

2 HTML & CSS – der Schnelleinstieg *

Diese Einführung ist für Leser gedacht, die noch keinerlei Kenntnisse in **HTML** oder **CSS** besitzen. Wenn Sie bereits Grundkenntnisse besitzen, dann können Sie mit dem Kapitel »Web-Anwendungen – Grundlagen«, S. 29, starten.

Hinweis

Die wichtigsten Konzepte und Sprachelemente von HTML werden in diesem intuitiven Schnelleinstieg anhand des Beispiels »Meine Website« vermittelt. Es wird eine einfache private **Website** entwickelt, die aus mehreren **Webseiten** besteht. Die private Website soll folgende Inhalte im Web anzeigen:

Nur Basiswissen

- Persönliche Angaben (Text und Bild),
- Angaben zum Studium und
- Angaben zu Hobbys und Interessen.

Außerdem sollen mit Hilfe eines Formulars Nachrichten verfasst und versendet werden.

Ausgangspunkt ist die sogenannte **Startseite** oder **Homepage** der Website. Die Startseite wird zunächst durch ein einzelnes HTML-Dokument realisiert, in dem alle Angaben zusammengefasst werden:



- »Startseite erstellen«, S. 6

Im Anschluss werden weitere Webseiten erstellt und miteinander verbunden:

- »Webseiten verbinden«, S. 12

Bilder, Formulare und Tabellen werden in die Webseiten eingefügt:

- »Bilder einfügen«, S. 15
- »Formulare verwenden«, S. 17
- »Tabelle einsetzen«, S. 26

Das Layout einer Website kann mit Hilfe der Zusatzsprache CSS gestaltet werden:

- »CSS verwenden«, S. 21

Das hier erlernte Wissen sollten Sie gleich dazu nutzen, eine eigene kleine Website zu erstellen.



HTML5

Die Beispiele in diesem Buch sind – wenn nicht anders angegeben – in der **HTML-Syntax von HTML5** angegeben. Im Kapitel »HTML und XHTML – die Grundlagen«, S. 45, wird auf die Unterschiede zwischen HTML und XHTML sowie auf die unterschiedlichen Standards für die Erstellung von Webseiten eingegangen.

Texteditor



Beim Durcharbeiten dieses Schnelleinstiegs sollten Sie die Beispiele – z. B. mit einem einfachen **Editor** – erfassen und in einer Datei mit der Endung `.html` speichern. Diese Datei kann dann in den **Webbrowser** geladen und sofort ausgeführt werden.

Tipp

Den Editor finden Sie im Betriebssystem Windows in der Startleiste. Wenn Sie eine Datei speichern, die *keine* Textdatei ist, müssen Sie darauf achten, dass Sie die richtige Dateierweiterung (`.html`) und die richtige Codierung (empfohlen wird UTF-8) angeben. HTML-Dateien werden im Editor beim Öffnen einer Datei nur angezeigt, wenn Sie bei Dateityp die Einstellung alle Dateien wählen. Wenn Sie ein UNIX/Linux-System einsetzen, dann können Sie als Editor z. B. den `vi` benutzen.

2.1 Eine Startseite erstellen *

Mit HTML lassen sich auf einfache Weise Websites erstellen, die im Webbrowser angezeigt werden. Einzelne HTML-Elemente klassifizieren den Inhalt in entsprechende Kategorien, z. B. Überschrift, Listen, Tabellen. HTML-Elemente bestehen in den meisten Fällen aus einer Anfangs-, einer Endmarkierung und aus einem Inhalt. Der Inhalt kann auch entfallen.

HTML (*HyperText Markup Language*) ist eine sogenannte Dokumenten- bzw. Textauszeichnungssprache, die HTML-Elemente zur Verfügung stellt, um Texte auszuzeichnen. Bei der Darstellung eines so ausgezeichneten Textes in einem Webbrowser werden die **HTML-Elemente** entsprechend ihrer in HTML festgelegten Bedeutung interpretiert und umgesetzt. Alle HTML-Elemente verwenden vorgegebene **Anfangs-** und **Endmarkierungen** zur Auszeichnung, welche als *Tags* bezeichnet werden.

Beispiel

In dem folgenden Text soll das Wort »nicht« hervorgehoben dargestellt werden. Mit Hilfe des HTML-Elements `` sieht das wie folgt aus:

Du sollst `nicht` lügen!

Die Anzeige im Webbrowser sieht wie folgt aus:

Du sollst **nicht** lügen!

Aufbau eines
HTML-
Dokuments

Ein **HTML-Dokument** besitzt im Allgemeinen folgenden Aufbau:

```

1 <!DOCTYPE HTML>
2 <html>
3 <head>
4   <meta charset="UTF-8">
5   <title>
6       Titel wird in der Titelleiste
7       des Browsers angezeigt

```

```

8     </title>
9 </head>
10 <body>
11     <p>Hier steht der Inhalt des HTML-Dokuments</p>
12 </body>
13 </html>

```

Jedes HTML-Dokument beginnt mit der Dokumenttyp-Deklaration. Diese gibt u. a. an, welche HTML-Elemente verwendet werden dürfen. In Zeile 1 wird mit dem Schlüsselwort **HTML** angegeben, dass der Standard-HTML-Dokumenttyp (*normal doctype*) verwendet wird.

Der Inhaltsbereich (Zeile 2 bis 12) gliedert sich in zwei Teile:

- Den **Kopfteil** (*header*) (Zeilen 3 bis 8) und
- das eigentliche Dokument, **Rumpf** (*body*) genannt (Zeilen 9 bis 11).

Beide Teile werden durch das **Wurzelement** (*root*) umklammert (Zeile 2 bis Zeile 12).

In der Regel wird ein HTML-Dokument als Datei mit der Dateierendung `.html` oder `.htm` (Microsoft-Standard) abgespeichert. Manchmal besteht ein HTML-Dokument aber auch aus mehreren Dateien. Daher wird zwischen dem logischen HTML-Dokument und der physischen HTML-Datei unterschieden.

Dokument vs.
Datei

HTML-Dokumente bestehen aus sogenannten HTML-Elementen, deren Inhalt durch eine einleitende Markierung (Anfangsmarkierung) und eine abschließende Markierung (Endmarkierung) umschlossen wird. Die Anfangsmarkierung enthält den Elementnamen, der in spitzen Klammern eingeschlossen wird. Die Endmarkierung wiederholt den Elementnamen, der jetzt das Zeichen `»/«` vorangestellt wird, und wird ebenfalls durch spitze Klammern begrenzt. Zwischen der Anfangs- und der Endmarkierung befindet sich der Elementinhalt.

Elementaufbau

Das HTML-Element `title` im Kopfteil nimmt den Titel des HTML-Dokuments auf, der in der Titelleiste des Browsers angezeigt wird:

```
<title>Titel des Dokuments</title>
```

Beispiel
HTML-Element

Es gibt HTML-Elemente ohne Inhalte, die *keine* explizite Endmarkierung benötigen. Dann wird das Zeichen `»/«` bereits in die Anfangsmarkierung – und damit einzige Markierung – aufgenommen, in diesem Fall *hinter* dem Elementnamen.

Das HTML-Element `br` bewirkt einen Zeilenumbruch. Es benötigt keine Endmarkierung:

Beispiel
HTML-Elemente
ohne Inhalt

```
Text vor dem Zeilenumbruch
<br/>
Text nach dem Zeilenumbruch
```

Attribute Einem HTML-Element können durch sogenannte Attribute weitere Eigenschaften zugeordnet werden. Attribute werden in der Anfangsmarkierung in folgender Schreibweise angegeben: `Attributname = "Attributwert"`

Beispiel
HTML-Element
mit Attribut

Mit Hilfe des Ankerbefehls `a` und dem Attribut `name` kann ein Ankerpunkt in einem HTML-Dokument gesetzt werden:

```
<a name="Zum Ende">Dies ist das Ende des Dokuments.</a>
```

Wurzelement
`html`

Der Inhalt eines HTML-Dokuments wird innerhalb des HTML-Elements `html` durch baumartig ineinander geschachtelte HTML-Elemente beschrieben. Das HTML-Element `html` ist daher das sogenannte Wurzelement dieser Baumstruktur.

Häufig
verwendete
Elemente

Im **Rumpf** werden folgende HTML-Elemente häufig verwendet:

- Überschriften in maximal 6 Stufen: `h1` bis `h6`
- Hervorhebungen und Betonungen: `strong` und `em`
- Textabsätze: `p`
- Geordnete und ungeordnete Listen: `ol` und `ul` .

Überschriften

HMTL bietet verschiedene **Formate für Überschriften** von der Überschriftsebene 1 (größte Schrift) bis zur Überschriftsebene 6 (kleinste Schrift). Mit Überschriften wird ein Dokument logisch strukturiert. Die Webbrowser stellen in der Regel die verschiedenen Überschriften in unterschiedlichen Schriftgrößen dar. Damit wird die logische Struktur auch in der Textdarstellung verdeutlicht.

Beispiel

Für die Überschriftenebene 2 sieht dies wie folgt aus:

```
<h2>Mein Name ist Marie Risser.</h2>
```

Hervorhebungen

Sollen Teile des Textes hervorgehoben oder betont werden, dann werden diese Texte innerhalb der HTML-Elemente `strong` oder `em` geschrieben:

```
<strong>Hervorgehobener Text (strong)</strong>
<em>Betonter Text (emphasis)</em>
```

Webbrowser stellen so gekennzeichnete Texte in Fettdruck (bei `strong`) oder Kursivschrift (bei `em`) dar. Die früher verwendeten Elemente `b` (Bedeutung »bold« (Fettdruck)) und `i` (Bedeutung »italic« (Kursivschrift)) sollten Sie *nicht* mehr verwenden. Die »veralteten« Elemente haben nicht nur die logische Struktur eines Dokuments gekennzeichnet, sondern auch die Darstellung dieser angeben. Zwecks besserer Wartbarkeit von Webseiten, sollte

die Struktur und die Darstellung möglichst vollständig getrennt werden.

Ein Text innerhalb des HTML-Elements `p` bildet einen **Textabsatz**, der vom Webbrowser automatisch durch Leerraum vom umgebenden Text abgesetzt wird.

Textabsatz

```
<p>Ich studiere an der FH Dortmund Wirtschaftsinformatik
und wohne in Bochum.</p>
```

Beispiel

Ein **Übergang in eine neue Zeile** wird durch das HTML-Element `br` erreicht. Da das Element über keinen Inhalt verfügt, existiert *keine* Endmarkierung: `
`.

Zeilenumbruch

Ältere Webbrowser können eine Anfangsmarkierung mit `/` u. U. nicht richtig verarbeiten. Wenn Sie davon ausgehen müssen, dass Ihre Zielgruppe ältere Webbrowser verwendet, dann sollten Sie vor den Schrägstrich ein Leerzeichen setzen, z. B. `
` statt `
`.

Hinweis

Auch Listen lassen sich einfach realisieren. Die folgenden Anweisungen zeigen, wie eine **ungeordnete Liste** ausgezeichnet wird.

Listen

```
<ul>
  <li><em>Stirb Ewig</em> von Peter James</li>
  <li><em>Romantiker</em> von Annette Mingels</li>
</ul>
```

Beispiel

Eine **geordnete Liste** erhalten Sie, wenn Sie anstelle des HTML-Elements `` und `` (für *ungeordnet*) das HTML-Element `` und `` (für *geordnet*) verwenden.

Das Element `meta` mit dem Attribut `charset` innerhalb des Kopfbereichs des HTML-Dokuments legt durch die Angabe eines Zeichensatzes fest, wie ein Webbrowser u. a. die Umlaute im HTML-Dokument interpretiert. Der Zeichensatz UTF-8 stellt eine weit verbreitete internationale Codierung dar. Dieser beinhaltet die Zeichen aller bekannten Schriften, somit auch die in europäischen Sprachen auftretenden Sonderzeichen, wie z. B. die deutschen Umlaute oder die Zeichen mit Akzenten im Französischen.

Zeichensatz

```
<head>
  <meta charset="UTF-8">
  <title> Titel des HTML-Dokuments </title>
</head>
```

Beispiel

Wie in anderen Programmiersprachen auch, können **Kommentare** in HTML-Dokumente aufgenommen werden. Die Kommentare werden bei der Auswertung und Interpretation der HTML-Doku-

Kommentare

mente ignoriert, sind aber im Quelltext des HTML-Dokuments sichtbar.

Beispiel

```
<!-- Dies ist ein Kommentar,
      der vom Browser nicht ausgewertet wird. -->
```

Beispiel
»Meine
Website«
Stufe 1

Mit den genannten HTML-Elementen kann die erste Variante der Website erstellt werden. Sie besteht in diesem Entwicklungsschritt aus genau einer Webseite, der Startseite (Homepage), die alle Texte enthält. Auf ein Bild und ein Formular wird zunächst verzichtet.

index.html

```
<!DOCTYPE HTML>
<html>
<head>
  <meta charset="UTF-8">
  <title>Personal Website Marie Risser</title>
</head>
<body>
  <h1>Schön, dass Sie meine Website besuchen!</h1>
  <h2>Mein Name ist Marie Risser.</h2>
  <p>Ich studiere an der FH Dortmund
    Wirtschaftsinformatik und wohne in Bochum.</p>

  <p>Sie finden hier Interessantes zu</p>
  <ol>
    <li>meinem Studium</li>
    <li>meinen Hobbys und Interessen</li>
  </ol>
  <p>Nehmen Sie sich etwas Zeit und lesen Sie,
    welche Erfahrungen ich gesammelt habe und
    was mich besonders interessiert.</p>
  <h2>Mein Studium</h2>

  <p>Besonders interessant waren bisher
    folgende Vorlesungen:</p>
  <ul>
    <li>Web Engineering</li>
    <li>... und noch manche andere Vorlesung ...</li>
  </ul>

  <h2>Meine Hobbys und Interessen</h2>

  <p>Ich treibe gerne Sport, lese viel und fotografiere
    Land und Leute mit meiner neuen Digitalkamera.</p>
  <p>Meine derzeitigen
    Lieblingsbücher sind:</p>
  <ul>
    <li><em>Stirb Ewig</em> von Peter James</li>
    <li><em>Romantiker</em> von Annette Mingels</li>
  </ul>
</body>
</html>
```

Die Abb. 2.1-1 zeigt die Anzeige der Startseite (Homepage) im Webbrowser.



Abb. 2.1-1: So sieht die Website in der einfachsten Version aus.

Um den Quelltext des HTML-Dokuments übersichtlicher zu gestalten, werden die HTML-Elemente, die innerhalb der HTML-Elemente head und body auftreten, drei Zeichen (oder eine andere festgelegte Anzahl von Zeichen) nach rechts eingerückt. Komfortable Editoren bieten Funktionen zur automatischen Formatierung (nach festgelegten Regeln) des gesamten HTML-Dokuments an.

Einrückung

Nehmen Sie das obige Beispiel als Vorbild und erstellen Sie eine eigene Startseite (Homepage). Legen Sie eine entsprechende Textdatei mit einem Editor an und speichern Sie die Datei mit der Endung `.html`. Starten Sie das HTML-Dokument dann durch einen Doppelklick auf den Dateinamen. Sie sehen das Ergebnis in Ihrem Browser.



2.2 Webseiten verbinden *

Eine Website besteht normalerweise aus mehreren HTML-Dokumenten, die durch Hyperlinks (Kurzform: Links) bzw. Verweise miteinander verbunden werden. Links werden standardmäßig unterstrichen. Klickt der Benutzer mit der Maus auf einen Link, dann wird das entsprechende HTML-Dokument im Webbrowser angezeigt. Für Links wird das HTML-Element `a` verwendet. Ein Link muss mindestens das Attribut `href` und einen Inhalt beinhalten.

- | | |
|----------------------|---|
| Webseiten aufteilen | In der Stufe 1 des Beispiels »Meine Website« wird ein einzelnes HTML-Dokument verwendet, das alle Angaben enthält. Diese Variante ist nur in Ausnahmefällen sinnvoll, wenn der Umfang der Webseite gering ist und die Inhalte in einem engen thematischen Zusammenhang stehen. Sobald mehr Informationen angeboten werden sollen, die sich zudem auf verschiedene Gesichtspunkte beziehen, ist eine Aufteilung des Informationsangebots auf mehrere Webseiten erforderlich. |
| Webseite vs. Website | Ein HTML-Dokument stellt in der Regel eine Webseite dar, die im Browser angezeigt wird. Mehrere Webseiten, die untereinander durch Hyperlinks sinnvoll miteinander verbunden sind, bilden eine sogenannte Website . Der Aufbau einer Website sieht beispielsweise wie in der Abb. 2.2-1 aus. |
| Startseite | Die Startseite stellt den Einstieg in die Website dar. Die meisten Internet-Benutzer beginnen mit dieser Seite den Einstieg in eine Website. Die Startseite gibt meist einen Überblick über den Inhalt der Website, oft in Form eines Inhaltsverzeichnisses (<i>site map</i>). Die Startseite wird oft auch Homepage genannt. Als Homepage bezeichnet man oft aber auch die gesamte Website einer Privatperson. |
| URI | Jede Webseite muss durch eine eindeutige Adresse im Internet identifizierbar sein. Dazu verwendet man den URI (<i>Uniform Resource Identifier</i>) mit einer normierten Syntax. Beispielsweise führt der URI <code>http://www.w3.org</code> zur Startseite des W3C (<i>World Wide Web Consortium</i>). Das W3C legt die Standards zur Nutzung des <i>World Wide Web</i> fest. |
| Hyperlink | Ein Hyperlink – kurz Link genannt –, der auf ein anderes HTML-Dokument verweist, wird mit dem HTML-Element <code>a</code> für Anker (<i>anchor</i>) beschrieben. Im einfachsten Fall wird direkt auf die Datei des HTML-Dokuments verwiesen, z. B. <code>vorlesung.html</code> . Als Inhalt des HTML-Elements <code>a</code> wird der Text angegeben, der im Webbrowser den Hyperlink optisch darstellt. Der Wert des Attributs <code>href</code> ist der URI des Verweisziels. |

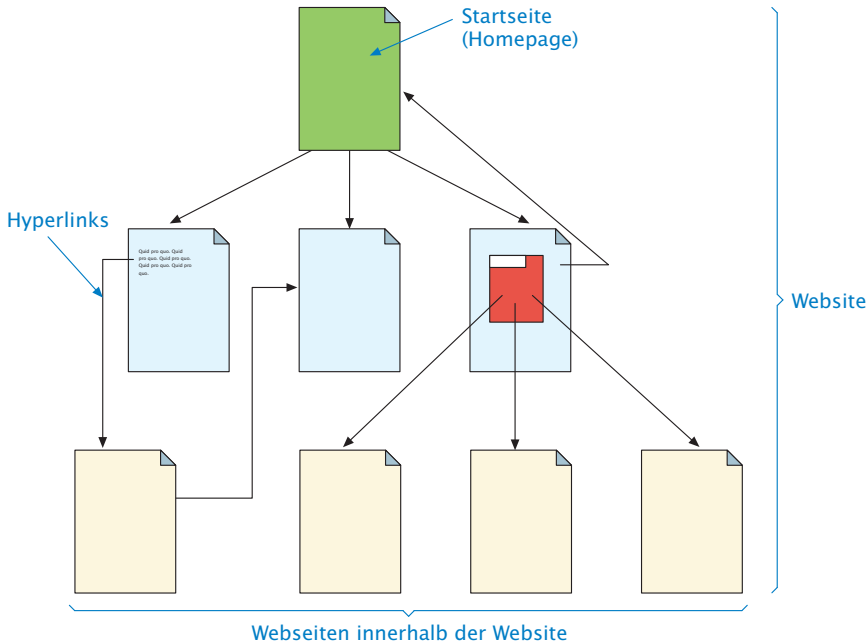


Abb. 2.2-1: So sieht beispielsweise die Struktur einer Website aus.

```
<a href="studium.html">Mein Studium</a>
```

Beispiel

Durch die Nutzung von Verweisen kann die einfache Website nun auf drei HTML-Dokumente verteilt werden. Die unter den Überschriften »Mein Studium« und »Meine Hobbys und Interessen« vorhandenen Inhalte werden Bestandteil der neuen HTML-Dokumente `studium.html` und `hobbys.html`. An die Stelle der Inhalte in der Startseite treten die Verweise auf die beiden neuen HTML-Dokumente. Es ist üblich, dass die Startseite den Namen `index.html` besitzt.

Beispiel
»Meine
Website«
Stufe 2

```
<!DOCTYPE HTML>
<html>
<head>
  <meta charset="UTF-8">
  <title>Personal Website Marie Risser</title>
</head>
<body>
  <h1>Schön, dass Sie meine Website besuchen!</h1>
  <h2>Mein Name ist Marie Risser.</h2>
  <p>Ich studiere an der FH Dortmund
    Wirtschaftsinformatik und wohne in Bochum.</p>
  <p>Sie finden hier Interessantes zu</p>
  <ol>
    <li>meinem Studium</li>
```

`index.html`

```

        <li>meinen Hobbys und Interessen</li>
    </ol>
    <p>Nehmen Sie sich etwas Zeit und lesen Sie,
        welche Erfahrungen ich gesammelt habe und
        was mich besonders interessiert.</p>
    <p>
        <a href="studium.html">
            Mein Studium
        </a><br/>
        <a href="hobbys.html">
            Meine Hobbys und Interessen
        </a>
    </p>
</body>
</html>

```

Auf der Startseite sind die Hyperlinks deutlich zu sehen (Abb. 2.2-2).



Abb. 2.2-2: Startseite mit Hyperlinks.

Alle Informationen, die das Studium betreffen, werden in das HTML-Dokument `studium.html` verschoben. Analog wird ein HTML-Dokument `hobbys.html` erstellt. In beiden Dokumenten wird ein Verweis auf die Startseite gesetzt.

`studium.html`
(Auszug)

```

<body>
    <h2>Mein Studium</h2>
    <p>Besonders interessant waren bisher folgende
        Vorlesungen:</p>
    <ul>
        <li>Web Engineering</li>
        <li>... und noch manche andere Vorlesung ...</li>
    </ul>

```

```

</ul>
<div>
  <a href="index.html">Zurück zur Startseite</a>
</div>
</body>

```

Die Abb. 2.2-3 zeigt das aufgerufene HTML-Dokument studium.html.



Abb. 2.2-3: Die Zielseite des Hyperlinks.

Erweitern Sie Ihre eigene Website entsprechend dem obigen Beispiel durch Links und zusätzliche HTML-Dokumente.



2.3 Bilder einfügen *

Bilder werden in HTML-Dokumenten *nicht* wie bei einem Textverarbeitungsprogramm direkt hinein kopiert, sondern sind immer als separate Datei verfügbar. Auf diese kann vom HTML-Dokument heraus mit dem HTML-Element `img` verwiesen werden.

Ein Bild (Grafik, Foto, Zeichnung) kann mit dem HTML-Element `img` in ein HTML-Dokument eingefügt werden, z. B.

```
.
```

Das Attribut `src` (für *source* = Quelle) gibt den Pfad zur Bilddatei an.

Bilder sollten immer in einem separaten Verzeichnis stehen. Es ist ein De-facto-Standard, dass dieses Verzeichnis den Namen `images` erhält.

Wird das HTML-Element `img` wie oben angegeben verwendet, dann muss der Webbrowser Höhe und Breite des Bildes selbst berechnen, was den Ladevorgang etwas verzögert. Daher ist es sinnvoll und allgemein üblich, Breite und Höhe explizit anzugeben. Man spricht hier von Attributen des Bildes. Das HTML-Element `img` wird dazu um zwei Attribute erweitert, z. B.

`img`-Element

Verzeichnis
`images`

Attribute
`width`, `height`

```
.
```

Attribut alt

Als weiteres Attribut *sollte* alt (für alternativ) angegeben werden. Der Wert des Attributs ist ein Text, der angezeigt wird, wenn das Bild durch den Webbrowser nicht dargestellt werden kann:

- Bei rein textbasierten Webbrowsern, die keine grafischen Elemente darstellen können,
- bei Webbrowsern für Personen mit Behinderungen, z. B. Blinde und stark Sehbehinderte; der Text kann dann in Brailleschrift ausgegeben oder auch gesprochen werden,
- bei modernen Webbrowsern, wenn auf die Bilddatei, z. B. mangels passender Zugriffsrechte, nicht zugegriffen werden kann.

Viele Suchmaschinen werten den Text des alt-Attributs aus. Bei modernen Webbrowsern wird dieser Text außerdem als **Tooltip** des Bildes angezeigt. Damit werden beim HTML-Element img jetzt vier Attribute verwendet:

```
.
```

Beispiel
»Meine
Website«
Stufe 3

index.html
(Auszug)

Im folgenden HTML-Dokument wird das HTML-Element img in einen Textabsatz geschachtelt (HTML-Element p). Damit wird erreicht, dass das Bild für sich steht, d. h. dass kein Text neben dem Bild angezeigt wird.

```
<body>
  <h1>Schön, dass Sie meine Website besuchen!</h1>
  <h2>Mein Name ist Marie Risser.</h2>
  <p>
    
  </p>
  <p>Ich studiere an der FH Dortmund
    Wirtschaftsinformatik und wohne in Bochum.</p>
  ...
</body>
```

Das Ergebnis im Webbrowser zeigt die Abb. 2.3-1.



Ergänzen Sie Ihre eigene Startseite um ein oder mehrere Bilder. Verwenden Sie das HTML-Element img mit und ohne Höhen- und Breitenangaben. Stellen Sie einen Unterschied fest? Schachteln Sie Ihre Bilder einmal in einen Textabsatz (HTML-Element p) und einmal nicht.



Abb. 2.3-1: Startseite mit Bild.

2.4 Formulare verwenden *

Über HTML-Formulare kann der Benutzer Daten erfassen. Dafür werden die HTML-Elemente `form`, `input` und `textarea` verwendet. Eine Verarbeitung der eingegebenen Daten kann *nicht* mit HTML erfolgen.

HTML-**Formulare** (*forms*) ermöglichen es, Daten des Benutzers zu erfassen und an den **Webserver** zur Verarbeitung zu übersenden. Eine Verarbeitung der Daten *vor* der Übertragung ist mit HTML *nicht* möglich. Dazu müssen andere Webtechniken eingesetzt werden.

Ein Formular wird durch das HTML-Element `form` gekennzeichnet. Sinnvoll, aber z. B. bei HTML5 nicht zwingend erforderlich, ist die Angabe des Attributs `action`, als dessen Wert ein **URI** (*Uniform Resource Identifier*) eingetragen wird. Damit kann durch den Webserver erkannt werden, welche Programmdatei für die Verarbeitung der Formulardaten auf dem Webserver vorgesehen ist.

Formularaufbau

Beispiel
form

```
<form action="" >
  <!-- Hier stehen die Eingabefelder -->
</form>
```

Hier wird im Attribut action eine leere Zeichenkette verwendet, weil noch keine Weiterverarbeitung der Daten stattfindet.

Eingabefelder

Innerhalb eines Formulars können Sie verschiedene Elemente verwenden. Besonders häufig wird das einfache Feld für Texteingaben (`<input type="text" name="feldname"/>`) benötigt. Der Name des Feldes ist notwendig, damit die Programme auf dem Webserver die eingegebenen Daten nach ihrer Bedeutung erkennen und entsprechend weiter verarbeiten können. Namen bestehen aus Buchstaben und Ziffern und dürfen nur mit einem Buchstaben beginnen. Die Größe des Feldes kann mit dem Attribut (*size*) angegeben werden. Die Größe entspricht nur dann der Anzahl der sichtbaren Zeichen, wenn zur Darstellung eine Schrift mit fester Schrittweite (einheitliche Breite der Zeichen) verwendet wird. Bei einer Proportionalschrift sind die Zeichen unterschiedlich breit, *size* bezieht sich dann nur auf die mittlere Zeichenbreite. Es können also unterschiedlich viele Zeichen im Feld dargestellt werden.

Beispiel
input

```
<form>
<input type="text" name="email" size="40" />
</form>
```

Hinweis

Belegen Sie das Attribut name immer mit einem sinnvollen Wert. Wenn Sie das Attribut weglassen, wird der Wert des Eingabefeldes *nicht* zum Webserver übertragen!

Mehrzeilige
Eingaben

Mit dem HTML-Element `textarea` können mehrzeilige Eingabefelder vorgesehen werden. Damit können Sie längere Texte erfassen. Die beiden Attribute `cols` und `rows` bestimmen die Anzahl der Zeichen je Zeile und die Anzahl der Reihen. Innerhalb des HTML-Elements `textarea` kann ein Text notiert werden, der in dem Textfeld als Voreinstellung erscheint.

Beispiel
textarea

```
<textarea cols="40" rows="5" name="anliegen">
  Voreinstellung
</textarea>
```

Führungstext

Damit der Benutzer weiß, welche Informationen er in ein Textfeld eingeben soll, sollte direkt vor dem HTML-Element `input` ein sogenannter **Führungstext** geschrieben werden, der die Bedeutung des Eingabefelds beschreibt.

Beispiel
Führungstext

```
E-Mail-Adresse
<input type="text" name="email" size="40" />
```

Diese einfache Form hat den Nachteil, dass es keinen logischen Zusammenhang zwischen dem Führungstext und dem Eingabefeld gibt.

Um diesen Zusammenhang herzustellen, wurde das HTML-Element `label` eingeführt. Es kennzeichnet einen Führungstext, der logisch mit einem Eingabefeld verbunden ist. Dazu sind zwei Angaben erforderlich:

- Das Eingabefeld muss durch das Attribut `id` eindeutig im HTML-Dokument identifizierbar sein.
- Beim HTML-Element `label` muss das Attribut `for` den exakten Wert des Attributs `id` des Eingabefeldes enthalten.

```
<label for="id_email">E-Mail-Adresse</label>
<input type="text" name="email" id="id_email" size="40"/>
```

HTML-Element
`label`

Beispiel
`label`

Die beiden Attribute `id` und `name` dienen in unterschiedlicher Weise zur Identifizierung von HTML-Elementen. Mit dem Attribut `id` kann ein Webbrowser einzelne HTML-Elemente eindeutig referenzieren. Dagegen dient das Attribut `name` zur Kennzeichnung von Daten, die zum Webserver übertragen werden. Es ist zulässig, für beide Attribute denselben Wert anzugeben.

`id` vs. `name`

Die Identifikation eines HTML-Elements, die mit dem Attribut `id` vorgenommen wird, muss innerhalb eines HTML-Dokuments eindeutig sein. D. h. es dürfen keine zwei HTML-Elemente mit derselben Identifikation vorkommen.

Eindeutige `id`

Innerhalb eines Formulars können Eingabefelder zu Gruppen zusammengefasst werden. Dazu verwenden Sie das HTML-Element `fieldset`. Innerhalb des HTML-Elements `fieldset` schreiben Sie die Eingabefelder, die Sie zu einer Gruppe zusammenfassen möchten. Die Gruppe wird automatisch durch eine Umrandung optisch hervorgehoben. Um einer Gruppe eine Beschriftung zu geben, wird innerhalb einer Gruppe – d.h. innerhalb des HTML-Elements `fieldset` – das HTML-Element `legend` verwendet.

Gruppierungen
mit `fieldset`

```
<fieldset>
  <legend>Ihre Daten</legend>
  <label for="id_name">Vor- und Nachname</label>
  <input type="text" name="name" id="id_name" size="40"/>
  <br/>
  <label for="id_email">E-Mail-Adresse</label>
  <input type="text" name="email" id="id_email" size="40"/>
</fieldset>
```

Beispiel
`fieldset`

Um die eingegebenen Daten zur Weiterverarbeitung abzusenden, wird eine Schaltfläche bzw. ein Druckknopf benötigt. Der Name dient wiederum zum Erkennen dieses Elementes bei der Verarbeitung durch den Webserver. Als »Wert« der Schaltfläche wird

Schaltfläche

der beim Attribut `value` vorhandene Wert verwendet. Diese Angabe dient gleichzeitig als Text auf der Schaltfläche.

Da mit dem Namen der Schaltfläche in vielen Fällen keine besondere Verarbeitung durch den Webserver verbunden ist, wird oft auf die Vergabe eines Namens und damit auch auf die Übertragung des Werts der Schaltfläche verzichtet.

Beispiel
submit

```
<input type="submit" value="Senden"/>
```

Beispiel
»Meine
Website«
Stufe 4
formular.html
(Auszug)

Damit das Formular der Abb. 2.4-1 angezeigt wird, sind im Inhaltsteil des HTML-Dokuments die nachstehenden Angaben erforderlich:

```
<body>
  <form>
    <h2>
      Wenn Sie mir eine Nachricht senden wollen ...
    </h2>
    <fieldset>
      <legend>Ihre Daten</legend>
      <label for="name">Vor- und Nachname</label>
      <input type="text" name="name" id="name"
        size="40"/>
      <br/>
      <label for="email">E-Mail-Adresse</label>
      <input type="text" name="email" id="email"
        size="40"/>
      <br/>
    </fieldset>
    <fieldset>
      <legend>Ihr Anliegen</legend>
      <label for="anliegen">Text</label>
      <textarea cols="40" rows="5" name="anliegen"
        id="anliegen">
      </textarea>
      <br/>
    </fieldset>
    <div>
      <input type="submit" value="Senden"/>
    </div>
    <p>
      Die Daten dieses Formulars werden noch nicht
      weiterverarbeitet!
    </p>
  </form>
  <div>
    <a href="index.html">Zurück zur Startseite</a>
  </div>
</body>
```



Ergänzen Sie Ihre eigene Website um ein Formular.

Personal Website Marie Risser - Mozilla Firefox

Personal Website Marie Risser

Wenn Sie mir eine Nachricht senden wollen ...

Ihre Daten

Vor- und Nachname

E-Mail-Adresse

Ihr Anliegen

Text

Die Daten dieses Formulars werden noch nicht weiterverarbeitet!

[Zurück zur Startseite](#)

Abb. 2.4-1: Einfaches Formular.

2.5 CSS verwenden *

Mit Hilfe der Zusatzsprache CSS können praktisch jedem HTML-Element Gestaltungsangaben zugeordnet werden. Eine einheitliche Gestaltung aller HTML-Dokumente einer Website ist leichter möglich, wenn die Gestaltungsregeln in einer separaten Datei gespeichert werden.

CSS (*Cascading Style Sheets*) ist eine Zusatzsprache zu HTML, mit der festgelegt werden kann, wie die Inhalte eines HTML-Dokuments im Browser dargestellt werden sollen.

Will man für alle HTML-Dokumente einer Website eine einheitliche Darstellung erreichen, dann werden die Stilangaben in CSS in eine separate Datei mit der Endung `.css` gespeichert und in alle HTML-Dateien, die diese Stilangaben verwenden sollen, importiert. Alle CSS-Dateien werden häufig in einem separaten Verzeichnis mit der Bezeichnung `css` abgelegt.

Zentrale
CSS-Datei

In jedem HTML-Dokument, in der die CSS-Stilregeln verwendet werden sollen, muss die entsprechende CSS-Datei »eingebunden« werden. Das erfolgt im HTML-Kopfteil `<head>` z. B. durch das Element `link`:

CSS-Datei
einbinden

```
<link rel="stylesheet" type="text/css" href="css/pws.css" />
```

Das Attribut `href` gibt in diesem Beispiel an, dass die CSS-Datei mit Namen `pws.css` in dem Verzeichnis `css` liegt.

Inhalt
CSS-Datei

Eine CSS-Datei enthält eine Abfolge aller **CSS-Stilregeln**, die verwendet werden sollen. Die Reihenfolge der Stilregeln ist dabei zunächst ohne Bedeutung. Jede Stilregel besteht aus

- einem **CSS-Selektor**, mit dem die Stilregel eindeutig identifiziert wird, und
- einer oder mehreren **CSS-Eigenschaften**, die durch den **CSS-Eigenschaftsnamen** und den CSS-Wert definiert werden.

Die CSS-Eigenschaften einer Stilregel werden zusammen in geschweifte Klammern gesetzt. Einzelne CSS-Eigenschaften werden durch Semikolons voneinander getrennt. Zwischen dem CSS-Eigenschaftsnamen und dem CSS-Wert wird ein Doppelpunkt notiert:

Syntax *CSS-Stilregel ::= Selektor { CSS-Eigenschaft + }*
CSS-Eigenschaft ::= CSS-Eigenschaftsnamen : CSS-Eigenschaftswert ... ;

Durch Kommata getrennt können auch mehrere Werte zu einer Eigenschaft aufgeführt werden.

Im einfachsten Fall stimmen die Werte der CSS-Selektoren mit den Bezeichnungen der HTML-Elemente überein.

Beispiel
»Meine
Website«

In allen HTML-Dokumenten der Website von Marie Risser sollen eine einheitliche Hintergrundfarbe und eine einheitliche Schriftart verwendet werden. Dazu werden dem HTML-Element `body` die Gestaltungseigenschaften **Hintergrundfarbe** und **Schriftfamilie** zugewiesen und als CSS-Stilregel in der CSS-Datei definiert:

```
body {
    background-color: GhostWhite;
    font-family: Verdana, Arial, sans-serif;
}
```

In diesem Beispiel ist der CSS-Selektor `body` identisch mit dem HTML-Element-Namen `body`.

Tipp

Notieren Sie innerhalb einer CSS-Stilregel jede CSS-Eigenschaft in einer eigenen Zeile und rücken Sie diese ein. Schließen Sie jede Zeile mit einem Semikolon ab.

Mit den CSS-Stilregeln können z.B. folgende Gestaltungseigenschaften spezifiziert werden:

- Vordergrund- und Hintergrundfarben
- Schriftgrößen und weitere Schrifteigenschaften
- Weitere Textgestaltung, insbesondere Ausrichtung
- Rahmen und deren Eigenschaften
- Größe und Positionierung von Elementen auf der Webseite

Bei vielen HTML-Elementen kann mit der CSS-Eigenschaft `background-color` eine Hintergrundfarbe eingestellt werden. Diese CSS-Eigenschaft wird vor allem genutzt, um HTML-Dokumenten als Ganzes eine Hintergrundfarbe zuzuweisen (HTML-Element `body`) oder um einzelne Textabschnitte hervorzuheben (HTML-Element `p`).

Hintergrund-
farbe

Mit der folgenden Angabe werden Textabschnitte durch einen helleren blauen Hintergrund hervorgehoben:

```
p {
  background-color: LightBlue;
}
```

Beispiel

In HTML können Farben durch **Farbnamen**, z. B. `LightBlue` oder `Red` angegeben werden. Alternativ sind auch numerische Angaben möglich. Die festgelegten Farbnamen können in beliebiger Klein- oder Großschreibung geschrieben werden. Es ist aber sinnvoll, sich an die in den W3C-Standards verwendeten Schreibweisen anzulehnen. Dort wird die sog. Kamelhöckernotation (*CamelCase*) verwendet, bei der jede Silbe im Farbnamen mit einem Großbuchstaben beginnt, z. B. bei `LightBlue`.

Farbangaben

Auch die **Schriftfarbe** kann ausgewählt werden.

Schriftfarbe

Die folgende Angabe sorgt dafür, dass die Schriftfarbe der Überschriftsebene 2 in einem speziellen rot dargestellt wird.

```
h2 {
  color: Crimson;
}
```

Beispiel

Die **Schriftgröße** lässt sich ebenfalls einstellen. Hierfür steht die CSS-Eigenschaft `font-size` zur Verfügung.

Schriftgröße

Die folgende Stilregel sorgt dafür, dass Textabschnitte größer dargestellt werden, als dies standardmäßig der Fall ist:

```
p {
  font-size: large;
}
```

Beispiel

Mit der CSS-Eigenschaft `font-family` kann eine Schriftart zur Darstellung vorgeschlagen werden. Ist die Schriftart auf dem System des Webbrowsers *nicht* installiert, muss der Webbrowser eine andere Schriftart ersatzweise verwenden.

Schriftart

Verwenden Sie bei der Angabe der Schriftart immer die Möglichkeit einer Vorzugsliste: Sie können die Schriftarten nacheinander angeben, die je nach Verfügbarkeit verwendet werden sollen. Schließen Sie die Liste mit einer der Ersatznamen ab, die durch das W3C definiert wurden. Damit ist sicherge-

Tipp

stellt, dass immer eine Schriftart verwendet wird, die der ursprünglich vorgesehenen Art am nächsten kommt.

Beispiel

Die folgende Angabe erweitert die Beschreibung der Gestaltungseigenschaften für das HTML-Element `body` um eine Voreinstellung für die Schriftarten. Wenn auf Ihrem System die Schriftarten installiert sind, wird der Webbrowser diese in der angegebenen Reihenfolge von links nach rechts verwenden. Die Schriftart `sans-serif` steht stellvertretend für alle auf dem System installierten serifenlosen Schriften, aus denen der Webbrowser dann eine auswählt, falls die anderen angegebenen Schriftarten nicht verfügbar sind.

```
body {
    font-family: Verdana, Arial, sans-serif;
}
```

Tipp

Die verschiedenen Webbrowser verwenden unterschiedliche Voreinstellungen für die Schriftarten und die Schriftgrößen, die in der Regel durch die Benutzer verändert werden können. Geben Sie in den CSS-Stilregeln Schriftgröße und Schriftart explizit an, wenn Sie ein möglichst einheitliches Aussehen der HTML-Dokumente erzielen wollen.

Textausrichtung

Die Ausrichtung von Textabschnitten kann durch die CSS-Eigenschaft `text-align` beeinflusst werden. Häufig wird die Eigenschaft zur Einstellung einer zentrierten oder rechtsbündigen Textausrichtung verwendet. Die linksbündige Textausrichtung ist der Standardfall.

Beispiel

```
p {
    text-align: right;
}
```

Beispiel
»Meine
Website«
Stufe 5
pws.css

Alle HTML-Dateien des Beispiels binden die CSS-Datei `pws.css` ein. Dadurch werden die CSS-Stilregeln für die gesamte Website wirksam.

```
/* Regeln für Standardelemente */
body {
    background-color: GhostWhite;
    font-family: Verdana, Arial, sans-serif;
}
h1 {
    background-color: CornflowerBlue;
    color: White;
    text-align: center;
}
h2 {
```

```

    color: Crimson;
}

<head>
  <title>Personal Website Marie Risser</title>
  <meta charset="UTF-8">
  <link rel="stylesheet" type="text/css"
        href="css/pws.css"/>
</head>

```

index.html
(Auszug)

Das Ergebnis im Webbrowser zeigt die Abb. 2.5-1.



Abb. 2.5-1: Stufe 5 der Startseite.

Erstellen Sie für Ihre eigene Website ebenfalls eine CSS-Datei und legen Sie dort die von Ihnen gewünschten Stilregeln fest. Binden Sie diese Datei dann in die HTML-Dateien mit dem HTML-Element link ein.



2.6 Tabelle einsetzen *

Tabellen sind ein häufig verwendetes Element in HTML-Dokumenten. Sie werden u.a. mit Hilfe der HTML-Elemente `table`, `caption`, `tr`, `th` und `td` spezifiziert.

In vielen Webseiten werden Tabellen zur Anzeige von Aufzählungen eingesetzt.

Beispiel
»Meine
Website«

hobbys.html
(Auszug)

Die Liste der Bücher in der Website von Marie Risser lässt sich sehr gut mit einer Tabelle darstellen (Abb. 2.6-1). Der HTML-Quellcode dazu sieht folgendermaßen aus:

```
<table>
  <caption>Favoriten</caption>
  <tr>
    <th>Titel</th>
    <th>Autor</th>
  </tr>
  <tr>
    <td>Stirb Ewig</td>
    <td>Peter James</td>
  </tr>
  <tr>
    <td>Romantiker</td>
    <td>Annette Mingels</td>
  </tr>
</table>
```

Favoriten	
Titel	Autor
Stirb Ewig	Peter James
Romantiker	Annette Mingels

Abb. 2.6-1: Einfache Tabelle.

HTML-Elemente
für Tabellen

- Die komplette Tabelle wird mit dem HTML-Element `table` definiert.
- Eine Tabelle ist standardmäßig randlos (blinde Tabelle). In älteren HTML-Spezifikationen durfte das Attribut `border` in der Form `border="Randbreite in Pixel"` verwendet werden. Hatte die Randbreite dann einen Wert größer als 0, wurde ein sichtbarer Rand dargestellt. Obwohl die Webbrowser das Attribut `border` noch korrekt interpretieren, sollte es *nicht* mehr verwendet werden. Moderne Webseiten verwenden hierfür CSS-Stilregeln (siehe »CSS verwenden«, S. 21).

- Eine Tabellenüberschrift wird innerhalb des HTML-Elements `caption` notiert. Dieses Element muss unmittelbar auf die Anfangsmarkierung `<table>` folgen.
- Tabellen werden in HTML zeilenweise definiert. Für jede Zeile (*row*) wird das HTML-Element `tr` verwendet.
- Das HTML-Element `th` gibt innerhalb einer Zeile den Text im Kopf der Tabelle (*table head*) an. Standardmäßig heben die Webbrowser diesen durch Fettdruck und Zentrierung hervor.
- Jedes Datenfeld der Tabelle wird mit dem HTML-Element `td` festgelegt. Dieses darf nur innerhalb einer Tabellenzeile, also innerhalb des Elements `tr` vorkommen.

Bei Verwendung von Tabellen ist es ganz besonders wichtig, auf exaktes Einrücken der HTML-Elemente – jeweils nach rechts – zu achten, da sonst sehr leicht Fehler gemacht werden. Beachten Sie, dass die Einrückung keine syntaktische Bedeutung hat!

Richtig
einrücken

Mit den CSS-Stileigenschaften `border` und `padding` können u. a. HTML-Tabellen Rahmen sowie Innenabstände zugewiesen werden. Bei der CSS-Stileigenschaft `border` kann man gleichzeitig die Stärke des Rahmens, z. B. in Pixel, sowie die Art und die Farbe des Rahmens festlegen.

Rahmen und
Ränder

Die CSS-Datei `pws.css` wird um drei Stilregeln ergänzt. Mit diesen wird der gesamten Tabelle und den einzelnen Tabellenzellen ein dünner, durchgezogener Rand in der Farbe `silver` zugewiesen. Außerdem wird ein Innenabstand (Abstand zwischen dem Zelleninhalt und dem Zellenrand) spezifiziert.

Beispiel
»Meine
Website«
Stufe 6

```
table {
    border: 1px solid silver;
}

th {
    border: 1px solid silver;
    padding: 4px;
}

td {
    border: 1px solid silver;
    padding: 4px;
}
```

`pws.css`
(Auszug)

Die Abb. 2.6-2 zeigt die neue Variante des HTML-Dokuments mit der »verschönerten« Tabelle.

Ergänzen Sie Ihre eigene Website um eine Tabelle.





Abb. 2.6-2: Neue Variante der Tabelle.