



HTML

Spielend gelingt die Website

TOBIAS HAUSER CHRISTIAN WENZ



HTML

HTML

Spielend gelingt die Website

TOBIAS HAUSER CHRISTIAN WENZ



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 Texten 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

12

ISBN 978-3-8272-4771-1

© 2012 by Markt+Technik Verlag,
ein Imprint der Pearson Deutschland GmbH,
Martin-Kollar-Straße 10–12, D-81829 München/Germany
Alle Rechte vorbehalten
Coverlayout: Thomas Arlt, tarlt@adesso21.net
Titelbild: plainpicture
Lektorat: Brigitte Bauer-Schiewek, bbauer@pearson.de,
Anne Herklotz, aherklotz@pearson.de
Fachlektorat: Dirk Frischalowski
Korrektorat: Marita Böhm, München
Herstellung: Monika Weiher, mweiher@pearson.de
Satz: Ulrich Borstelmann, Dortmund (www.borstelmann.de)
Druck und Verarbeitung: Drukarnia Dimograf, Bielsko-Biala
Printed in Poland



Kapitel 3

Verknüpfungen

Eine Webseite besteht aus einer einzelnen HTML-Datei und eventuell noch ein paar Grafiken (mehr dazu in Kapitel 5). Eine Website dagegen besteht aus mehreren HTML-Dateien, die miteinander verbunden sind. Einen Verweis von einer HTML-Datei auf eine andere nennt man Link (oder Hyperlink). In diesem Kapitel erfahren Sie, wie Sie solche Links einrichten können und welche Möglichkeiten Ihnen hierbei zur Verfügung stehen.

Allgemeines

Für Links ist das `<a>`-Tag zuständig. Das `a` steht für *anchor*, engl.: Anker.

Syntax

Ein Link hat folgenden Aufbau:

```
<a href="...">Linktext</a>
```

Zu den einzelnen Elementen:

- Das `<a>`-Tag umfasst wie gesagt den Link.
- Das `href`-Attribut gibt das Linkziel an. Beispiele hierfür finden Sie im weiteren Verlauf des Kapitels zuhaufl!
- `Linktext` gibt den Text an, der als Link fungieren soll.

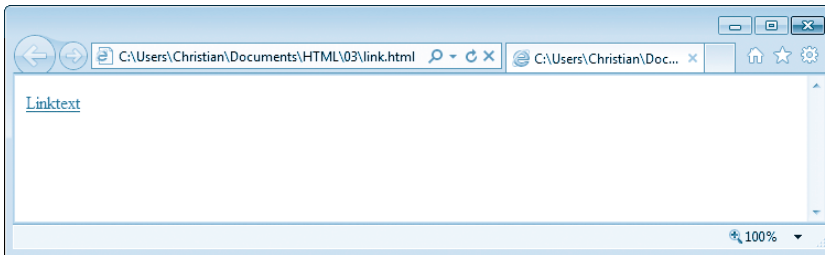


Abbildung 3.1: Der Link im Browser

In der Abbildung sehen Sie das Ergebnis: Der Linktext ist blau und unterstrichen. Wenn Sie mit dem Mauszeiger darüberfahren, verwandelt er sich in eine Hand. Ein Klick bewirkt, dass das Ziel des Links aufgerufen wird. Doch mehr dazu später!

Hinweis

Die Formatierung von Links ist nicht sonderlich anders als die Formatierung von Text, aber es gibt (noch) mehr Möglichkeiten. *Kapitel 4* enthält alle weiteren Informationen hierzu.

Interne Links

Wenn Sie auf eine andere Webseite verweisen wollen, gibt es prinzipiell zwei Möglichkeiten: Entweder ist diese Webseite Teil Ihres eigenen Angebots (und

damit auf demselben Server) oder sie ist Teil eines fremden Angebots (beispielsweise ein Link auf die Homepage des Markt+Technik Verlags). Als Basis stellen wir den Verzeichnisaufbau eines fiktiven Webangebots in den Raum, den Sie der Abbildung entnehmen können.

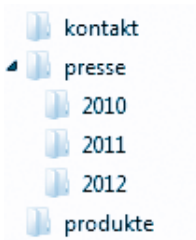


Abbildung 3.2: Der Verzeichnisbaum der (fiktiven) Website

Nehmen wir weiterhin an, in jedem der angegebenen Verzeichnisse befindet sich eine Datei *index.html* sowie eine Datei *info.html*. Von diesen Prämissen ausgehend, werden wir nun die verschiedenen Möglichkeiten der Verlinkung darstellen.

Links im gleichen Verzeichnis

Angenommen, Sie befinden sich gerade auf der Seite *index.html* im Verzeichnis *produkte* und wollen auf *info.html* verlinken. Dann geben Sie das genau so an:

```
<a href="info.html">Hier geht's zu info.html</a>
```

Sie sehen also: Einfach den Dateinamen angeben und schon ist der Link aktiv! Analog können Sie natürlich von der Datei *info.html* auf die Datei *index.html* zurückverweisen:

```
<a href="index.html">Zurück zu index.html</a>
```

So weit, so gut, das ist auch noch kein wirkliches Problem. Etwas schwieriger – aber eigentlich immer noch ziemlich einfach – ist es, auf Dateien in anderen Verzeichnissen zu verlinken. Sie haben dort diverse Möglichkeiten.

Relative Links

Verzeichnisnamen sind durch Trennzeichen aufgeteilt. Unter Windows ist das Trennzeichen der Backslash (\), unter Unix/Linux der Schrägstrich (/), unter Macintosh der Doppelpunkt (:). Bei einem Webserver ist es immer ein Schrägstrich. Wenn Sie nun auf eine Datei verweisen wollen, die in einem anderen Verzeichnis liegt als die aktuelle Datei, müssen Sie den Weg durch den Ver-

zeichnispfad angeben und die Namen der einzelnen Verzeichnisse durch Schrägstriche voneinander trennen.

Beginnen wir mit einem einfachen Beispiel: Wir befinden uns auf der Seite *index.html* im Hauptverzeichnis des Webserver und wollen die Datei *index.html* im Verzeichnis *presse* aufrufen. Der Link kann dann folgendermaßen aussehen:

```
<a href="presse/index.html">Zum Pressebereich</a>
```

Ist der Webserver so konfiguriert, dass bei der bloßen Angabe eines Verzeichnisnamens automatisch die Datei *index.html* geladen wird, geht auch folgende Variante:

```
<a href="presse/">Zum Pressebereich</a>
```

In der Regel ist es aber günstiger, direkt auf den Dateinamen zu verlinken, denn dann sind Sie von der Konfiguration des Webserver weitestgehend unabhängig.

Nun folgt ein minimal schwierigeres Beispiel: Von der Datei *index.html* im Hauptverzeichnis aus soll auf die Datei *info.html* im Unterverzeichnis 2012 des *presse*-Verzeichnisses verlinkt werden. Folgender HTML-Code erledigt das:

```
<a href="presse/2012/info.html">2012</a>
```

Sie sehen also: Verzeichnisnamen und Dateinamen werden aneinandergehängt und durch Schrägstriche voneinander getrennt.

Bis dato haben Sie nur gesehen, wie man ein oder mehrere Verzeichnisse tiefer in den Verzeichnisbaum einsteigen kann. Der Weg zurück blieb Ihnen bis dato verwehrt. Es gibt zwei besondere »Verzeichnisnamen«, mit denen so etwas möglich ist:

- `.` – bezeichnet das aktuelle Verzeichnis
- `..` – bezeichnet das darüberliegende Verzeichnis

Der einzelne Punkt zur Angabe des aktuellen Verzeichnisses wird eher selten eingesetzt. Angenommen, Sie befinden sich auf der Seite *info.html* im Hauptverzeichnis des Webserver und wollen auf die Datei *index.html* im gleichen Verzeichnis verlinken, dann können Sie das folgendermaßen erledigen:

```
<a href="./index.html">Zur Startseite</a>
```

Bei entsprechender Konfiguration des Webserver ist alternativ auch folgende Kurzform möglich – da kein Dateiname angegeben wird, ergänzt der Webserver automatisch *index.html*:

```
<a href=".">Zur Startseite</a>
```

Jetzt aber zum eigentlich Interessanten: dem Sprung zurück. Wir befinden uns in der Datei *info.html* im Unterverzeichnis *2012* des *presse*-Verzeichnisses und wollen auf die Datei *index.html* im Verzeichnis *presse* verlinken. Dieses Verzeichnis ist das direkt darüberliegende, also:

```
<a href="../../index.html">Zum Pressebereich</a>
```

Wie üblich ist bei entsprechender Webserver-Konfiguration auch die Kurzform möglich:

```
<a href="../../">Zum Pressebereich</a>
```

Um nun mehrere Verzeichnisse zurückzuspringen, müssen Sie den zweifachen Punkt mehrmals verwenden. Um also von der Datei *info.html* im Verzeichnis *presse/2012* auf die Datei *index.html* im Hauptverzeichnis zu verlinken, verwenden Sie folgenden Code:

```
<a href="../../../index.html">Zur Hauptseite</a>
```

Alternativ geht natürlich auch folgende Kurzform – bei den meisten Webservern jedenfalls:

```
<a href="../../../">Zur Hauptseite</a>
```

Wollen Sie nun aus dem Pressebereich (wie zuvor: Datei *presse/2012/info.html*) in den Kontaktbereich springen (dort Datei *index.html*), geht das mit derselben Technik:

```
<a href="../../../kontakt/index.html">Zum  
Kontaktbereich</a>
```

Oder alternativ die Kurzform:

```
<a href="../../../kontakt/">Zum Kontaktbereich</a>
```

Was passiert hier?

1. Am Anfang befinden wir uns im Verzeichnis *presse/2012*.
2. Durch das erste *../* wird ein Verzeichnis zurückgesprungen, ins Verzeichnis *presse*.
3. Durch das zweite *../* wird wiederum ein Verzeichnis zurückgesprungen, ins Hauptverzeichnis.
4. Durch *kontakt/* wird ins Verzeichnis *kontakt* gewechselt.
5. Durch *index.html* wird schließlich die Datei *index.html* aufgerufen.

Absolute Links

Die bisherige Verweistechnik wird als *relativ* bezeichnet – alle Verweise werden relativ zum momentanen Standpunkt aus angegeben, beginnen also mit einem Dateinamen oder einem Verzeichnisnamen.

Bei *absoluten* Links dagegen beginnt der Verweis mit einem Schrägstrich (/). Dies ist ein Verweis auf das Hauptverzeichnis des Webserver. Sie müssen also den gesamten Verzeichnisnamen der Datei angeben, auf die Sie verlinken wollen.

Wenn Ihnen das Ganze noch ein wenig zu theoretisch klingt, sollte es anhand folgender Beispiele klarer werden:

Egal von welcher Stelle aus, in Ihrer Anwendung verlinken Sie wie folgt auf die Datei *index.html* im Hauptverzeichnis:

```
<a href="/index.html">Zur Hauptseite</a>
```

Bei einem entsprechend konfigurierten Webserver geht das auch folgendermaßen, ohne Angabe des Dateinamens:

```
<a href="/">Zur Hauptseite</a>
```

Von jeder Seite Ihrer Anwendung aus verlinken Sie wie folgt auf die Datei *info.html* im Verzeichnis *presse/2012*:

```
<a href="/presse/2012/info.html">Presse 2012</a>
```

Wenn Sie sich also im Verzeichnis *produkte* und dort auf der Seite *index.html* befinden, können Sie auf die folgenden Arten und Weisen auf die Datei *info.html* im selben Verzeichnis verweisen:

1. Direkt durch Angabe des Dateinamens:

```
<a href="info.html">Linktext</a>
```

2. Relativ:

```
<a href="./info.html">Linktext</a>
```

3. Absolut:

```
<a href="/produkte/info.html">Linktext</a>
```

Optimale Strategie

Es bleibt die Frage zu klären, welche Art von Links Sie verwenden sollten – relative oder absolute Links (»direkte« Links bei Dateien im selben Verzeichnis sind streng genommen auch relative Links).

Absolute Links haben den Vorteil, dass sie von jeder beliebigen Seite und von jedem beliebigen Verzeichnis aus identisch sind. Sie können von überall aus mit / auf die Hauptseite verlinken. Wenn Sie bei all Ihren Seiten denselben Kopf verwenden (beispielsweise mit Ihrem Firmenlogo) und dabei auf Ihre Homepage verlinken wollen, sind absolute Links keine schlechte Idee. Sie können das beispielsweise bei allen Unterseiten von <http://www.pearson.de/> beobachten: Das Logo links oben ist überall gleich und verwendet absolute Links (siehe Hervorhebung in der Abbildung).

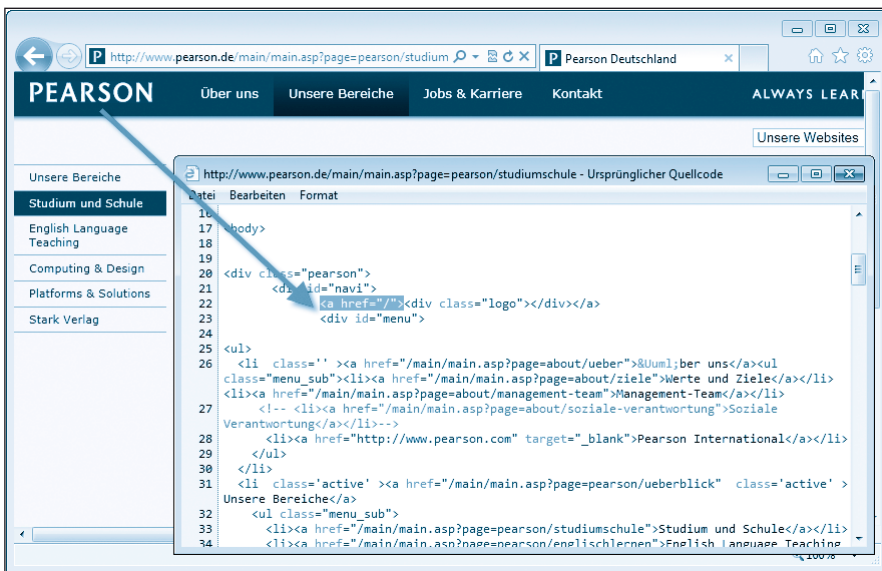


Abbildung 3.3: Pearson verwendet absolute Links.

Bei relativen Links ist es genau anders herum: Da die Links relativ zum Speicherort der aktuellen Datei sind, ist eine Verschiebung eines einzelnen Verzeichnisses meistens tödlich: Die relativen Links auf übergeordnete Verzeichnisse funktionieren dann nicht mehr. Andererseits haben relative Links auch ihre Vorteile: Ein Link auf "info.html" ist deutlich weniger Schreibaufwand als ein Link auf "/presse/2012/info.html".

Betrachten wir zwei mögliche Szenarien. Zunächst einmal der Pressebereich. Die Marketingabteilung hat festgestellt, dass neben Pressemitteilungen auch aktuelle News angeboten werden sollen. Konsequenz: Es gibt ein neues Verzeichnis */news* und diesem Verzeichnis wird das Verzeichnis *presse* untergliedert. Die ganze Sache kann dann ungefähr wie in der Abbildung aussehen.

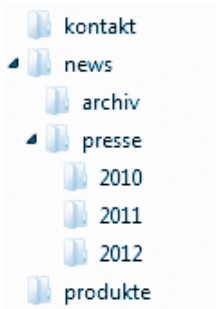


Abbildung 3.4: Die neue Verzeichnisstruktur

Was stellen Sie fest, wenn Sie nun bereits bestehende Links ausprobieren?

- Alle absoluten Links *in* das *presse*-Verzeichnis funktionieren nicht mehr.
- Alle anderen absoluten Links funktionieren weiterhin.
- Alle relativen Links (komplett) innerhalb des *presse*-Verzeichnisses oder (komplett) außerhalb des *presse*-Verzeichnisses funktionieren weiterhin.
- Relative Links vom *presse*-Verzeichnis nach außen oder umgekehrt funktionieren nicht mehr.

Im Rahmen der Fusionswelle hat sich Ihre Firma mit dem Hauptkonkurrenten zusammengeschlossen. Unter einem neuen Namen (möglichst mit *.com* oder *.de* im Firmentitel) versuchen Sie, den Markt zu beherrschen. Die neue Website ist eigentlich eine Portalseite mit zwei Verzeichnissen. Das erste enthält Ihren bisherigen Webauftritt, das zweite den Ihres neuen Partners (siehe Abbildung).



Abbildung 3.5: Die neue Verzeichnisstruktur

Ein Blick auf die Links bringt Folgendes zutage:

- Alle absoluten Links funktionieren nicht mehr.
- Alle relativen Links (komplett) innerhalb der einzelnen Anwendungen (Verzeichnis *firma1* bzw. *firma2*) funktionieren weiterhin.

Daraus können Sie das Fazit ziehen, dass relative Links in der Regel der bessere Weg sind, auch wenn das zweite Szenario ungleich unwahrscheinlicher ist als das erste. Oft hilft es auch, einen Mittelweg zu gehen: Verwenden Sie immer relative Links, außer wenn Sie auf Ihre Homepage (Startseite) verlinken, dort dann immer mit `/` oder `/index.html`.

Externe Links

In den Anfängen des WWW waren Linklisten *der* Hit. Jedermann stellte Verweise auf seine Lieblingsseiten online, denn zu diesem Zeitpunkt waren die grafischen Möglichkeiten auf Grund der fehlenden Infrastruktur noch nicht gegeben. Heutzutage ist das natürlich anders, es muss blitzen und krachen, Farbe regiert und Linklisten sind eher in den Hintergrund gerückt. Dennoch ist der gute alte *externe Link* nicht tot.

HTTP-Links

Wenn Sie auf ein anderes Webangebot verlinken wollen, müssen Sie den vollständigen Namen angeben – Servername, Verzeichnisse und Dateiname. Folgendermaßen verlinken Sie beispielsweise auf die Homepage des Markt+Technik Verlags:

```
<a href="http://www.mut.de/">Markt+Technik</a>
```

Genau genommen ist das sogar eine Kurzform, denn wie Sie sehen, haben wir den Dateinamen unterschlagen. Ausführlicher wäre folgende Form gewesen:

```
<a  
  href="http://www.mut.de/default.asp"  
>Markt+Technik</a>
```

Genauso können Sie in Unterverzeichnisse verlinken – folgender Link beispielsweise verweist den Besucher auf das Support-Angebot der Autoren, wo Sie übrigens unter anderem zusätzliche Informationen zu diesem Buch finden:

```
<a  
  href="http://www.hauser-wenz.de/support/"  
>Support zum "HTML Easy"</a>
```

FTP-Links

FTP steht für *File Transfer Protocol* und ist *das* Internetprotokoll für Dateiübertragungen im World Wide Web. Natürlich können Sie Dateien auch mit dem herkömmlichen Protokoll HTTP übertragen (HyperText Transfer Protocol, mit dem u. a. auch HTML-Seiten übertragen werden), doch FTP ist für diese Fälle besser geeignet. Beispielsweise können abgebrochene Downloads oftmals wieder aufgenommen werden, und zwar an der Stelle, an der sie unterbrochen wurden. Wenn Sie also Software aus dem Internet herunterladen wollen und die Wahl zwischen einem HTTP- und einem FTP-Link haben, sollten Sie im Normalfall den FTP-Server wählen.

An der Linksyntax ändert sich nichts, außer dass Sie als Protokollbezeichner *ftp* verwenden müssen:

```
<a href="ftp://ftp.microsoft.com/">FTP-Server  
Microsoft</a>
```

Natürlich können Sie auch direkt auf Dateien auf dem FTP-Server verlinken (Achtung, fiktiver Dateiname!):

```
<a href="ftp://ftp.microsoft.com/Windows8.exe">  
Die neue Windows-Version gratis zum Download!</a>
```

In das FTP-Protokoll sind Kommandos zur Authentifizierung via Benutzername und Passwort eingebaut. Die meisten FTP-Server erlauben den so genannten *anonymen Zugriff* auf ihren Datenbestand, das heißt, als Benutzername wird *anonymous* gesendet, als Passwort die E-Mail-Adresse des Benutzers. Die meisten Browser sind vom letzten Punkt inzwischen auch abgegangen (aus Datenschutzgründen) und senden nur noch eine Zeichenkette der Marke *hallo@du.da*.

Wenn eine Verbindung aber dennoch ein Passwort fordert, können Sie dieses Passwort direkt im Link angeben:

```
<a  
  href="ftp://benutzer:passwort@ftp.server.de/">  
FTP</a>
```

Achtung

Beachten Sie, dass das Passwort direkt aus dem Link ersichtlich und damit absolut unsicher ist. Sie sollten es sich also genauestens überlegen, bevor Sie einen solchen Link in Ihre Seiten einbauen.

Newsgroups

In den guten alten Zeiten, als das World Wide Web noch eine ganz kleine Rolle gespielt hat, geschah das meiste im Internet per Mail oder in Newsgroups. Eine Newsgroup ist ein weltweites Diskussionsforum. Per E-Mail können Sie eine Anfrage in einer Newsgroup stellen und auf der anderen Seite des Globus antwortet Ihnen vielleicht jemand. Newsgroups sind hierarchisch in mehrere Gruppen aufgeteilt.



Abbildung 3.6: Nachrichten in einer Newsgroup (über Webinterface)

Wir wollen an dieser Stelle nicht weiter in die Materie einsteigen, schließlich ist das hier ein HTML-Buch. Dennoch wollen wir Ihnen zeigen, wie Sie einen Link an eine Newsgroup richten können. Als Beispiel haben wir uns (natürlich) eine Newsgroup herausgesucht, die als Thema HTML hat. Die Newsgroup heißt *comp.infosystems.www.authoring.html*, ist englischsprachig und erhält sehr viele *Postings* (Beiträge). Wenn Sie direkt darauf linken wollen, bieten Sie folgenden Link an:

Zur HTML-Newsgroup

Achtung

Beachten Sie, dass auf *news:* kein *//* folgt!

Newsgroups werden auf vielen Servern weltweit gespeichert. Wenn eine Nachricht auf einem Server veröffentlicht wird, gibt dieser sie an die anderen Server weiter. Andererseits ist es so, dass nicht jeder Server alle Newsgroups beherbergt. Außerdem benötigen Sie zum Lesen von News ein spezielles Programm, einen so genannten Newsreader. Bei den meisten Browsern ist dieses Programm integriert (oder kann zusätzlich heruntergeladen werden), aber wenn dieses Programm fehlt oder nicht korrekt konfiguriert ist, geht der Link ins Leere.

Mail-Links

Internetzeitschriften berichten immer wieder: Viele Firmen machen es dem Besucher ihrer Webseiten sehr schwer, mit ihnen in Kontakt zu treten. Dabei wäre es doch so einfach!

Ein Mail-Link funktioniert folgendermaßen: Sobald der Benutzer darauf klickt, wird das Standardmailprogramm seines Systems gestartet und er kann anfangen zu tippen. Hier die einfachste Form des Links:

```
<a href="mailto:support@pearson.de">Mail an  
M+T Support</a>
```

Durch Klick auf diesen Link öffnet sich ein Mailfenster und als Empfänger ist *support@pearson.de* voreingestellt (siehe Abbildung). Sie können so dem Support von Markt+Technik eine kurze (oder lange) Nachricht zukommen lassen.

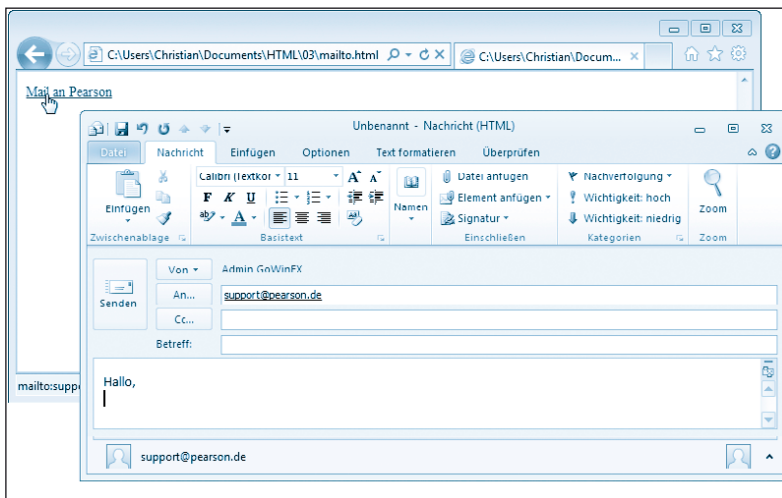


Abbildung 3.7: Das Mailfenster mit vorgegebenem Empfänger

Sie können mehrere Empfänger angeben, wenn Sie diese durch Kommata voneinander trennen:

```
<a href="mailto:support@pearson.de,info@pearson.de">
```

```
Mail an mehrere</a>
```

Microsoft-Mailprogramme verlangen Strichpunkte als Trennzeichen zwischen den E-Mail-Adressen, womit aber viele andere Mailprogramme überhaupt nicht zurechtkommen. Deswegen sollten Sie auf Kommata setzen, denn damit kommen Outlook & Co. ebenfalls zurande. Worauf Sie allerdings überhaupt keinen Einfluss haben, ist, ob die Kommunikation mit dem Mailprogramm überhaupt funktioniert. Auf manchen Systemen fehlt ein Mailprogramm, auf manchen ist es falsch konfiguriert, und unter Umständen stellt sich dem Benutzer noch eine Warnmeldung wie in der Abbildung in den Weg.

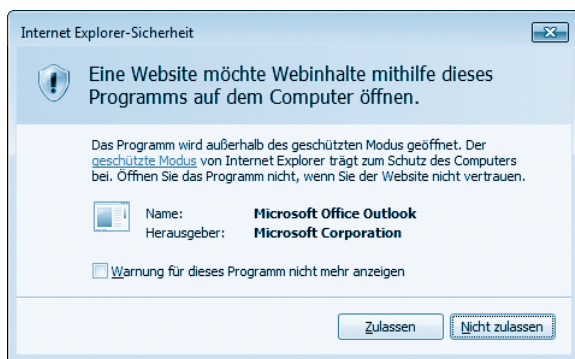


Abbildung 3.8: Sicherheitsabfrage vor dem Öffnen des Mailfensters

Nun noch ein paar Worte zu »inoffiziellen Tricks«. Mail-Links bieten noch weitere Möglichkeiten, die in der offiziellen Spezifikation zwar nicht vorgesehen sind, aber dennoch in den meisten Browsern funktionieren. Sie können an die E-Mail-Adresse des Empfängers noch mehrere weitere Optionen nach folgendem Muster anhängen:

```
<a href="mailto:support@pearson.de?option1=abc&option2=def&option3=ghi">
Mail </a>
```

Ihnen stehen die folgenden Optionen zur Verfügung:

- `bcc=` – fügt einen Blindkopie-Empfänger in die Mail ein
- `cc=` – fügt einen Kopie-Empfänger in die Mail ein
- `subject=` – setzt den Betreff der Mail
- `body=` – setzt den Text der Mail

Hier ein etwas ausführlicherer Link, der all diese Möglichkeiten nutzt:

```
<a href="mailto:support@pearson.de?cc=info@pearson.de&bcc=meine.mutter@xy.de&subject=EASY HTML&body=Ein tolles Buch!">Mail an Pearson</a>
```

Und wie Sie der Abbildung entnehmen können – es funktioniert! Allerdings nicht bei jedem Browser und es muss natürlich beim Benutzer ein Mailprogramm vorhanden sein. Die wichtigste Angabe ist und bleibt die E-Mail-Adresse des Empfängers, der Rest ist als nettes Beiwerk zu betrachten.

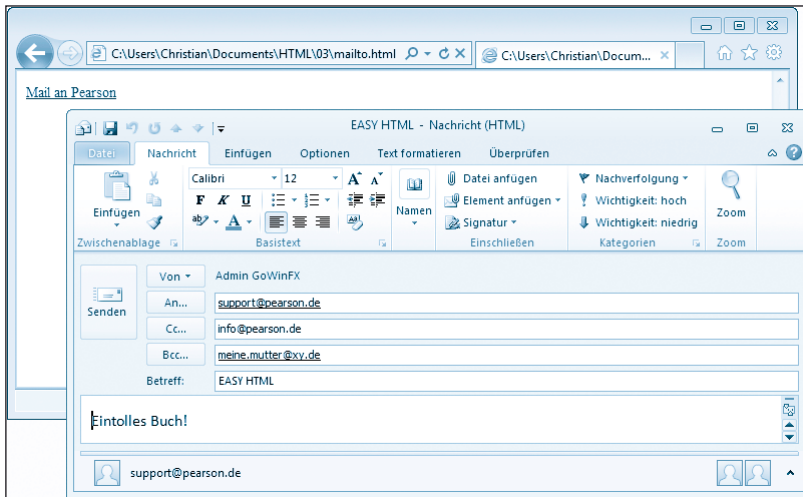


Abbildung 3.9: Das Mailfenster ist vorausgefüllt.

Downloads

Wenn Sie Software oder andere Dateien zum Herunterladen (Download) anbieten wollen, können Sie die gewohnte Syntax verwenden. Sie verweisen also nicht auf eine HTML-Seite, sondern auf die entsprechende Datei. Je nach Konfiguration des Browsers wird die Datei dann direkt im Browser (beispielsweise PDF-Dateien) oder in der dazugehörigen Applikation (beispielsweise Textverarbeitungsdokumente) geöffnet oder der Browser fordert Sie dazu auf, einen Speicherort für die Datei auf Ihrer Festplatte anzugeben. Für den Link selbst können Sie eine beliebige der hier bereits vorgestellten Methoden verwenden. In der Abbildung sehen Sie ein mögliches Download-Dialogfenster am Beispiel des Firefox-Browsers.

```
<a href="downloads/datei.zip">datei.zip</a>
```

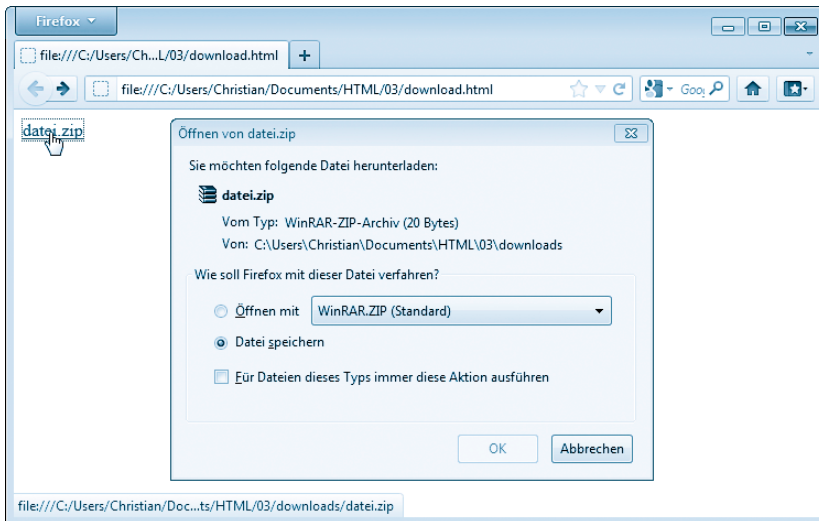


Abbildung 3.10: Die Aufforderung zum Speichern oder Ausführen (im Firefox)

Textmarken

Sie können nicht nur auf HTML-Seiten verlinken – sondern auch auf bestimmte Bereiche einer Seite. Oft sehen Sie auf Webseiten Folgendes: Ganz unten befindet sich ein Link »nach oben«, der auf geradezu wundersame Art und Weise an den Anfang der Seite scrollt. Das geht ganz ohne Magie, nur mit ein wenig HTML.

Sie können sich eine Textmarke wie eine Positionsmarkierung auf einer HTML-Seite vorstellen. Diese Markierung wird im Browser nicht explizit angezeigt, aber es ist klar, an welcher Stelle sie sich befindet. Sie setzen eine solche Markierung wie folgt:

```
<a name="oben">Irgendein Text</a>
```

Damit ist die Textmarke definiert. Die Stelle darf nicht leer sein, auch wenn wir später ein Gegenbeispiel präsentieren werden. Um eine Textmarke ganz oben auf einer Seite zu definieren, verwenden Sie doch als Text die Überschrift oder setzen Sie ein (kaum sichtbares) Leerzeichen innerhalb des <a>-Elements.

Der Name einer Textmarke muss die folgenden Bedingungen erfüllen, damit das Ganze auch in allen Browsern funktioniert:

- Er darf nur aus den Buchstaben a–z bzw. A–Z bestehen.
- Er sollte insbesondere keine Leerzeichen enthalten (das mögen einige Browser, insbesondere ältere Versionen, überhaupt nicht).

- Erst recht sollten Sonderzeichen wie beispielsweise Umlaute vermieden werden.

Um nun auf eine Textmarke zu verweisen, verlinken Sie einfach darauf! Um klarzumachen, dass es sich um eine Textmarke handelt und nicht um eine Datei, stellen Sie dem Namen der Textmarke ein Doppelkreuz (#) voran:

```
<a href="#oben">nach oben</a>
```

In folgendem Listing sehen Sie das Ganze einmal an einem etwas realitäts-näheren Beispiel. Auf der Seite befindet sich viel Text (zugegeben, er wurde ein wenig gestreckt), sodass die Überschrift nicht mehr sichtbar ist, wenn man am Ende des Textes angekommen ist (zugegeben, bei einem kleinen Browserfenster). Ein Klick auf den Link springt zur Textmarke und damit zurück nach oben:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

```
<html>
```

```
<head>
```

```
  <title>Textmarken</title>
```

```
</head>
```

```
<body>
```

```
<a name="oben">
```

```
<h2>Wilhelm Meisters Wanderjahre</h2>
```

```
</a>
```

```
<p>
```

```
  Der Mensch ist ein beschr&auml;nktes Wesen. <br />
```

```
  Unsere Besch&auml;nkung zu &uuml;berdenken, <br />
```

```
  ist der Sonntag gewidmet. Sind es <br />
```

```
  k&ouml;rperliche Leiden, die wir im <br />
```

```
  Lebenstaumel der Woche vielleicht gering <br />
```

```
  achteten, so m&uuml;ssen wir am Anfang der <br />
```

```
  neuen alsobald den Arzt aufsuchen. <br />
```

```
  Ist unsere Beschr&auml;nkung &ouml;konomisch<br />
```

```
  und sonst b&uuml;rgerlich, so sind unsere <br />
```

```
  Beamten verpflichtet, ihre Sitzungen zu <br />
```

```
  halten. Ist es geistig, sittlich, was uns <br />
```

```
  verd&uuml;stert, so haben wir uns an einen <br />
```

```

Freund, an einen Wohldenkenden zu wenden, <br />
dessen Rat, dessen Einwirkung zu erbitten: <br />
Genug, es ist das Gesetz, dass niemand eine <br />
Angelegenheit, die ihn beunruhigt oder <br />
qu&uuml;lt, in die neue Woche <br />
hin&uuml;bernehmen d&uuml;rft.
</p>
<p><a href="#oben">nach oben</a></p>
</body>
</html>

```

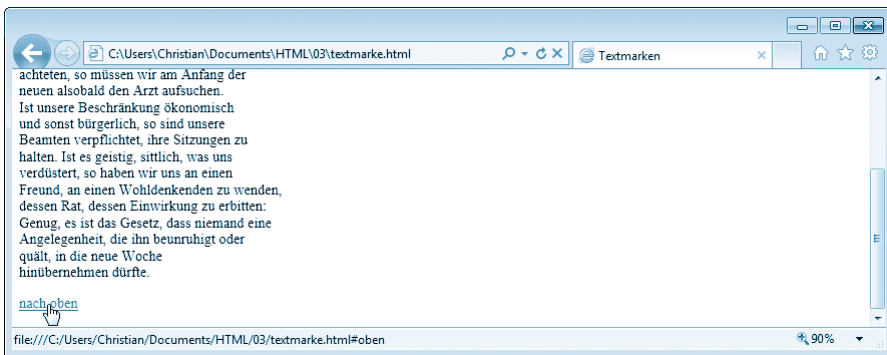


Abbildung 3.11: Der Link unten springt ...



Abbildung 3.12: ... wieder nach oben!

Längere Seiten lassen sich so untergliedern. Am Anfang der Seite stehen die Kernpunkte der Seite. Jeder Punkt ist mit dem entsprechenden Abschnitt auf der Seite verlinkt und kann so direkt angesprungen werden. Ein Beispiel hierfür sind zahlreiche Einträge in der Online-Enzyklopädie Wikipedia (siehe Abbildung unten, hier am Beispiel der deutschen Seite zu HTML). In dem Kasten oben auf der Seite stehen die wichtigsten Punkte, die per Textmarke direkt angesprungen werden können.



Abbildung 3.13: Die Wikipedia-Seite zu HTML (natürlich keine Konkurrenz zum Buch ;-))

Hier ein (angepasster und verkürzter) Auszug aus dem Code:

```
<a href="#Entstehung">1 Entstehung</a>
<a href="#Syntax">2 Syntax</a>
<a href="#Sprachtyp">3 Sprachtyp</a>
<a href="#Versionen">4 Versionen</a>
```

Betrachten wir den zweiten Link auf *2 Syntax*. Dieser verweist auf eine Textmarke namens *Syntax*, und tatsächlich, etwas weiter unten im Code finden wir:

```
<span class="mw-headline" id="Syntax">Syntax</span>
```

Wie Sie sehen, werden Anker in modernen Seiten nicht mehr mittels dem `<a>`-Tag, sondern mittels anderer Tags und dem `id`-Attribut definiert. Dies erlaubt die Namensvergabe über ein einheitliches Attribut (`id`).

Sie können auch von anderen Seiten aus direkt auf Textmarken verlinken. Verwenden Sie dazu die Techniken, die wir in diesem Kapitel bis jetzt vorgestellt haben, und hängen Sie an den Dateinamen # und den Namen der Textmarke an. Folgender Link führt beispielsweise direkt zu der eben betrachteten Textmarke auf Wikipedia:

```
<a href="http://de.wikipedia.org/wiki/HTML#Sprach-
typ">Infos zum Sprachtyp von HTML bei Wikipedia</a>
```

Zielfenster

An dieser Stelle wollen wir noch kurz auf einen weiteren Punkt eingehen, der in *Kapitel 11* weiter ausgeführt wird. Normalerweise werden Links im aktuellen Fenster geöffnet, Sie haben aber die Möglichkeit, die Links auch in einem anderen Fenster zu aktivieren. Dazu müssen Sie das `target`-Attribut des `<a>`-Tags setzen. Sie haben zunächst zwei Möglichkeiten:

- `"_self"` – Hiermit öffnen Sie das Ziel des Links im aktuellen Fenster (Standardeinstellung, muss nicht angegeben werden).
- `"_blank"` – Hiermit öffnen Sie das Ziel des Links in einem neuen Fenster.
- `"beliebig"` – öffnet das Ziel des Links auch in einem neuen Fenster, außer der Name *beliebig* wurde bereits einmal verwendet, dann in diesem Fenster.

Sie sehen die beiden Möglichkeiten im Überblick: zunächst einmal die Option `_blank`:

```
<a href="http://www.pearson.de/" target="_blank">
  Pearson Education</a><br />
<a href="http://www.mut.de/" target="_blank">
  Markt+Technik</a><br />
<a href="http://www.addison-wesley.de/" target="_blank">
  Addison-Wesley</a>
```

Egal, wie oft Sie auf welchen Link klicken, jedes Mal öffnet sich ein neues Fenster.

Anders ist es in folgendem Fall:

```
<a href="http://www.pearson.de/" target="neu">
  Pearson Education</a><br />
<a href="http://www.mut.de/" target="neu">
```

```

    Markt+Technik</a><br />
<a href="http://www.addison-wesley.de/" target="neu">
    Addison-Wesley</a>

```

Beim ersten Klick auf einen Link öffnet sich ein neues Fenster; alle weiteren Klicks auf diesen Link öffnen die Seite ebenfalls in diesem neuen Fenster, denn der Fensternamen *neu* ist mit dem geöffneten Fenster assoziiert.

Sie können automatisch alle Links in einem bestimmten Fenster öffnen, wenn Sie in den `<head>`-Abschnitt der HTML-Seite `<base target="neu" />` schreiben. Das gilt dann für alle Links, außer Sie geben bei einem Link das `target`-Attribut explizit an.

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html>
<head>
    <title>Links</title>
    <base target="neu" />
</head>
<body>
<h3>Eine Auswahl von Links</h3>
<a href="http://www.pearson.de/" target="_self">
    Pearson Education</a><br />
<a href="http://www.mut.de/">Markt+Technik</a><br />
<a href="http://www.addison-wesley.de/">Addison-Wesley</a>
</body>
</html>

```

Zur Analyse des Codes: Durch `<base target="neu" />` werden die Ziele aller Links in einem neuen Fenster namens *neu* geöffnet. Eine Ausnahme ist der Link auf Pearson, denn der wird im aktuellen Fenster geöffnet; `target="_self"` überschreibt die Vorgabe durch `<base target />`. In Browsern, die mehrere Fenster über Registerkarten (Tabs) verwalten, werden neue Fenster in einem neuen Register geöffnet.

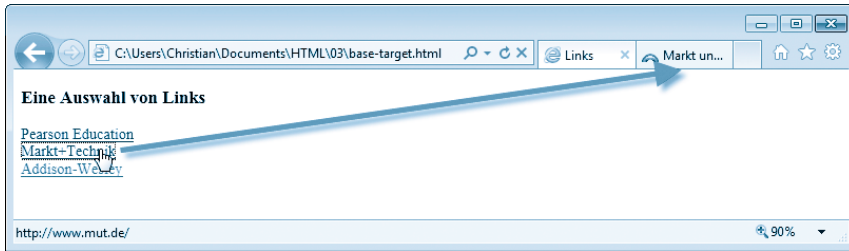


Abbildung 3.14: (Fast) alle Links werden im neuen Fenster geöffnet.

Rechtliches

Normalerweise sollte sich der Adressat eines Links geehrt fühlen, wenn jemand auf ihn verweist, also damit quasi Werbung für ihn macht. Dennoch gibt es Fälle, in denen ein Link nicht gern gesehen wird:

- im Zusammenhang mit negativen Äußerungen (»Das nicht funktionierende Gerät habe ich bei XY gekauft«) nebst Link auf die Homepage von XY)
- Links mitten in ein Angebot hinein (so genannte *deep links*, also nicht auf die Hauptseite), insbesondere wenn der Websitebetreiber von Werbeschaltungen lebt und deswegen ein Interesse daran hat, dass auch seine Hauptseite aufgerufen wird
- Links auf Seiten mit verwerflichem Inhalt

Die Rechtslage ist nicht immer eindeutig. Viele Websitebetreiber behelfen sich damit, folgende Erklärung abzugeben: »Externer Link; wir sind nicht für die Inhalte einer externen Website verantwortlich.« Und auch wenn die Wirksamkeit einer solchen Klausel rechtlich umstritten ist, können Sie sie in Erwägung ziehen. Und: Holen Sie sich wenn möglich eine Erlaubnis, wenn Sie auf ein fremdes Angebot linken!

Fragen

1. Was ist der Unterschied zwischen einem Link auf `"/presse"` und `"presse/"`?
2. Erstellen Sie einen Verweis auf die Textmarke namens `"top"` auf der Seite `http://www.hauser-wenz.de/!`



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>