<p>`-Tag steht für paragraph, sprich Absatz. `<br>` steht für break, also Zeilenumbruch.

TIPP
FACHWORT
HINWEIS

Start

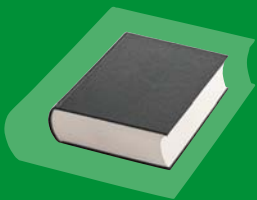
Das ist ein „Ü“.

```
1 <html>
  <head>
    <title>&Uuml;berschriften</title>
  </head>
  <body>
    <h1>&Uuml;berschrift 1</h1>
  </body>
</html>
```

2 **Überschrift 1**

```
3 <body>
  <h1>&Uuml;berschrift 1</h1>
  <h2>&Uuml;berschrift 2</h2>
</body>
```

- 1 Legen Sie eine Überschrift erster Ordnung an.
- 2 Prüfen Sie das Ergebnis im Browser.
- 3 Ergänzen Sie eine Überschrift der zweiten Ebene.



Überschriften dienen dazu, eine Webseite zu strukturieren. HTML besitzt sechs Überschriftenebenen, von `<h1>` bis `<h6>`. Eine Überschrift verhält sich wie ein Absatz, allerdings besitzt sie bereits eine vorgegebene Schriftformatierung.

4

Überschrift 1

Überschrift 2

Behalten Sie bei Überschriften nach Möglichkeit die logische Rangfolge bei – Überschrift zwei sollte unterhalb von Überschrift eins stehen.

5

```
<body>
  <h1>&Uuml;berschrift 1</h1>
  <h2>&Uuml;berschrift 2</h2>
  <h3>&Uuml;berschrift 3</h3>
</body>
```

6

Überschrift 1

Überschrift 2

Überschrift 3

- 4 Auch hier werfen Sie einen Blick in den Browser.
- 5 Nun folgt noch eine Überschrift der dritten Ebene.
- 6 Und auch das testen Sie gleich.

Ende

Die Standardformatierung von Überschriften ändern Sie per Style Sheets problemlos.

Barrierefreiheit bedeutet, dass auch Nutzer mit Behinderungen eine Website verwenden können.

Überschriften der ersten und zweiten Ordnung sind sowohl für die Barrierefreiheit als auch für die Suchmaschinenoptimierung wichtig. Achten Sie also darauf, entsprechende Überschriften mit aussagekräftigem Text einzusetzen.

TIPP

FACHWORT

TIPP

Start

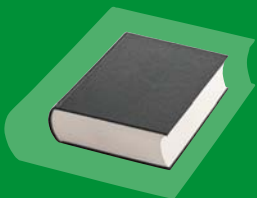
```
1 <html>
  <head>
    <title>Text auszeichnen</title>
  </head>
  <body>
    <p>
      Text in einem Absatz mit fetter und kursiver
      Hervorhebung. Wichtig ist auch 23.
    </p>
  </body>
</html>
```

Auch hier gilt: Die
Einrückung dient
der Lesbarkeit.

```
2 <body>
  <p>
    Text in einem Absatz mit <b>fetter</b> und kursiver
    Hervorhebung. Wichtig ist auch 23.
  </p>
</body>
```

```
3 <body>
  <p>
    Text in einem Absatz mit <b>fetter</b> und
    <i>kursiver</i> Hervorhebung. Wichtig ist auch 23.
  </p>
</body>
```

- 1 Schreiben Sie einen Beispieltext in einen Absatz.
- 2 Heben Sie ein Wort mit einem `<b>`-Tag fett hervor.
- 3 Heben Sie ein Wort mit einem `<i>`-Tag kursiv hervor.



HTML bietet einige Tags, um Elemente innerhalb einer HTML-Seite auszuzeichnen. Diese Tags werden einfach um den auszuzeichnenden Bereich gelegt.

4

```
<body>
  <p>
    Text in einem Absatz mit <b>fetter</b> und
    <i>kursiver</i> Hervorhebung. <strong>Wichtig
    </strong> ist auch 23.
  </p>
</body>
```

5

```
<body>
  <p>
    Text in einem Absatz mit <b>fetter</b> und
    <i>kursiver</i> Hervorhebung. <strong>Wichtig
    </strong> ist auch 2<sup>3</sup>.
  </p>
</body>
```

Sehen Sie hier einen Fehler?
Das Tag müsste </sup> heißen.
Der Browser beschwert sich
jedoch nicht, sondern inter-
pretiert, so gut er kann.

6

Text in einem Absatz mit **fetter** und *kursiver*
Hervorhebung. **Wichtig** ist auch 2³

4 Heben Sie ein Wort mit einem -Tag hervor.

5 Stellen Sie eine Zahl mit dem <sup>-Tag hoch.

6 Sehen Sie sich das Ergebnis im Browser an.

Ende

Das Auszeichnungstag besitzt selbst keine Formatierungseigenschaften. Die Formatierung wird hier komplett über CSS vorgenommen.

Für die Tags <sub> und <sup> zum Hoch- und Tiefstellen gibt es keine direkte CSS-Entsprechung.

Vorsicht, Tags dürfen nicht ineinander verschachtelt werden: fett und <i>auch kursiv</i> ist also falsch!

TIPP

HINWEIS

TIPP

Start

```

1 <html>
  <head>
    <title>Sonderzeichen</title>
  </head>
  <body>
    <p>
      ÄäÖöÜü €
    </p>
  </body>
</html>

```

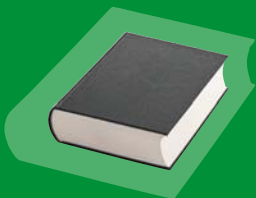
2 ÄäÖöÜü €

```

3 <p>
  &Auml;&auml;&Ouml;&ouml;&Uuml;&uuml; &euro;
</p>

```

- 1 Tippen Sie Text mit Umlauten und einem Eurozeichen ein.
- 2 Rufen Sie die Seite im Browser auf. In den meisten Browsern werden auch ohne Zeichensatz die richtigen Zeichen angezeigt.
- 3 Fügen Sie nun statt der Umlaute die entsprechenden Sonderzeichen ein: `ü` für ein ü, `ä` für ein ä und `€` für ein Eurozeichen.



Zu den Sonderzeichen gehören die deutschen Umlaute, aber auch beispielsweise das Eurozeichen. Ein Sonderzeichen beginnt immer mit dem Et-Zeichen (&). Dann folgt der Name, anschließend ein Strichpunkt. Je nach verwendetem Zeichensatz benötigen Sie die Sonderzeichen, aber nicht unbedingt. In UTF-8 sind alle relevanten Sonderzeichen enthalten.

4

ÄäÖöÜü €
ÄäÖöÜü €

Der Umbruch hier erfolgt aus drucktechnischen Gründen; Sie können das ganze Tag in eine Zeile schreiben.

5

```
<html>
<head>
  <title>Sonderzeichen</title>
  <meta http-equiv="Content-Type" content="text/html;
    charset=utf-8" />
</head>
<body>
  <p>
    ÄäÖöÜü €
  </p>
  <p>
    &Auml; &auml; &Ouml; &ouml; &Uuml; &uuml; &euro;
  </p>
</body>
</html>
```

6

ÄäÖöÜü €
ÄäÖöÜü €

- 4 Testen Sie das Ergebnis erneut im Browser.
- 5 Nun ändern Sie im Kopf der HTML-Datei den Zeichensatz auf UTF-8. Eventuell müssen Sie den Zeichensatz im Editor ändern.
- 6 Nun können Sie prüfen, dass sowohl Sonderzeichen als auch normale Umlaute und Eurosymbol korrekt angezeigt werden.

Ende

Für das Web existieren zwei Zeichensätze: ISO-8859-1 mit 256 Zeichen inklusive der deutschen Umlaute und UTF-8 mit wesentlich mehr Zeichen.

Sie müssen ein Dokument im Editor als UTF-8 speichern, wenn Sie UTF-8 als Zeichensatz verwenden möchten.

Zeichensatz ist eine Menge an Zeichen, die der Browser darstellt. Browser kennen verschiedene Arten von Zeichensätzen. Normalerweise verwendet der Browser den in der HTML-Datei angegebenen Zeichensatz.

Start

```

1 <html>
  <head>
    <title>Style Sheets einsetzen</title>
  </head>
  <body>
    <h1>
      &Uuml;berschrift 1
    </h1>
    <p>
      Ein erster Absatz.
    </p>
    <p>
      Ein zweiter Absatz.
    </p>
  </body>
</html>

```

```

2 <h1 style="">
  &Uuml;berschrift 1
</h1>
<p style="">
  Ein erster Absatz.
</p>
<p style="">
  Ein zweiter Absatz.
</p>

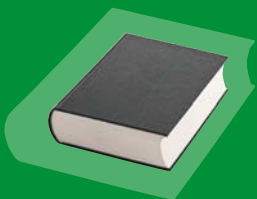
```

```

3 <h1 style="color:">
  &Uuml;berschrift 1
</h1>
<p style="color:">
  Ein erster Absatz.
</p>
<p style="color:">
  Ein zweiter Absatz.
</p>

```

- 1 Erstellen Sie ein Dokument mit einer Überschrift und zwei Absätzen.
- 2 Weisen Sie allen drei ein *style*-Attribut zu.
- 3 Schreiben Sie nun den CSS-Befehl *color* in jedes der *style*-Attribute.



Style Sheets, die mit der Technologie CSS geschrieben werden, sind Stilanweisungen im HTML-Code, die HTML-Elemente formatieren. Um einem Tag direkt einen Stil zuzuweisen, verwenden Sie einen sogenannten Inline-Style. Hierzu stehen die Stilangaben im *style*-Attribut des jeweiligen Tags.

```
4 <h1 style="color:red">
    &Uuml;berschrift 1
</h1>
<p style="color:green">
    Ein erster Absatz.
</p>
<p style="color:blue">
    Ein zweiter Absatz.
</p>
```

```
5 <h1 style="color:red">
    &Uuml;berschrift 1
</h1>
<p style="color:green; font-family:Arial">
    Ein erster Absatz.
</p>
<p style="color:blue">
    Ein zweiter Absatz.
</p>
```

Das Leerzeichen ist nicht notwendig, fördert aber die Lesbarkeit.

6

Überschrift 1

Ein erster Absatz.

Ein zweiter Absatz.

- 4 Vergeben Sie für jeden *color*-Befehl nach einem Doppelpunkt eine Farbe mit englischem Namen: *red* für Rot, *green* für Grün und *blue* für Blau.
- 5 Im zweiten Absatz ergänzen Sie hinter dem *color*-Befehl durch einen Strichpunkt getrennt den Befehl *font-family* mit dem Wert *Arial*.
- 6 Testen Sie das Ergebnis im Browser.

Ende

In Inline-Stilen können Sie beliebig viele CSS-Angaben durch Strichpunkte getrennt aufreihen.

CSS steht für Cascading Style Sheets. Das sind Stilanweisungen, die aufeinander aufbauen (abgeleitet von kaskadieren).

Das *style*-Attribut können Sie für das *<body>*-Tag und alle anderen sichtbaren HTML-Elemente vergeben.

Start

```

1 <html>
  <head>
    <title>Style Sheets im Kopf der Seite</title>
  </head>
  <body>
    <h1>
      &Uuml;berschrift 1
    </h1>
    <p>
      Ein erster Absatz.
    </p>
    <p>
      Ein zweiter Absatz.
    </p>
  </body>
</html>

```

```

2 <head>
  <title>Style Sheets im Kopf der Seite</title>
  <style type="text/css">

  </style>
</head>

```

```

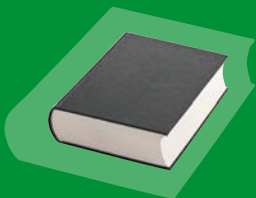
3 <style type="text/css">
  h1 {

  }
</style>

```

Wie bei Tags gilt auch bei den geschweiften Klammern: Wenn eine aufgeht, muss später eine zugehen.

- 1 Starten Sie wieder mit einer Überschrift und zwei Absätzen.
- 2 Erstellen Sie im Kopf der HTML-Seite einen `<style>`-Block.
- 3 Fügen Sie den Tag-Selektor `h1` für Überschriften in den Block ein.



Inline-Stile im `style`-Attribut haben den Nachteil, dass man sie für jedes Tag separat vergeben muss. Sie können stattdessen auch im Kopf der HTML-Seite zentral Stile vergeben. Diese schreiben Sie in einen `<style>`-Block. Einzelne Tags formatieren Sie mit sogenannten Tag-Selektoren.

```
4 <style type="text/css">
    h1 {
        color: red;
        font-family: Arial;
    }
</style>
```

```
5 <style type="text/css">
    h1 {
        color: red;
        font-family: Arial;
    }
    p {
        color: blue;
        font-family: Arial;
    }
</style>
```

6 Überschrift 1

Ein erster Absatz.

Ein zweiter Absatz.

- 4 Schreiben Sie in geschweiften Klammern einige Stilanweisungen.
- 5 Ergänzen Sie Stilanweisungen für Absätze im `<style>`-Block.
- 6 Testen Sie Ihre Basisformatierung.

Ende

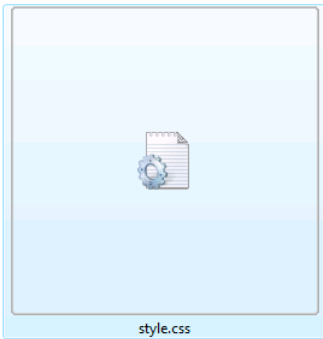
Erledigen Sie mit Tag-Selektoren grundlegende Formatierungen, die für alle Tags auf der Seite gelten. Ausnahmen können Sie dann in Inline-Stilen oder mit anderen Selektoren definieren.

Selektor heißt der Befehl, mit dem Sie angeben, für welche Elemente ein Stil im Kopf der HTML-Seite gilt. Ein Tag-Selektor vergibt beispielsweise Stilanweisungen für eine bestimmte HTML-Elementart wie Absätze oder Überschriften.

Gibt es einen CSS-Befehl sowohl im Kopf der HTML-Seite als auch im Inline-Stil, so gewinnt der Inline-Stil.

Start

1



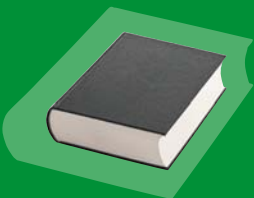
2

```
h1 {
  color: red;
  font-family: Arial;
}
p {
  color: blue;
  font-family: Arial;
}
```

3

```
<html>
<head>
  <title>Externe Style Sheets</title>
</head>
<body>
  <h1>
    &Uuml;berschrift 1
  </h1>
  <p>
    Ein erster Absatz.
  </p>
  <p>
    Ein zweiter Absatz.
  </p>
</body>
</html>
```

- 1 Erstellen Sie eine neue CSS-Datei mit der Dateiendung .css und speichern Sie sie in Ihrem Übungsverzeichnis.
- 2 Fügen Sie in die CSS-Datei Formatierungen für eine Überschrift erster Ebene und für Absätze ein.
- 3 Erstellen oder öffnen Sie eine HTML-Datei mit einer Überschrift erster Ordnung und einigen Absätzen. Speichern Sie sie im Übungsverzeichnis.



Ein CSS im Kopf einer HTML-Seite gilt immer nur für das aktuelle Dokument. Wollen Sie Stile zentral verwalten, legen Sie sie in einer externen CSS-Datei ab. Dabei handelt es sich um eine normale Textdatei mit der Dateiendung .css. Die Datei enthält nur die Selektoren und Stilanweisungen.

```
4 <head>
  <title>Externe Style Sheets</title>
  <link type="text/css" />
</head>
```

```
5 <head>
  <title>Externe Style Sheets</title>
  <link href="style.css" rel="stylesheet"
    type="text/css" />
</head>
```

Die Dateiendung .css hat sich so eingebürgert, ist aber eigentlich keine Pflicht.

6 **Überschrift 1**

Ein erster Absatz.

Ein zweiter Absatz.

- 4 Fügen Sie in den Kopf der HTML-Seite ein `<link>`-Tag ein.
- 5 Vergeben Sie als Adresse im Attribut *href* den Namen der CSS-Datei. Sollte sie in einem anderen Verzeichnis liegen, müssen Sie sie korrekt adressieren.
- 6 Testen Sie das Ergebnis im Browser: Sie sehen dasselbe Ergebnis wie beim Style Sheet im Kopf der Seite – nur zentral gesteuert.

Ende

Lagern Sie in einem externen CSS die Stilanweisungen aus, die Sie auf allen Webseiten Ihrer Website benötigen.

Website steht für einen Webauftritt, sprich eine Sammlung von Webseiten. Externe CSS können als zentrale Style Sheets dazu dienen, den Auftritt einheitlich zu gestalten und zu formatieren.

Neben dem Verlinken können Sie ein externes CSS auch importieren. Dazu schreiben Sie `@import "style.css";` in einen `<style>`-Block.

Start

```

1 <html>
  <head>
    <title>CSS-Klassen</title>
    <style type="text/css">

    </style>
  </head>
  <body>
    <h1>
      &Uuml;berschrift 1
    </h1>
    <p>
      Ein erster Absatz.
    </p>
    <p>
      Ein zweiter Absatz.
    </p>
  </body>
</html>

```

```

2 <body>
  <h1 class="format1">
    &Uuml;berschrift 1
  </h1>
  <p class="format1">
    Ein erster Absatz.
  </p>
  <p>
    Ein zweiter Absatz.
  </p>
</body>

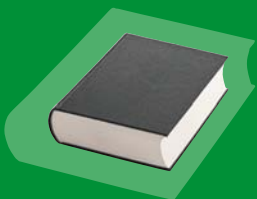
```

```

3 <body>
  <h1 class="format1">
    &Uuml;berschrift 1
  </h1>
  <p class="format1">
    Ein erster Absatz.
  </p>
  <p class="format2">
    Ein zweiter Absatz.
  </p>
</body>

```

- 1 Ausgangsbasis ist eine HTML-Seite mit einer Überschrift und zwei Absätzen.
- 2 Fügen Sie der Überschrift und einem Absatz mit dem Attribut *class* eine Klasse mit dem Namen *format1* hinzu.
- 3 Vergeben Sie für den zweiten Absatz eine Klasse *format2*.



CSS-Klassen erlauben die Formatierung ganz bestimmter HTML-Elemente. In diesen Elementen müssen Sie mit dem HTML-Attribut *class* angeben, dass eine CSS-Klasse zum Einsatz kommt. Klassen sind wegen ihrer flexiblen Einsetzbarkeit eine der wichtigsten CSS-Funktionen.

```
4 <style type="text/css">
    .format1 {
        color: red;
        font-family: Arial;
    }
    .format2 {
        color: green;
        font-family: Verdana;
    }
</style>
```

```
5 <style type="text/css">
    .format1 {
        color: red;
        font-family: Arial;
    }
    .format2 {
        color: green;
        font-family: Verdana;
    }
    h1.format1 {
        color: blue;
    }
</style>
```

Diese Anweisung bezieht sich auf alle <h1>-Elemente mit der Klasse „format1“.

6 Überschrift 1

Ein erster Absatz.

Ein zweiter Absatz.

- 4 Legen Sie für beide Klassen im Kopf der HTML-Seite einige CSS-Befehle an. Die Klassen werden dabei mit Punkt geschrieben.
- 5 Schreiben Sie nun noch einen CSS-Befehl speziell für die Klasse *format1* bei <h1>-Überschriften. Dafür steht der Tag-Name vor dem Klassen-Selektor.
- 6 Im Browser sehen Sie, dass der CSS-Befehl speziell für Überschriften die allgemeine Angabe für die Klasse überschreibt.

Ende

Eine häufige Fehlerquelle: Im HTML-Attribut wird die Klasse ohne Punkt geschrieben, im Style Sheet selbst mit Punkt.

Je spezieller eine Stilangabe ist, desto eher kommt sie zum Zug. Oder um es fachlich zu sagen: umso höher ist die Präferenz. Das heißt beispielsweise, eine Klasse mit Tag-Selektor ist wichtiger als eine Klasse ohne.

Schreiben Sie Klassen durch Leerzeichen getrennt in das *class*-Attribut, um einem HTML-Element die Formatierungen aus mehreren Klassen zuzuweisen.

Start

```
1 <html>
  <head>
    <title>Verschachtelte Tags</title>
    <style type="text/css">

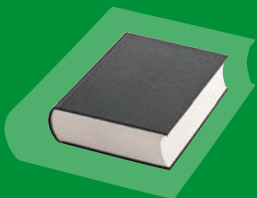
      </style>
  </head>
  <body>
    <p>
      Ein Absatz mit <b>fetter</b> Hervorhebung.
    </p>
  </body>
</html>
```

Salopp gesprochen: alle `<b>`-Elemente, die sich innerhalb von `<p>`-Elementen befinden.

```
2 <style type="text/css">
  p b {
    color: red;
  }
</style>
```

3 Ein Absatz mit **fetter** Hervorhebung.

- 1 Erstellen Sie einen Absatz mit darin enthaltenem `<b>`-Tag.
- 2 Der Zugriff erfolgt über den `p`-Tag-Selektor und durch Leerzeichen getrennt den `b`-Selektor.
- 3 Überprüfen Sie das Ergebnis im Browser.



Eine Verschachtelung von Tags können Sie auch mit den Tag-Selektoren in CSS nachbauen. So steuern Sie sehr exakt, welche untergeordneten Elemente formatiert werden. Wollen Sie mehrere Tags mit den gleichen Formatierungen versehen, schreiben Sie die Tag-Namen durch Komma getrennt hintereinander.

```
4 <body>
  <h1>&Uuml;berschrift 1</h1>
  <p>
    Ein Absatz mit <b>fetter</b> Hervorhebung.
  </p>
</body>
```

```
5 <style type="text/css">
  p b {
    color: red;
  }
  h1, p {
    color: green;
  }
</style>
```

6

Überschrift 1

Ein Absatz mit **fetter** Hervorhebung.

- 4 Erstellen Sie zusätzlich eine Überschrift `<h1>`.
- 5 Um Überschrift und Absatz mit denselben Befehlen zu formatieren, schreiben Sie einen Selektor, der die beiden Tag-Namen durch ein Komma trennt.
- 6 Testen Sie auch dieses Ergebnis im Browser.

Ende

Stilangaben für mehrere Tags und der Zugriff auf verschachtelte Tags lassen sich auch beliebig mischen. So formatieren Sie noch effizienter.

Absätze und Überschriften dürfen Sie nicht ineinander verschachteln, wenn Sie korrektes HTML schreiben wollen.

Start

```
1 <html>
  <head>
    <title>IDs</title>
    <style type="text/css">

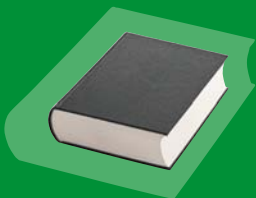
    </style>
  </head>
  <body>
    <p>
      Ein Absatz mit Text.
    </p>
  </body>
</html>
```

```
2 <body>
  <p id="absatz1">
    Ein Absatz mit Text.
  </p>
</body>
```

```
3 <style type="text/css">
  #absatz1 {

  }
</style>
```

- 1 Ausgangsbasis ist ein Absatz in einem leeren HTML-Dokument.
- 2 Vergeben Sie mit dem Attribut *id* die ID *absatz1*.
- 3 Greifen Sie im CSS auf die ID zu. Schreiben Sie dazu einen Hash und dann den Namen der ID.



Eine ID vergeben Sie mit dem Attribut *id*. Per CSS greifen Sie dann darauf zu und formatieren das Element über den ID-Selektor. Im Selektor steht vor dem Namen der ID ein Doppelkreuz (#, auch Hash genannt).

```
4 <style type="text/css">
  #absatz1 {
    color: red;
    font-family: Arial;
  }
</style>
```

```
5 <style type="text/css">
  #absatz1 {
    color: red;
    font-family: Arial;
  }
  p#absatz1 {
    color: green;
  }
</style>
```

Alle <p>-Elemente mit der ID „absatz1“ – wobei es wohl meist nur ein Element mit einer ID gibt.

6 Ein Absatz mit Text.

- 4 Legen Sie einige Stilanweisungen in geschweiften Klammern an.
- 5 Fügen Sie als zweite Stilanweisung dieselbe ID hinzu, schreiben aber vor das Doppelkreuz den Tag-Selektor. Überschreiben Sie eine Stilanweisung.
- 6 Testen Sie die Wirkung der zwei ID-basierten Formatierungen im Browser.

Ende

Verwenden Sie eine ID immer nur einmal in einem Dokument. HTML lässt Ihnen zwar auch eine Mehrfachverwendung durchgehen, aber beispielsweise bei Skripten führt das schon zu Problemen.

ID steht für Identifier. Dabei handelt es sich um einen eindeutigen Identifikator für ein Element.

Eine Formatierung in einer ID hat immer den Vorzug gegenüber der gleichen Formatierung in einer Klasse.

Start

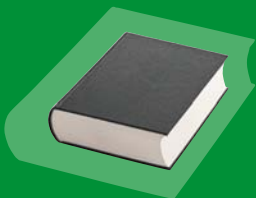
```
1 <html>
  <head>
    <title>Schriftart</title>
  </head>
  <body>
    <p style="">
      Ein Absatz mit Text.
    </p>
  </body>
</html>
```

```
2 <body>
  <p style="font-family:Arial">
    Ein Absatz mit Text.
  </p>
</body>
```

```
3 <body>
  <p style="font-family:Arial, Helvetica,
    sans-serif">
    Ein Absatz mit Text.
  </p>
</body>
```

Der Browser sucht die
Schriften nach der Reihen-
folge ihres Auftretens.

- 1 Erstellen Sie einen Absatz mit Inline-Stil.
- 2 Mit dem CSS-Befehl *font-family* geben Sie die Schriftart an.
- 3 Geben Sie die Schriftarten, die verwendet werden sollen, nacheinander durch Kommata getrennt an. Die letzte Angabe ist die Schriftfamilie.



Die Schriften, die Sie in HTML einsetzen, müssen auf dem System des Nutzers installiert sein. Jede Schrift ist im Grunde eine Datei, die alle benötigten Zeichen enthält. Ein Betriebssystem wird mit Standardschriften ausgeliefert und kann um eigene Schriften erweitert werden. Ist eine Schrift nicht vorhanden, wird eine andere Schrift – meist eine Standardschrift – verwendet.

```
4 <body>
  <p style="font-family:Arial, Helvetica,
    sans-serif">
    Ein Absatz mit Text.
  </p>
  <p style="font-family:">
    Ein zweiter Absatz mit Text.
  </p>
</body>
```

```
5 <body>
  <p style="font-family:Arial, Helvetica,
    sans-serif">
    Ein Absatz mit Text.
  </p>
  <p style="font-family: 'Times New Roman', serif">
    Ein zweiter Absatz mit Text.
  </p>
</body>
```

```
6 Ein Absatz mit Text.
  Ein zweiter Absatz mit Text.
```

Achtung, verwenden Sie keine doppelten Anführungszeichen, das kollidiert mit denen nach „style“!

- 4 Erstellen Sie einen zweiten Absatz mit Schriftartangabe.
- 5 Vergeben Sie dem zweiten Absatz eine andere Schriftart. Ist der Schriftname länger als ein Wort, muss er in Anführungszeichen stehen.
- 6 Testen Sie die Schriften im Browser.

Ende

Verwenden Sie als letztes Sicherheitsnetz die Schriftfamilien wie sans-serif (serifenlos), serif (Serifenschriften) und monospace (Diktengleiche).

Serifen sind die Schmuckfortsätze an Buchstaben.

Als sichere Schriften, sprich auf allen Betriebssystemen verfügbar, gelten einige wenige Systemschriften wie Arial, Verdana, Courier und Times New Roman. Eine Übersicht liefert <http://www.ampsoft.net/webdesign-I/WindowsMacFonts.html>.

Start

```
1 <head>
  <title>Schriftfarbe</title>
</head>
<body>
  <p>
    Ein erster Absatz mit Text.
  </p>
  <p>
    Ein zweiter Absatz mit Text.
  </p>
  <p>
    Ein dritter Absatz mit Text.
  </p>
</body>
</html>
```

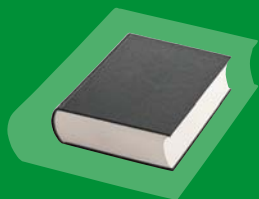
```
2 <p style="color: red">
  Ein erster Absatz mit Text.
</p>
<p>
  Ein zweiter Absatz mit Text.
</p>
<p>
  Ein dritter Absatz mit Text.
</p>
```

```
3 Ein erster Absatz mit Text.

Ein zweiter Absatz mit Text.

Ein dritter Absatz mit Text.
```

- 1 Ausgangsbasis ist eine HTML-Datei mit drei Absätzen.
- 2 Vergeben Sie für den ersten Absatz die Farbe als Farbname.
- 3 Sehen Sie sich den umgefärbten Absatz an.



Die Schriftfarbe ist die Vordergrundfarbe, die mit dem CSS-Befehl *color* bestimmt wird. Eine Farbe wird als RGB-Wert angegeben. Sie können diese Farbe auf drei Arten in HTML festsetzen: mit einem Farbnamen, den Werten als RGB oder als hexadezimaler Farbwert.

```
4 <p style="color: red">
    Ein erster Absatz mit Text.
</p>
<p style="color: #00ff00">
    Ein zweiter Absatz mit Text.
</p>
<p>
    Ein dritter Absatz mit Text.
</p>
```

```
5 <p style="color: red">
    Ein erster Absatz mit Text.
</p>
<p style="color: #00ff00">
    Ein zweiter Absatz mit Text.
</p>
<p style="color: rgb(0, 0, 255)">
    Ein dritter Absatz mit Text.
</p>
```

6

Ein erster Absatz mit Text.

Ein zweiter Absatz mit Text.

Ein dritter Absatz mit Text.

- 4 Der zweite Absatz erhält eine Farbe als hexadezimale Zahl. Die Farbe beginnt mit einem Doppelkreuz.
- 5 Der dritte Absatz bekommt eine Farbe mit den RGB-Werten. Dies erfolgt mit *rgb* (Rotwert, Grünwert, Blauwert).
- 6 Testen Sie nun die Gesamtwirkung mit drei umgefärbten Absätzen.

Ende

Der Monitor setzt jede Farbe aus drei Grundfarben, Rot, Grün und Blau, zusammen. Jede Grundfarbe kann von 0 (ganz dunkel) bis 255 (höchste Helligkeit) leuchten. Diese drei Helligkeitswerte ergeben dann eine Farbe.

Hexadezimale Farbwerte stellen die Helligkeitswerte in hexadezimaler Schreibweise dar. Die Basis sind die Zahlen von 0 bis 15. 10 bis 15 werden mit den Zeichen A bis F dargestellt. 255 entspricht in hexadezimaler Schreibweise FF.

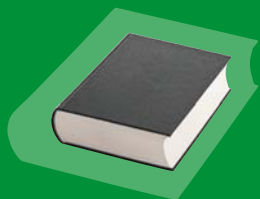
Start

```
1 <head>
  <title>Schriftgröße</title>
</head>
<body>
  <p>
    Ein erster Absatz mit Text.
  </p>
  <p>
    Ein zweiter Absatz mit Text.
  </p>
</body>
</html>
```

```
2 <body style="font-size: 0.8em">
  <p>
    Ein erster Absatz mit Text.
  </p>
  <p>
    Ein zweiter Absatz mit Text.
  </p>
</body>
```

```
3
Ein erster Absatz mit Text.
Ein zweiter Absatz mit Text.
```

- 1 Starten Sie mit einem HTML-Dokument mit zwei Absätzen.
- 2 Vergeben Sie im `<body>`-Tag die Schriftgröße 0.8 em, sprich 80 % von der Standardschriftgröße.
- 3 Testen Sie die Seite im Browser. Sie sehen die verkleinerte Standardschrift.



Die Schriftgröße wird mit dem CSS-Befehl *font-size* gesteuert. Sie geben die Schriftgröße in absoluten oder relativen Einheiten an. Absolute Einheiten sind Pixel und Punkt. Relative Einheiten sind Prozent und em.

```
4 <body style="font-size: 0.8em">
  <p style="font-size: 24px">
    Ein erster Absatz mit Text.
  </p>
  <p>
    Ein zweiter Absatz mit Text.
  </p>
</body>
```

```
5 <body style="font-size: 0.8em">
  <p style="font-size: 24px">
    Ein erster Absatz mit Text.
  </p>
  <p style="font-size: 0.8em">
    Ein zweiter Absatz mit Text.
  </p>
</body>
```

6

Ein erster Absatz mit Text.

Ein zweiter Absatz mit Text.

- 4 Ändern Sie die Schriftgröße im ersten Absatz auf 24 px, um die Schrift deutlich zu vergrößern.
- 5 Geben Sie im zweiten Absatz die Schriftgröße 0.8 em an. Die Angabe bezieht sich auf die übergeordnete Schriftgröße des `<body>`-Tags.
- 6 Testen Sie die Wirkung. Die Schrift im ersten Absatz wurde vergrößert, im zweiten Absatz hat sich die doppelte Verkleinerung stark ausgewirkt.

Ende

Die Standardschriftgröße für Fließtext beträgt in den meisten Browsern 16 Pixel.

em steht eigentlich für die Breite des Buchstabens M. 1 em entspricht 100 %.

Start

```

1 <head>
  <title>Schriftformate</title>
</head>
<body>
  <p>
    Ein erster Absatz mit Text.
  </p>
  <p>
    Ein zweiter Absatz mit Text.
  </p>
  <p>
    Ein dritter Absatz mit Text.
  </p>
  <p>
    Ein vierter Absatz mit Text.
  </p>
</body>
</html>

```

```

2 <p style="font-weight: bold">
  Ein erster Absatz mit Text.
</p>

```

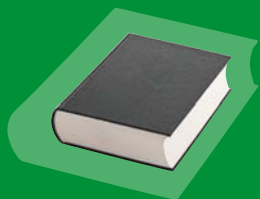
Der Begriff „weight“ steht eigentlich für „Gewicht“.

```

3 <p style="font-style: italic">
  Ein zweiter Absatz mit Text.
</p>

```

- 1 Starten Sie mit einer HTML-Seite mit vier Absätzen.
- 2 Im ersten Absatz verwenden Sie den CSS-Befehl *font-weight*, um den Absatz fett zu formatieren.
- 3 Der zweite Absatz erhält mit dem CSS-Befehl *font-style* und dem Wert *italic* eine kursive Formatierung.



Die Schriftformatierung dient zur Hervorhebung von Textpassagen. CSS bietet hier – vor allem im Vergleich zu alten HTML-Zeiten – hervorragende Möglichkeiten.

```
4 <p style="text-transform: uppercase">  
    Ein dritter Absatz mit Text.  
</p>
```

```
5 <p style="font-variant: small-caps">  
    Ein vierter Absatz mit Text.  
</p>
```

6

Ein erster Absatz mit Text.

Ein zweiter Absatz mit Text.

EIN DRITTER ABSATZ MIT TEXT.

EIN Vierter ABSATZ MIT TEXT.

- 4** Im dritten Absatz wandeln Sie per *text-transform* mit dem Wert *uppercase* alle Buchstaben in Großbuchstaben um.
- 5** Der vierte Absatz wird mit *font-variant* und dem Wert *small-caps* in Kapitälchen umgewandelt.
- 6** Testen Sie die verschiedenen Schriftformatierungen im Browser.

Ende

Mit dem Tag `<span>` können Sie die CSS-Auszeichnungen innerhalb eines Textes verwenden. `<span>` besitzt selbst keine Formatierungsfunktion.

Kapitälchen sind in der Typografie leicht verkleinerte und angepasste Großbuchstaben.

Verwenden Sie bei der Umwandlung in Großbuchstaben oder auch in Kleinbuchstaben (*lowercase*) den CSS-Befehl, statt die Buchstaben einzutippen. Dann lässt sich die Formatierung auch nachträglich ändern.

Start

```

1 <head>
  <title>Einrückungen</title>
</head>
<body>
  <p>
    Der Anfang des Absatzes. Ein Absatz mit Text
    und mehreren Zeilen. Ein Absatz mit Text und
    mehreren Zeilen. Ein Absatz mit Text und
    mehreren Zeilen. Ein Absatz mit Text und
    mehreren Zeilen.
  </p>
</body>
</html>

```

```

2 <head>
  <title>Einrückungen</title>
  <style type="text/css">

  </style>
</head>

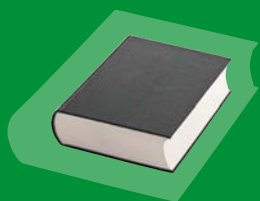
```

```

3 <style type="text/css">
  p {
    text-indent: 10px;
  }
</style>

```

- 1 Ausgangspunkt ist ein Absatz in einer HTML-Seite.
- 2 Erstellen Sie einen `<style>`-Block im Kopf der HTML-Seite.
- 3 Rücken Sie die erste Zeile für den `p`-Selektor per *text-indent* ein.



Die erste Zeile spielt in der klassischen Typografie eine Sonderrolle. Auch CSS hat hier einiges zu bieten. Sie können eine eigene Einrückung festlegen und mit zwei Pseudoklassen den ersten Buchstaben und die erste Zeile anders formatieren.

```
4 p:first-letter {  
  color: red;  
  font-size: 2em;  
}
```

```
5 p:first-line {  
  font-family: Verdana, sans-serif;  
}
```

6 **D**er Anfang des Absatzes. Ein Absatz mit Text und mehreren Zeilen. Ein Absatz mit Text und mehreren Zeilen. Ein Absatz mit Text und mehreren Zeilen. Ein Absatz mit Text und mehreren Zeilen.

- 4 Mit der Pseudoklasse *p:first-letter* definieren Sie eigene CSS-Angaben für den ersten Buchstaben.
- 5 Die Pseudoklasse *p:first-line* vergibt CSS-Angaben für die komplette erste Zeile.
- 6 Das Ergebnis sehen Sie nun im Browser.

Ende

Für *text-indent* verwenden Sie die Maßeinheiten, die auch bei Schriftgrößen zum Einsatz kommen. Am gebräuchlichsten sind Pixel und Prozent.

Pseudoklasse bezeichnet eine Klasse, die von CSS vorgegeben wird. Sie gilt nur für ein bestimmtes Tag und wird mit Doppelpunkt vom Tag-Selektor abgetrennt.

Die Formatierungen für die erste Zeile und den ersten Buchstaben sieht man im Web eher selten. Dennoch müssen Sie sich keine Sorgen um die Browserkompatibilität machen. Alle modernen Browser kommen damit zurecht.

Start

```

1 <head>
  <title>Horizontale Linien</title>
</head>
<body>
  <p>
    Ein erster Absatz.
  </p>
  <p>
    Ein zweiter Absatz.
  </p>
</body>
</html>

```

```

2 <body>
  <p>
    Ein erster Absatz.
  </p>
  <hr />
  <p>
    Ein zweiter Absatz.
  </p>
</body>

```

Das `<hr>`-Tag hat keinen Inhalt und wird deswegen gleich wieder geschlossen. Die Langform wäre `<hr></hr>`.

```

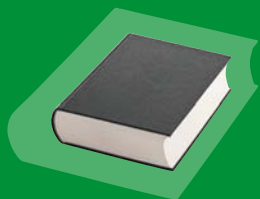
3
Ein erster Absatz.


---


Ein zweiter Absatz.

```

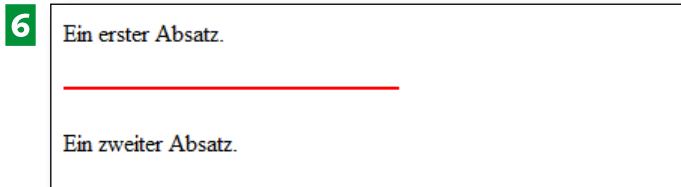
- 1 Starten Sie mit einer HTML-Seite, die mindestens zwei Absätze enthält.
- 2 Fügen Sie zwischen die beiden Absätze ein `<hr>`-Tag ein.
- 3 Sehen Sie sich die Linie im Browser an. Sie hat ein Standardaussehen, das sich je nach Browser leicht unterscheidet.



Horizontale Linien lassen sich dazu verwenden, Absätze voneinander zu trennen und logische Einheiten in Texten zu bilden. Die Basis ist das `<hr>`-Tag, ein Blockelement. Für die Linie können Sie die Breite und die Farbe steuern. Dies funktioniert entweder über die – als veraltet geltenden – HTML-Attribute oder über CSS.

```
4 <body>
  <p>
    Ein erster Absatz.
  </p>
  <hr style="color: red" />
  <p>
    Ein zweiter Absatz.
  </p>
</body>
```

```
5 <body>
  <p>
    Ein erster Absatz.
  </p>
  <hr align="left" style="color: red;
width: 50%" />
  <p>
    Ein zweiter Absatz.
  </p>
</body>
```



4 Vergeben Sie über einen Inline-Stil eine andere Farbe. Dies geschieht mit dem CSS-Befehl *color*.

5 Mit dem CSS-Befehl *width* fügen Sie eine Breite hinzu. Mit dem HTML-Attribut *align* vergeben Sie die Ausrichtung.

6 Sehen Sie sich das Resultat erneut im Browser an.

Ende

Soll die Trennlinie einen Verlauf enthalten oder optisch stärker angepasst sein, greifen Sie statt der Linie zu einem Bild.

Die HTML-Attribute für die horizontale Linie gelten als veraltet, weil die Formatierung mittlerweile komplett von CSS übernommen werden soll.

Um eine Linie ohne 3D-Effekt darzustellen, vergeben Sie eine Höhe (*height*) von 0 px und stattdessen mit dem CSS-Befehl *border* einen Rahmen.

Copyright

Daten, Texte, Design und Grafiken dieses eBooks, sowie die eventuell angebotenen eBook-Zusatzdaten sind urheberrechtlich geschützt. Dieses eBook stellen wir lediglich als **persönliche Einzelplatz-Lizenz** zur Verfügung!

Jede andere Verwendung dieses eBooks oder zugehöriger Materialien und Informationen, einschließlich

- der Reproduktion,
- der Weitergabe,
- des Weitervertriebs,
- der Platzierung im Internet, in Intranets, in Extranets,
- der Veränderung,
- des Weiterverkaufs und
- der Veröffentlichung

bedarf der **schriftlichen Genehmigung** des Verlags. Insbesondere ist die Entfernung oder Änderung des vom Verlag vergebenen Passwortschutzes ausdrücklich untersagt!

Bei Fragen zu diesem Thema wenden Sie sich bitte an: info@pearson.de

Zusatzdaten

Möglicherweise liegt dem gedruckten Buch eine CD-ROM mit Zusatzdaten bei. Die Zurverfügungstellung dieser Daten auf unseren Websites ist eine freiwillige Leistung des Verlags. **Der Rechtsweg ist ausgeschlossen.**

Hinweis

Dieses und viele weitere eBooks können Sie rund um die Uhr und legal auf unserer Website herunterladen:

<http://ebooks.pearson.de>