Löwenstark Digital Group GmbH
Petzvalstr. 38
38104 Braunschweig
Tel.: 0531 21360 5500
Fax: 0531 21360 5795
kontakt@loewenstark.com

HTML Formulare

Stand: 13.10.2022

Formulare
Auswahllisten und Drop-down-Listen

HTML bietet ein Element für Listenauswahl in **Formularen** an: das `select`-Element. Ob es sich um eine längere Liste mit mehreren sichtbaren Einträgen handelt oder um eine Dropdown-Liste, die der Anwender aufklappen kann, bestimmen
Sie durch ein Attribut.

Info Die Liste wird in `<select>...</select>` eingeschlossen. Die einzelnen Listeneinträge werden durch `<option>...</option>` markiert. Der Text zwischen `<option>`und `</option>` ist der Text, den der Anwender als Listeneintrag angeboten bekommt. Der Wert des Attributs `value=` ist dagegen der Wert, der beim Absenden der Formulardaten als der ausgewählte Listenwert an das verarbeitende
Script übertragen wird. Das einleitende `<select>`-Element erhält bei `name=` den Namen des Parameters, unter dem die Listenauswahl an das verarbeitende Script übergeben wird. Das Attribut `size=` bestimmt das Aussehen
der Liste. Weisen Sie als Wert 1 zu, ist das Ergebnis eine Dropdown-Liste. Weisen Sie einen höheren Wert zu, ist das Ergebnis eine Liste mit so vielen sichtbaren Einträgen wie im Wert angegeben. Enthält die Liste mehr Einträge als bei `size=`angegeben, kann der Browser dem Anwender automatisch eine Scrollleiste anzeigen.

Das obige Beispiel zeigt noch eine andere Möglichkeit:

Auswahllisten können eine Default-Auswahl enthalten. Eines der notierten `option`-Elemente können Sie durch das Standalone-Attribut `selected` als den vorausgewählten Codeeintrag bestimmen. Wenn Sie mit XHTML arbeiten, notieren Sie
`selected="selected"`. In vielen Fällen ist es auch sinnvoll, einen Eintrag anzubieten, der keinen Nutzwert hat und stattdessen zur Auswahl auffordert, so wie im Beispiel das erste notierte `option`-Element. Für längere Listen mit
logisch gruppierbaren Einträgen bietet **HTML** die Möglichkeit an, Listeneinträge zu Gruppen zusammenzufassen.

Info Die Gruppierung wird durch Einfügen von `optgroup`-Elementen erreicht. Zwischen `<optgroup>` und `</optgroup>` werden die jeweils zur Gruppe gehörigen Listeneinträge wie gewohnt mithilfe von `option`-Elementen
notiert. Das einleitende `<optgroup>`-Tag erhält über das Attribut `label=` den Wert, der als Gruppenüberschrift angezeigt werden soll.

Bei manchen Listen kann es auch sinnvoll sein,

Löwenstark Digital Group GmbH
Petzvalstr. 38
38104 Braunschweig
Tel.: 0531 21360 5500
Fax: 0531 21360 5795
kontakt@loewenstark.com

dem Anwender die Auswahl mehrerer Einträge zu erlauben. HTML-seitig ist das kein Problem. Im einleitenden `<select>`-Tag wird einfach das Standalone-Attribut `multiple` (in XHTML: `multiple="multiple"`) eingefügt.
Problematischer ist es, dass der Anwender nicht weiß, dass er mehrere Einträge auswählen kann, und meistens auch gar nicht, wie. Bei expliziter Mehrfachauswahl ist es also zweckmäßig, im umgebenden Text des Elements auf die Möglichkeit der Mehrfachauswahl
hinzuweisen. Da viele Anwender auch nicht wissen, dass sie mehrere Einträge durch Anklicken bei gedrückter (`Strg`)-Taste auswählen können, kann auch dazu ein entsprechender Hinweis im Text nicht schaden.

Formular-Eingaben versenden

Um die Formulareingaben zu versenden, muss das eigentliche `form`-Tag noch mit einer `action` verbunden werden: Wenn der Ausfüller auf den Button Senden klickt, soll ein **CGI-Script** die Daten als E-Mail versenden.
Für die Wertzuweisung `form action="` konsultieren Sie Ihren Provider bzw. überprüfen Ihre Unterlagen. Hier steht in der Regel die Adresse des CGI-Scripts. z.B.:

```
<form action="http://www.Ihre-Domain.de/cgi-bin/mailer.pl">
```

Info Ebenfalls abhängig davon, welches Script Sie einsetzen, müssen Sie als zusätzlichen Parameter die Methode des Versendens eintragen: `method="get"` oder `method="post"`. Auch hier gilt: Überprüfen Sie, was Ihr Provider
bzw. dessen CGI erwartet. In unserem Fall `method="POST"`. Weiterhin benötigt das komplette Formular noch einen Namen: `name="Mail"`. Beachten Sie bitte die Hinweise, exakte Pfadangabe, manche Scripte sind `case-sensitiv`und unterscheiden Gross-Kleinschreibung.

```
<formaction="http://www.Ihre-Domain.de/cgi-bin/mailer.pl"name="Mail"
method="POST">
.
.
</form>
```

Info Alle Formularmailer erwarten zusätzlich Angaben. Bislang wurde dem CGI-Script noch nicht mitgeteilt, an wen die Einträge versandt werden sollen bzw. aus welchem **Formular** heraus die Daten versandt wurden. Nahezu alle Formmailer
ermöglichen die Ausgabe einer Bestätigungsseite, ein HTML-Dokument, dass nach erfolgreichem Senden am Browserbildschirm erscheint. Diese und weitere Angaben werden über versteckte Eingabefelder realisiert: `<input type=hidden>`. Mit dem
Wert `hidden` wird das Feld nicht am Bildschirm dargestellt. Von Ihrem Provider werden einige dieser Felder zwingend vorgeschrieben. In unserem Fall: Die Empfänger(`recipient`)-Adresse:

```
<Input type=hidden name="recipient" value="webmaster@Ihre-Domain.de">
```

Löwenstark Digital Group GmbH
Petzvalstr. 38
38104 Braunschweig
Tel.: 0531 21360 5500
Fax: 0531 21360 5795
kontakt@loewenstark.com

Damit wird festgelegt, wer die E-Mail erhalten soll. Das Thema, Titel des Formulars:

```
<Input type=hidden name="subject" value="HTML-Aufbaukurs">
```

Diese Angabe erscheint in der Betreff-Zeile der E-Mail. Die Adresse der Bestätigungsseite (redirect):

```
<Input type=hidden name="redirect"
value="http://www.Ihre-Domain/html/dankemail.html">
```

Info Unter `redirect` stellen Sie sich die HTML-Datei vor, auf die der Besucher dirigiert wird, nachdem er das Formular ausgefüllt hat. Sie können jederzeit zusätzliche versteckte (`hidden`) Felder einfügen. Diese werden nicht
am Bildschirm dargestellt, aber in der E-Mail, die Sie erhalten, aufgelistet. Das komplette Formular:

```
<formaction="http://www.Ihre-Domain.de/cgi-bin/mailer.pl"name="Mail"
method="POST">
<input type="hidden" name="redirect"
value="http://www.Ihre-Domain.de/html/dankemail.html">
<inputtype="hidden" name="recipient" value="webmaster@Ihre-Domain.de">
<inputtype="hidden" name="subject" value="Anfrage aus Homepage">
<p>
Name: <input type="text" size="15" name="name" maxlength="50">
<br>
Vorname: <input type="text" size="15" name="vorname" maxlength="50">
<br>
PLZ: <input type="text" size="5" name="plz" maxlength="5">
<br>
<textarea name="nachricht" cols="30" rows="5" >
</textarea>
<br>
<input type="submit" value="Senden">
<input type="reset" value="Löschen">
</form>
```

<formaction=“http: www.ihre-domain.de=““ cgi-bin=““ mailer.pl“name=“Mail“ method=“POST“>
<inputtype=“hidden“ name=“recipient“ value=“webmaster@Ihre-Domain.de“> <inputtype=“hidden“
name=“subject“ value=“Anfrage aus Homepage“> Name: Vorname: PLZ:

```
<br /><br /><br /><br /><br
/><br /><br /><br /><br /><br
/><br /><br /><br /><br /><br
/><br /><br /><br /><br /><br
/><br /><br /><br /><br /><br
```

</inputtype=“hidden“></inputtype=“hidden“></formaction=“http:>
Info Testen Sie das Formular, geben Sie Zeichen ein. Der Versand des Formulars klappt nicht, die Funktion

Löwenstark Digital Group GmbH
Petzvalstr. 38
38104 Braunschweig
Tel.: 0531 21360 5500
Fax: 0531 21360 5795
kontakt@loewenstark.com

Löschen funktioniert allerdings. Sie können die Textarea und die Inputfelder vorbelegen, d.h. die Felder sind nicht leer. Für eine Textarea
notieren Sie den Text zwischen Anfang- und Ende-Tag: `<textarea>` Text im Feld `</textatrea>`. Eingabefeldern weisen Sie mit dem Attribut `value="Vorgabe"` einen Wert zu. Um ein **Eingabefeld**mit Passwortcharakter zu erzeugen, setzen Sie das Attribut `type="password"`. Anstatt des eingegebenen Textes erscheinen dann **** am Bildschirm. Achtung: Das ist kein echtes Passwort, sondern nur eine geschützte Darstellung am Bildschirm.
Die Daten werden trotzdem unverschlüsselt übertragen. Sie könnten Eingabefelder auch sperren (nur lesbar) mit `<input readonly>`. Das bewirkt, dass zwar ein Eingabefeld erscheint, man kann aber nichts eingeben. Wozu? Eingabefelder sind
bereits unschön, wenn Sie etwas zu sagen haben, schreiben Sie es in Ihr Dokument.

Felder für Datei-Upload

Diese Sorte Formularelement erlaubt dem Anwender, eine Datei von seinem lokalen Rechner zusammen mit dem Formular zu übertragen. Wenn ein CGI-Script die ankommenden **Formulardaten** auf dem Server-Rechner verarbeitet, ist es dadurch möglich,
dem Anwender das Uploaden (Hochladen) von Dateien auf den Server-Rechner zu ermöglichen.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
<title>Felder f&uuml;r Datei-Upload definieren</title>
</head>
<body>
<h1>Schicken Sie uns was Schickes!</h1>
<form action="input_file.htm" method="post" enctype="multipart/form-data">
<p>Wählen Sie eine Textdatei (txt, html usw.) von Ihrem Rechner aus:
<br>
<input name="Datei" type="file" size="50" maxlength="100000"
accept="text/*"></p>
</form>
</body>
</html>
```

Info Mit `<input type="file">` definieren Sie ein Element für Datei-Upload (input = Eingabe, file = Datei). Der Web-Browser sollte dann ein Eingabefeld anzeigen, das die Eingabe einer Datei (in den meisten Fällen mit Pfadnamen)
erlaubt. Rechts daneben sollte der Browser einen Button anzeigen, bei dessen Anklicken ein lokaler Dateiauswahl-Dialog am Bildschirm erscheint. Die Größe des Eingabefeldes (Anzahl Zeichen) können Sie mit dem Attribut `size=` bestimmen (size
= Größe). Das Attribut `maxlength=` wurde in HTML 3.2 definiert als Hinweis an den Browser, nur Dateien bis zu einer Bytegröße dieser Angabe zur Auswahl zuzulassen. In HTML 4.01 wird auf diese Funktion nicht mehr eingegangen. Verlassen Sie
sich also nicht auf diese Angabe. Sicherer ist es, bei der Weiterverarbeitung mit einem **CGI-Script** im

Löwenstark Digital Group GmbH
Petzvalstr. 38
38104 Braunschweig
Tel.: 0531 21360 5500
Fax: 0531 21360 5795
kontakt@loewenstark.com

Script die Dateigröße zu ermitteln und das Script davon abhängig entscheiden zu lassen, ob die Datei akzeptiert oder verworfen wird.
Uns ist auch kein Browser bekannt, der diese Angabe in irgendeiner Weise einschränkend umsetzt.

Wenn Sie nur bestimmte Dateitypen zulassen wollen,

können Sie mit der Angabe von `accept=` die erlaubten Dateitypen eingrenzen (accept = akzeptieren). Hinter dem Istgleichzeichen können Sie einen **MIME-Typ** angeben. Dabei ist auch das Wildcard-Zeichen (*) bei Subtypen
erlaubt. Im obigen Beispiel werden mit `text/*` alle Textdateien akzeptiert. Dazu gehören reine Textdateien (*.txt*), aber z.B. auch HTML-Dateien (*.html*,*.htm*). Verlassen Sie sich aber auch bei dieser Angabe nicht
darauf, dass der Browser vor dem Versenden des Formulars das tatsächlich prüft. Auch bei dieser Angabe ist uns kein Browser bekannt, der die Auswahl des Benutzers in irgendeiner Form kontrolliert.
Info Datei-Uploads funktionieren nur mit `method="post"`. Wichtig ist außerdem, dass Sie im einleitenden `<form>`-Tag die Angabe `enctype="multipart/form-data"` notieren. Andernfalls erhalten Sie lediglich den Dateinamen der ausgewählten Datei übermittelt, nicht jedoch die Datei selbst. Das Attribut `value=` in Verbindung mit `<input type="file">`, mit dem das Vorbelegen der Dateiauswahl möglich wäre, wird von aktuellen Browsern
aus Sicherheitsgründen nicht unterstützt.

Optische Verfeinerung von Formularen mit CSS

CSS ist durchaus auch auf Formularelemente anwendbar. Bei Eingabefeldern, aber auch bei Auswahllisten und Schaltflächen lässt sich mit der CSS-Eigenschaft `width` eine gewünschte Breite festlegen. Bei `textarea`-Elementen für mehrzeilige
Eingaben kann die Höhe wie erwartet durch `height` bestimmt werden. Möglich ist jedoch noch viel mehr: angefangen von eher ästhetischer Beeinflussung wie rahmenlose Eingabefelder, farbige Feldhintergründe oder **Schriftformatierung** bei Eingabefeldern bis hin zu sinnigen Möglichkeiten wie der rechtsbündigen Ausrichtung von Eingabefeldern für numerische Werte bleiben kaum Wünsche offen.

Da die bekannten Browser zum Rendern von Formularelementen teilweise auf Betriebssystemressourcen zurückgreifen, sind den Gestaltungsmöglichkeiten auf dieser Ebene jedoch auch Grenzen gesetzt. So kann es passieren, dass die Unterdrückung des Rahmens bei
Auswahllisten nicht funktioniert oder eine Checkbox einfach keine andere Hintergrundfarbe annehmen will. Ein Beispiel soll einige der Möglichkeiten demonstrieren:

```
form {
    background - color: rgb(255, 204, 153);
    padding: 10 px;
}
input.text {
    border: none;padding: 3 px;
    background - color: rgb(255, 255, 153);
```

Löwenstark Digital Group GmbH
Petzvalstr. 38
38104 Braunschweig
Tel.: 0531 21360 5500
Fax: 0531 21360 5795
kontakt@loewenstark.com

```
    font - family: Times,
    serif;
    font - size: 17 px;
    font - weight: bold;
}
```

Das `form`-Element erhält eine orangefarbene Hintergrundfarbe und einen Innenabstand von 10 Pixel zum Inhalt. Für Eingabefelder, die das Attribut `class="text"` haben, wird festgelegt, dass der Rahmen unterdrückt wird. Zum Eingabetext
soll ein Innenabstand von 3 Pixel eingehalten werden. Das Feld erhält eine gelbe Hintergrundfarbe. Vom Anwender eingegebener Text soll in Times oder einer anderen Serifenschrift mit einer Schriftgröße von 17 Pixel und fett formatiert werden.

HTML-Formulare

Webseiten mit der Funktion „Webpräsenz“ sind nur eine mögliche Variante. Vieles, was das Web attraktiv macht, fällt eher unter den Begriff „Webanwendung“. Der „Seitenbesucher“ wird zum „User“, der Suchen startet, Beiträge in Foren schreibt, persönliche
digitale Fotoalben verwaltet, Kleinanzeigen aufgibt usw. Auch dabei basieren die Webseiten auf **HTML**.

Die nötige Interaktion mit dem Anwender beginnt in der Regel mit einem Formular. HTML stellt eine Reihe von Standard-Formularelementen zur Verfügung, die sich an den Dialogelementen moderner grafischer Benutzeroberflächen orientieren. So gibt es Eingabefelder,
Radiobuttons, Checkboxen, Auswahllisten, Schaltflächen usw.

> Mit dem Formular allein ist es jedoch nicht getan.

Daten, die ein Anwender in einem Formular bestimmt und absendet, müssen irgendwie verarbeitet werden. Das Instrument dafür sind serverseitige Scripts, beispielsweise PHP-Scripts. Richtige Anwendungen für Formulare werden wir deshalb auch erst im PHP und
MySQL finden. Im vorliegenden Abschnitt konzentrieren wir uns auf die Strukturierungs- und Gestaltungsmöglichkeiten von Formularen auf Webseiten.

Formularbereiche

Formulare werden in HTML durch `<form...>...</form>` markiert. Im einleitenden `<form>`-Tag werden wichtige Informationen notiert, z.B. was passieren soll, wenn der Anwender das Formular
absendet. Zwischen `<form>` und `</form>` können Formularelemente notiert werden, aber auch Text und andere Elemente, die dazu beitragen, das Formular zu strukturieren. Betrachten wir zunächst das einleitende `<form>`-Tag
in einem Beispiel:

```
<form action="/index.php" method="post">
```

Löwenstark Digital Group GmbH
Petzvalstr. 38
38104 Braunschweig
Tel.: 0531 21360 5500
Fax: 0531 21360 5795
kontakt@loewenstark.com

```
<!-- Formularinhalt -->
</form>
```

In diesem Beispiel wird bestimmt, dass die im Formular eingegebenen oder ausgewählten Daten beim Absenden an ein PHP-Script namens index.php, das sich im Wurzelverzeichnis des Webservers befindet, übertragen werden. Zuständig für die Verarbeitungsadresse
ist das Attribut `action=`, das als Wert einen URI erwartet. Es kann sich also um eine beliebige geeignete Internetadresse oder um ein lokal referenziertes Script handeln. Erlaubt sind auch Angaben des Typs `mailto:jemand@irgendwo.xy`.
In diesem Fall sollte im `<form>`-Tag unbedingt zusätzlich `enctype="text/plain"` notiert werden. Der Browser versucht dann, die Formulardaten über eine lokale Mail-Schnittstelle an die angegebene Mail-Adresse zu senden. Sicherlich
ist diese Form der Formularverarbeitung die einfachste. Als Seitenanbieter gibt man einfach seine eigene Mail-Adresse in dieses Schema ein und schon bekommt man alle **Formulardaten**, die Anwender absenden, als E-Mail zugeschickt. Allerdings
ist von dieser Praxis dringend abzuraten. Der Mail-Versand funktioniert nämlich bei vielen Anwendern nicht, da Browser normalerweise keine eigene Schnittstelle für den Mailversand besitzen und die erfolgreiche Kommunikation der Formulardaten vom Browser
an ein Mail-Programm von vielen vagen Systemfaktoren abhängt. Bei betroffenen Anwendern wird Ihr Formular also für Ärger sorgen, da sie ein Formular, für dessen Ausfüllen sie möglicherweise einige Zeit investiert haben, am Ende nicht absenden können.
Als Wert für das `action`-Attribut sollte also bei allen professionellen Angeboten ein Script angegeben werden, das die eingegebenen Daten ordentlich verarbeitet. Ebenso wichtig wie das `action`-Attribut ist das Attribut `method=`.
Als Werte sind hier entweder `get` oder `post` möglich. Damit wird die Art bestimmt, wie dem datenverarbeitenden Script die Daten übergeben werden. Auf die Details werden wir erst später eingehen. Doch auf eine leicht erkennbare
Unterscheidung zwischen den Übertragungsmethoden GET und POST sei hier schon hingewiesen: Bei der GET-Methode werden die Formulardaten über den URI übertragen. In der Adresszeile des Browsers ist das daran erkennbar,
dass dort nach Absenden des Formulars, also beim Aufruf des bei `action=` angegebenen Scripts, alle Namen und Werte der Formularelemente in einer Zeichenkette erscheinen. Nach Eingabe eines Suchbegriffs in Google erscheint im Browser beispielsweise:

```
http://www.google.de/search?hl=de&q=Siebenb%C3%BCrgen&btnG=Google-Suche&meta=
```

Ein Quelltextauszug aus der Google-Seite mit dem Formular zur Suche lautet:

```
<form name="gs" method="GET" action="/search">
```

Das datenverarbeitende Script hat also den Dateinamen `search` und liegt im Wurzelverzeichnis des Webservers. Die Übertragungsmethode GET sorgt dafür, dass die eingegebenen Daten des Suchformulars im URI stehen. Bei der POST-Methode
ist dies nicht der Fall. Das Script erhält die Daten in diesem Fall über den Standard-Input-Kanal des Server-

Löwenstark Digital Group GmbH
Petzvalstr. 38
38104 Braunschweig
Tel.: 0531 21360 5500
Fax: 0531 21360 5795
kontakt@loewenstark.com

Rechners.

Welche Übertragungsmethode Sie in einem Formular angeben müssen, hängt davon ab,

welche der Übertragungsmethoden vom verarbeitenden Script ausgewertet werden. Wenn Sie das **Script** selber programmieren, können Sie die Methode selbst bestimmen. Handelt es sich um ein fremdprogrammiertes Script, muss die Schnittstelle
von Daten, die diesem Script übergeben werden können, vom Programmierer oder Anbieter ohnehin dokumentiert sein. Dazu gehört auch die vom Script erwartete Übertragungsmethode. Bei `mailto`-Formularen muss die Methode `POST` gewählt
werden. Das einleitende `<form>`-Tag kann noch weitere Attribute enthalten, z.B. das `name`-Attribut, wie im obigen Beispielformular von Google erkennbar. Dieses ist vor allem in Zusammenhang mit JavaScript von Bedeutung. über
JavaScript werden wir später darauf noch näher eingehen.

Formulare als E-Mail

Viele Anwender, die von großen Discount-Providern Speicherplatz für eigene Web-Seiten erhalten, haben keine Möglichkeit, eigene **CGI-Scripts** zur Datenverarbeitung auf dem Server-Rechner einzusetzen. Zu diesem Zweck besteht die Möglichkeit,
sich ausgefüllte Formulare als E-Mail zuschicken zu lassen.

Ein Problem stellen bei dieser Lösung jedoch Web-Browser dar, die nicht in der Lage sind,

Formulare mit `action="mailto:..."` zu versenden. Dabei klappt aber nicht unbedingt immer die Kommunikation zwischen Browser und E-Mail-Programm so, dass die Formulardaten übergeben und vom E-Mail-Programm anstandslos versendet werden. Es ist
also ein Glücksspiel, ob der Formularversand via E-Mail bei Ihren Seitenbesuchern klappt oder nicht. Um das Problem zu umgehen, können Sie einen öffentlichen CGI-Service für so genannte **Form-Mailer** in Anspruch nehmen. Dort werden die
Formulardaten von einem CGI-Programm verarbeitet und Ihnen dann als E-Mail zugeschickt. Ein anderes Problem bei `mailto`-Formularen besteht darin, dass die Formulardaten beim Abschicken per Voreinstellung nach einem bestimmten MIME-Typ kodiert
werden, dem MIME-Typ `application/x-www-form-urlencoded`. Dabei werden alle Leerzeichen, verschiedene Sonderzeichen und Umlaute durch spezielle Zeichenfolgen ersetzt. So lautet beispielsweise der Satz Danke für die Hilfe! nach der
Umwandlung: `Danke+f%FCr+die+Hilfe%21`. Für Menschen ist das ziemlich ungenießbarer Lesestoff. Um die Kodierung zu verhindern, können Sie im einleitenden `<form>`-Tag zwar die Angabe `enctype="text/plain"` angeben.
Von Anwendern, deren Web-Browser diese Angabe jedoch nicht interpretiert, werden Sie dennoch kodierte Formulardaten erhalten.

Löwenstark Digital Group GmbH
Petzvalstr. 38
38104 Braunschweig
Tel.: 0531 21360 5500
Fax: 0531 21360 5795
kontakt@loewenstark.com

Formulare und CGI

Wenn Sie beispielsweise Ihr gesamtes Web-Projekt nach einem Begriff durchsuchbar machen wollen, können Sie dem Anwender ein Formular anbieten, in dem er den Suchbegriff eingeben kann. Nach dem Abschicken des Formulars muss ein Script oder Programm auf
dem Server-Rechner alle Dateien des Projekts nach dem gewünschten Begriff oder einen zuvor erzeugten Index oder eine Datenbank absuchen und eine dynamische HTML-Datei erzeugen, in der Verweise auf alle Seiten enthalten sind, die den Suchbegriff enthalten.

> Die Standard-Programmierschnittstelle auf Web-Servern hierfür ist CGI

(Common Gateway Interface). Manche Web-Server haben – je nach installierter HTTP-Software – auch noch andere Programmierschnittstellen. Der Provider, bei dem Sie Ihre Web-Seiten speichern, muss Ihnen Zugriff auf diese Programmierschnittstelle erlauben,
damit Sie solche Anwendungen selbst realisieren können. Viele Discount-Provider erlauben ihren Mitgliedern jedoch keinen solchen Zugriff. Die meist verwendete Programmiersprache für CGI-Scripts ist Perl. Wenn Sie in Perl programmieren wollen, fragen Sie
den Provider, ob der Server über einen Perl-Interpreter verfügt, und wo genau der Interpreter auf dem Server liegt.

Versand per E-Mail

Es lassen sich alle Arten von Formularen erstellen, kurze und lange. Hier ist ein solches: ein **Feedback-Formular** für Ihre Website (ohne Formatierungen, um es möglichst kurz und übersichtlich zu halten):

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<!-- feedbackformular-mail.html -->
<html>
<head>
<title>Formulare</title>
</head>
<body>
<form ...>
<input type="radio" name="Anrede" value="Herr" />
Herr
<input type="radio" name="Anrede" value="Frau" />
Frau
<br>
Vorname:
<input type="text" name="Vorname" />
<br>
Nachname:
<input type="text" name="Nachname" />
```

Löwenstark Digital Group GmbH
Petzvalstr. 38
38104 Braunschweig
Tel.: 0531 21360 5500
Fax: 0531 21360 5795
kontakt@loewenstark.com

```
<br>
E-Mail:
<input type="text" name="Mail" />
<br>
Kommentar zu meiner Website:
<textarea name="Kommentar" cols="50" rows="8">
Was ich schon immer mal sagen wollte ...
</textarea>
<br>
Woher kennen Sie diese Website?
<select name="Quelle" size="3">
<option value="such">Suchmaschine</option>
<option value="mund">Mund-zu-Mund-Propaganda</option>
<option value="tv">Fernsehwerbung</option>
<option value="sonst">Sonstiges</option>
</select>
<br>
<input type="checkbox" name="Newsletter" value="ja" />
Ja, ich möchte den Newsletter abonnieren!
<br>
<input type="image" src="grafik.gif" name="Versendegrafik" style="border:0px;"
alt="Formular versenden" />
</form>
</body>
</html>
```

Info Eine „Mund-zu-Mund-Propaganda“ gibt es eigentlich nicht, der korrekte Begriff ist „Mundpropaganda“. Das Kunstwort in diesem Beispiel haben wir in der Zwiebelfisch-Kolumne von Bastian Sick im SPIEGEL gefunden und fanden es sehr gelungen – klingt
wie Mund-zu-Mund-Beatmung -, so dass wir es hier einsetzen. Fehlt nur noch ein Punkt, im Code durch drei Punkte markiert: Wohin soll das Formular? In der Regel sollen die Daten gleich per Mail verschickt werden. Da gibt es eine gute und eine schlechte
Möglichkeit. Zunächst die schlechte. Die folgenden Angaben müssen alle im `form`-Tag ergänzt werden:

- 1. Setzen Sie die Versandmethode des Formulars auf POST: `method="post"`.
- 2. Geben Sie als Versandziel die E-Mail-Adresse des Empfängers, also meist Ihre E-Mail-Adresse, in folgendem Format an: `action="mailto:empfaenger@xy.de"`.
- 3. Sorgen Sie dafür, dass die Maildaten in einem gut lesbaren Format verschickt werden: `enctype="text/plain"`.

Manchmal scheint das zu funktionieren – die Maildaten kommen beim Empfänger an. Leider ist das jedoch nur die halbe Wahrheit. Das Ganze klappt nämlich nur, wenn der Webbrowser und das E-Mail-Programm zusammenarbeiten. Das ist aber beileibe nicht immer
der Fall. Je nach System passiert entweder nichts, oder es öffnet sich eine neue Mail, die aber nicht automatisch verschickt wird. Noch schlimmer: Selbst wenn es funktioniert, erscheint in den meisten Browsern beim Versand des Formulars eine Furcht einflößende
Warnmeldung.

Löwenstark Digital Group GmbH
Petzvalstr. 38
38104 Braunschweig
Tel.: 0531 21360 5500
Fax: 0531 21360 5795
kontakt@loewenstark.com

Info Warum ist das so? Irgendwoher muss die Website ja wissen, wie vom Client aus eine E-Mail verschickt werden kann. **HTML** macht es sich hier sehr einfach: Es delegiert die Arbeit an den Webbrowser. Und wenn der kein integriertes
Mailprogramm hat, delegiert er diese Arbeit an das Standardmailprogramm im System. Und hier ist es wie im wahren Leben: Je mehr delegiert wird, desto mehr Informationen können verloren gehen.

Versand per E-Mail, aber richtig

Doch zum Glück gibt es noch Alternativen. Beispielsweise den Anbieter Formmailer, im Web erreichbar unter Formmailer.

Die Idee dahinter
ist genauso einfach wie pfiffig. Auf dem Webserver von Formmailer liegt ein Programm, das dazu in der Lage ist, zuverlässig E-Mails zu versenden. Alles, was Sie nun tun müssen, ist, dem Formmailer mitzuteilen, wohin die E-Mails geschickt werden soll.
Das geschieht grob in zwei Schritten: Zuerst melden Sie sich bei Formmailer an, dann passen Sie Ihr Formular an:

- 1. Klicken Sie auf http://www.formmailer.com/ auf der Hauptseite in der linken Navigationsleiste auf Neues Konto.
- 2. Akzeptieren Sie die angezeigten Bedingungen für die Nutzung des Formulars.
- 3. Wählen Sie eine Benutzerkennung aus und geben Sie Ihre Anschrift an.

Sie erhalten eine Bestätigungsmeldung (oder eine Fehlermeldung, wenn etwas fehlt oder der von Ihnen gewählte Benutzername bereits vergeben ist).

- 4. Klicken Sie jetzt auf Kontozugang, wieder in der linken Navigationsleiste.
- 5. Geben Sie Ihre Nutzerdaten an, um sich einzuloggen.
- 6. Klicken Sie auf den Link Neues Formular einrichten.
- 7. Geben Sie im Formular drei Daten an: den URL des Formulars, den URL der Ergebnisseite sowie die E-Mail-Adresse, an die das Formular verschickt werden soll.

Die Ergebnisseite wird angezeigt, nachdem das Formular versendet wurde. Hier kann auch beispielsweise schlicht „Danke“ stehen. Ganz wichtig: Unter Verwendungszweck geben Sie an, ob Sie das Formular privat oder geschäftlich nutzen. Nur im ersteren Fall
ist Formmailer gratis; kommerzielle Nutzer müssen einen geringen Obolus entrichten.

- 8. Sie bekommen als Ergebnis ein HTML-Element angezeigt, das Sie in das gewünschte Formular einbauen müssen, nach folgendem Muster:

```
<input type="hidden" name="formmailer" value="XXXXX" />
```

Anstelle von XXXXX erhalten Sie fünf Ziffern. Diese müssen Sie an der entsprechenden Stelle statt XXXX schreiben!
Info Es handelt sich bei `<input type="hidden" />` um ein weiteres Formularfeld: ein verstecktes Feld. Das wird im Browser nicht angezeigt, aber beim Formularversand mit übermittelt. Richtig „versteckt“ ist das Feld jedoch nicht,

Löwenstark Digital Group GmbH
Petzvalstr. 38
38104 Braunschweig
Tel.: 0531 21360 5500
Fax: 0531 21360 5795
kontakt@loewenstark.com

da es im **HTML-Quelltext** steht.

- 9. Ändern Sie das Formular so, dass es an Formmailer verschickt wird, d.h. ergänzen Sie im `<form>`-Element: `action="http://send.formmailer.com/" method="post"`.

Am Ende könnte also Ihr Formular wie folgt aussehen. Denken Sie auf jeden Fall daran, anstelle von XXXXX Ihre persönlichen fünf Ziffern anzugeben:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<!-- feedbackformular-formmailer.html -->
<html>
<head>
<title>Formulare</title>
</head>
<body>
<form action="http://send.formmailer.com/" method="post">
<input type="hidden" name="formmailer" value="XXXXX" />
<input type="radio" name="Anrede" value="Herr" />
Herr
<input type="radio" name="Anrede" value="Frau" />
Frau
<br>
Vorname:
<input type="text" name="Vorname" />
<br>
Nachname:
<input type="text" name="Nachname" />
<br>
E-Mail:
<input type="text" name="Mail" />
<br>
Kommentar zu meiner Website:
<textarea name="Kommentar" cols="30" rows="2">
Was ich schon immer mal sagen wollte ...
</textarea>
<br>
Woher kennen Sie diese Website?
<select name="Quelle" size="3">
<option value="such">Suchmaschine</option>
<option value="mund">Mund-zu-Mund-Propaganda</option>
<option value="tv">Fernsehwerbung</option>
<option value="sonst">Sonstiges</option>
</select>
<br>
<input type="checkbox" name="Newsletter" value="ja" />
Ja, ich möchte den Newsletter abonnieren!
```

Löwenstark Digital Group GmbH
Petzvalstr. 38
38104 Braunschweig
Tel.: 0531 21360 5500
Fax: 0531 21360 5795
kontakt@loewenstark.com

```
<br>
<input type="image" src="grafik.gif" name="Versendegrafik" style="border:0px;"
alt="Formular versenden" />
</form>
</body>
</html>
```

Info Wichtig: Jedes Formular müssen Sie im Administrationsmenü von http://www.formmailer.com/ einzeln registrieren (Sie erhalten dann auch eine neue Nummer)!

Formulardesign und Formularoptimierung

Seiten mit Formularen werden von Anwendern meist länger angezeigt und intensiver wahrgenommen als andere Seiten. Allein schon deshalb ist es Anbieterpflicht, **Formulare** so zu gestalten, dass der Anwender sich ohne Probleme darin zurechtfindet.
Folgende Punkte sind wichtig:

- Formularbereiche sollten optisch erkennbar sein.
- Feldbeschriftungen und Formularelemente sollten einheitlich zueinander ausgerichtet sein. Der Zusammenhang zwischen einer Feldbeschriftung und dem zugehörigen Feld muss eindeutig erkennbar sein.
- Eingabefelder sollten ordentlich positioniert sein und so weit sinnvoll eine einheitliche Länge haben, damit das Formular nicht zu unruhig wirkt.
- In umfangreicheren Formularen sollten logisch/thematische zusammengehörige Elemente gruppiert werden. Die Gruppen sollten Gruppenüberschriften erhalten.
- Wenn Felder Standardwerte haben, kann dem Anwender Arbeit erspart werden, indem die Felder mit den Standardwerten vorbelegt sind oder Standardwerte voreingestellt oder vorausgewählt sind.
- Felder, deren Zweck sich nicht selbstverständlich erschließt, oder Felder, die als Wert ein bestimmtes Format oder einen bestimmten Wertebereich erhalten, sollten erläutert werden.
- Felder, die für das verarbeitende Script Pflichtfelder sind, sollten dem Anwender als solche kenntlich gemacht werden.

Anhand eines einfachen Beispielformulars sollen diese Punkte erläutert werden. Der vollständige **Quelltext** der HTML-Datei des Beispiels lautet:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
<meta http-equiv="content-type" content="text/html; charset=ISO-8859-1">
<style type="text/css">
input, fieldset, legend {
font-size:13px;
font-family:Arial,
sans-serif;
}
```

Löwenstark Digital Group GmbH
Petzvalstr. 38
38104 Braunschweig
Tel.: 0531 21360 5500
Fax: 0531 21360 5795
kontakt@loewenstark.com

```
</style>
<title>Formular</title>
</head>
<body>
<h1>Login</h1>
<form action="login.php" method="post" style="background-color:#DDDDDD;
padding:4px">
<fieldset>
<legend>Zugangsdaten</legend>
<table>
<tr>
<td style="width:150px; font-weight:bold; text-align:right">
<label for="user">* Benutzername:</label>
</td>
<td>
<input type="text" maxlength="40" name="user" id="user" style="width:200px">
(max. 40 Zeichen)
</td>
</tr>
<tr>
<td style="font-weight:bold; text-align:right">
<label for="pwd">* Passwort:</label>
</td>
<td>
<input type="password" maxlength="10" name="pwd" id="pwd" style="width:200px">
(max. 10 Zeichen)
</td>
</tr>
</table>
</fieldset>
<fieldset>
<legend>Optionen</legend>
<table>
<tr>
<td style="width:150px; font-weight:bold; text-align:right; vertical-align:top">
<label for="cookies">Cookies:</label>
</td>
<td>
<input type="radio" name="cookies" id="cookies" value="yes" checked>ja
<br>
<input type="radio" name="cookies" value="no">nein
</td>
</tr>
<tr>
<td style="font-weight:bold; text-align:right; vertical-align:top">
<label for="public">Benutzername:</label>
</td>
<td>
```

Löwenstark Digital Group GmbH
Petzvalstr. 38
38104 Braunschweig
Tel.: 0531 21360 5500
Fax: 0531 21360 5795
kontakt@loewenstark.com

```
<input type="radio" name="public" id="public" value="yes">anderen Usern anzeigen
<br>
<input type="radio" name="public" value="no" checked>anderen Usern nicht
anzeigen
</td>
</tr>
</table>
</fieldset>
<p><button type="submit" accesskey="s" style="font-weight:bold">
Login <span style="text-decoration:underline">s</span>enden</button></p>
<fieldset>
<legend>Hinweis</legend>
<p>Mit <span style="font-weight:bold">*</span> gekennzeichnete  Felder müssen
ausgefüllt werden!</p>
</fieldset>
</form>
</body>
</html>
```

Info Das Formular selbst wird im Beispiel optisch gekennzeichnet durch die hellgraue Hintergrundeinfärbung (`<form ... style="background-color:#DDDDDD; ...">`). Für die Gruppierung der einzelnen Bereiche wird ein eigens dafür gedachtes
HTML-Element benutzt: das `fieldset`-Element. Insgesamt werden im Beispiel drei solcher Fieldsets definiert, erkennbar an den Rändern, die sie erzeugen. Mit `<fieldset>...</fieldset>` wird der gewünschte Bereich des Formulars eingeschlossen. Hinter jedem einleitenden `<fieldset>`-Tag folgt jedoch noch ein Text, der mit `<legend>...</legend>` ausgezeichnet ist. Dieser Text wird als Gruppenüberschrift
des `fieldset`-Bereichs angezeigt und vom Browser bei der Anzeige links oben in den `fieldset`-Rahmen positioniert. Einen Nachteil haben die `fieldset`-Bereiche allerdings: Sie verhindern, dass eine durchgängige rahmenlose
Tabelle notiert werden kann, um Beschriftungen und Formularelemente zueinander auszurichten. Deshalb übernehmen im Beispiel zwei solcher Tabellen diese Aufgabe. Durch geeignete CSS-Angaben werden die Zellen mit den Feldbeschriftungen rechtsbündig und
wo nötig am oberen Rand ausgerichtet. Die einzeiligen Eingabefelder erhalten durch die CSS-Eigenschaft `width` die gleiche Länge, auch wenn im Passwortfeld nur zehn Zeichen eingegeben werden können. Durch die einheitliche Länge wirkt das Formular
jedoch ruhiger. Stattdessen wird der Anwender hinter den Eingabefeldern durch Kurzhinweise darüber aufgeklärt, wie lange die Feldeingaben maximal sein dürfen.

Durch das Markup der rahmenlosen Tabelle stehen Feldbeschriftung und Formularfeld

nicht unmittelbar beieinander. Bei nicht visueller, also z.B. akustischer Ausgabe der Webseite könnte dies zu Problemen bei der Zuordnung von Beschriftung und Feld führen. HTML bietet hierfür eine Lösung an, die wir im Beispielformular ebenfalls verwendet
haben. Der Beschriftungstext wird jeweils mit `<label for="...">...</label>` ausgezeichnet. Das `for`-

Löwenstark Digital Group GmbH
Petzvalstr. 38
38104 Braunschweig
Tel.: 0531 21360 5500
Fax: 0531 21360 5795
kontakt@loewenstark.com

Attribut erhält als Wert einen `id`-Namen. Im zugehörigen Formularelement wird dieser Name in einem `id`-Attribut notiert. Dadurch wird ein logischer Zusammenhang zwischen Beschriftung und Formularelement hergestellt. Für die Felder Benutzername und Passwort gibt es keine sinnvollen Vorbelegungswerte. Die beiden Radiobutton-Gruppen der Optionen Cookies und Benutzername dagegen erhalten mithilfe des Attributs `checked` je eine Default-Auswahl. Auf diese Weise ist sichergestellt, dass in jedem Fall eine gültige Option beim Absenden des Formulars übertragen wird. Der Submit-Button zum Absenden des Formulars wird im Beispiel mithilfe des `button`-Elements realisiert. Grund ist, dass die Beschriftung einen als unterstrichen formatierten Buchstaben enthält (das „s“ bei Login senden). Bei einer Wertzuweisung des Beschriftungstextes an das `value`-Attribut eines `<input type="submit">`-Konstrukts wäre das nicht möglich. Unterstrichen wird das „s“ deshalb, weil im einleitenden `<button>`-Tag das Attribut `accesskey="s"` notiert ist. Das `accesskey`-Attribut funktioniert in **Formularelementen** genauso wie bei Hyperlinks. Durch einen Hotkey wie `(Alt)+(s)` (browserabhängig, kann der Anwender das Formular über die Tastatur absenden. Ebenso wie das `accesskey`-Attribut kann in Formularelementen analog zu Hyperlinks ein `tabindex`-Attribut notiert werden. Und genauso wie bei den Links kann durch die als Wert zugewiesene Zahl die Reihenfolge beim Anspringen der Felder über die `(Umschalt)`-Taste festgelegt werden.

Formular-Eingabe-Felder

Formulare werden in **HTML** mit dem Tag `<form>` eingeleitet und mit `</form>` abgeschlossen. Achten Sie darauf, dass alle weiteren Angaben zum Formular zwischen diesen beiden Tags stehen müssen. Wenn Sie mit einem Wysiwig-Editor arbeiten (Frontpage o.ä.) können Sie mit Maus-Klicks die Formularfelder einfügen. Achten Sie darauf, dass auch hier nur ein einziges einleitendes `<form>` Tag am Anfang des Formulars und nur ein einziges abschliessendes `</form>` Tag am Ende des Formulars im Quelltext erscheint.

Eingabe-Felder werden in HTML mit dem Tag `<input>` eingefügt. Um ein einfaches einzeiliges Eingabefeld zu erzeugen, muss das `input`-Tag mit dem attribut `type="text"` erweitert werden. Mit dem Attribut `size`legen Sie die Länge des Eingabefeldes fest. Die Wertangabe bezieht sich auf die Anzahl der Zeichen: `size="5"` erzeugt ein fünf Zeichen langes Eingabefeld. Jedem Eingabefeld muss über das Attribut `name` ein Name zugewiesen werden:
`name ="alter"`.

```
<form>
<input type="text" size="5" name="alter">
</form>
```

Erzeugt dieses Eingabe-Feld. Sie können Text eingeben und es werden maximal fünf Zeichen im Eingabefenster dargestellt. Geben Sie mehr als fünf Zeichen ein, wird der Text gescrollt (ohne Scrollbalken).
Bei vielen Eingabefeldern besteht die Möglichkeit, mehr Zeichen einzugeben, wobei das Eingabefeld gescrollt wird. Mit dem Attribut `maxlength` legen Sie die maximale Länge fest.

Löwenstark Digital Group GmbH
Petzvalstr. 38
38104 Braunschweig
Tel.: 0531 21360 5500
Fax: 0531 21360 5795
kontakt@loewenstark.com

```
<form>
<inputtype="text" size="5" name="alter" maxlength="5">
</form>
```

Testen Sie im Browser. Diese Möglichkeit empfiehlt sich bei der Eingabe von Postleitzahlen (zip-code), um unsinnige Eingaben zu unterdrücken. Fügen Sie nach diesem Schema weitere **Eingabefelder** hinzu, um Ihr Formular zu erweitern. Wählen
Sie für `maxsize` einen Wert, der realistisch ist: Eingabefelder für Namen sind mit 50 Zeichen sicherlich ausreichend lang. Die Beschriftung der Eingabefelder erfolgt wie gewohnt in HTML:

```
<form>
Name:
<input type="text" size="15" name="name" maxlength="50">
<br>
Vorname:
<input type="text" size="15" name="vorname" maxlength="50">
<br>
PLZ:
<input type="text" size="5" name="plz" maxlength="5">
<br>
</form>
```

Um die Positionierung der Beschriftung und der Eingabefelder
zu steuern, binden Sie alles in eine Tabelle ein. z.B. zweispaltige Tabelle, in der linken Spalte stehen rechtsbündig die Beschriftungen, in der rechten Spalte stehen linksbündig die Eingabefelder. Überprüfen Sie Ihre Einstellungen in Browsern. Die Unterschiede
sind gravierend. Sie haben keine Kontrolle über die Art des Eingabefeldes, der Browser allein bestimmt das Aussehen. Damit Ihre Seitenbesucher auch Nachrichten hinterlassen können, benötigen Sie ein echtes Eingabefeld. Mit dem Tag `textarea`wird dieses erzeugt. Die Textarea muss einen Namen erhalten. Die Grösse des Textarea wird mit Reihen (`rows`) und Spalten (`cols`) definiert. Die Zahlenangabe bezieht sich auf die Anzahl der Zeichen.

```
<textareaname="nachricht" cols="30" rows="5">
</textarea>
```

Mit den obigen Tags und Attributen wird ein fünfzeiliges Eingabefeld mit 30 Zeichen pro Zeile erzeugt. Mit den beiden Eingabefelder `textarea` und `input` lassen sich bereits sehr schöne und wirkungsvolle E-Mail-Formulare erstellen.
Um das Formular abzusenden oder die Formulareingaben zu löschen, sind noch Buttons (Schalter) erforderlich, um diese Aktionen auszulösen. Mit dem Tag `input type=submit` wird eine Schaltfläche erzeugt, die das **Formular** absenden
würde. Die Beschriftung des Buttons wird mit dem Attribut `value="Senden"` bestimmt. Ebenso wie `type=submit` ist `type=reset` bereits in HTML definiert. Die damit verbundenen Aktionen werden automatisch ausgeführt, Sie

Löwenstark Digital Group GmbH
Petzvalstr. 38
38104 Braunschweig
Tel.: 0531 21360 5500
Fax: 0531 21360 5795
kontakt@loewenstark.com

müssen sich um nichts kümmern. `Reset` setzt die Formulareingaben zurück, konkret: Es wird alles gelöscht.

```
<inputtype="submit" value="Senden">
<inputtype="reset" value="Löschen">
```

Das E-Mail-Formular im Überblick:

```
<form>
Name: <input type="text" size="15" name="name" maxlength="50">
<br>
Vorname: <input type="text" size="15" name="vorname" maxlength="50">
<br>
PLZ: <input type="text" size="5" name="plz" maxlength="5">
<br>
<textarea name="nachricht" cols="30" rows="5">
</textarea>
<input type="submit" value="Senden">
<inputtype="reset" value="Löschen">
</form>
```

Mehrzeilige Eingabebereiche

Ein mehrzeiliger **Eingabebereich** eignet sich für Freitext aller Art, etwa für Notizen, Bemerkungen, oder bei einem Forum oder Gästebuch für den Beitragstext. In HTML gibt es dafür ein eigenes Element:

```
<textarea name="bemerkung" cols="50" rows="10">
Platz für Ihre Bemerkungen!
</textarea>
```

Info Das *textarea*-Element besteht aus Anfangs- und End-Tag. Dazwischen kann Text notiert werden. Dieser wird als Vorbelegungstext interpretiert. Falls der Anwender mehr Text eingibt als die Größe des Bereichs zulässt, fügt der Browser
automatisch eine Scrollleiste ein. Die Größe des Bereichs können Sie durch die Attribute *rows*= (Anzahl Zeilen) und *cols*= (Anzahl Spalten) festlegen. Diese Angaben sind allerdings nur für textmodusbasierte Browser wirklich von Bedeutung.
Für grafische Browser empfiehlt es sich, die Größe des Eingabefelds zusätzlich mit entsprechenden CSS-Eigenschaften (*width* und *height*) festzulegen (in Zusammenhang mit der optischen Gestaltung von Formularen gehen wir später noch
genauer darauf ein). Eingegebener Text wird übrigens automatisch umgebrochen. Durch Betätigen der (`Enter`)-Taste kann der Anwender Zeilenumbrüche im Text erzwingen.

Mehrzeilige Textfelder

Löwenstark Digital Group GmbH
Petzvalstr. 38
38104 Braunschweig
Tel.: 0531 21360 5500
Fax: 0531 21360 5795
kontakt@loewenstark.com

Wenn eine Textzeile nicht mehr ausreicht, müssen es mehrere sein. Dazu gibt es ein eigenes HTML-Tag, das ausnahmsweise nicht <*input*> heißt. Die Rede ist von <*textarea*>. Sie finden das Element auf vielen Websites, beispielsweise
wenn Sie eine aktuelle Java-Version herunterladen und installieren möchten. Im Textfeld stehen dann die Nutzungsbedingungen. Die meisten mehrzeiligen Textfelder sind allerdings dazu da, dass dort vom Benutzer etwas eingetragen werden kann. Die Anpassung
ist sehr einfach:

- 1. Geben Sie im Attribut *cols* die Breite des Textfelds (in Zeichen) an.
- 2. Geben Sie im Attribut *rows* die Höhe des Textfelds (in Zeilen) an.

Natürlich können Sie auch die **CSS-Befehle** *width* und *height* verwenden.

```
<textarea cols="50" rows="8">
</textarea>
```

- 3. Wenn das Textfeld vorausgefüllt werden soll, geben Sie den entsprechenden Text zwischen <*textarea*> und <*/textarea*> an:

```
<textarea>
Bitte tippen Sie munter drauflos ... Ich freue mich auf Ihr Feedback
</textarea>
```

- 4. Packen Sie Stilangaben ins *style*-Attribut:

```
<textarea style="background-color:orange; font-weight:bold;">
</textarea>
```

- 5. Außerdem benötigt das Textfeld noch einen ordentlichen Namen, verwenden Sie dazu das *name*-Attribut:

```
<textarea name="Mehrzeilig">
</textarea>
```

Nachfolgend ein komplettes Formular:

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

Löwenstark Digital Group GmbH
Petzvalstr. 38
38104 Braunschweig
Tel.: 0531 21360 5500
Fax: 0531 21360 5795
kontakt@loewenstark.com

```
<title>Formulare</title>
</head>
<body>
<form action="formulardaten.html" method="get">
<textarea cols="50" rows="8" style="background-color:orange; font-weight:bold;"
name="Mehrzeilig">
Bitte tippen Sie hier ...
Ich freue mich auf Ihr Feedback!
</textarea>
<br>
<input type="submit" value=">Hier klicken<" name="Versand" style="background-
color:red; font-weight:bold;" />
</form>
</body>
</html>
```

Info Sie sehen hier zwei Unterschiede zu herkömmlichem HTML:

- Zeilenumbrüche innerhalb eines mehrzeiligen Textfelds werden tatsächlich als solche dargestellt.
- Mehrere Leerzeichen hintereinander werden innerhalb eines mehrzeiligen Textfelds ebenfalls als solche dargestellt.

Der Versand schließlich bringt zu Tage, was übermittelt wird: der Name des Felds (*name*-Attribut) sowie das, was eingetragen worden ist – hier der Vorgabetext, weil wir nichts geändert haben. Ansonsten wäre es der Text, den der Surfer eingegeben
hat.

Radiobuttons und Checkboxen

Die nachfolgend vorgestellten Formularelemente können Sie zwischen `<form>` und `</form>` notieren.

Eingabefelder, Radiobuttons, Checkboxen und Schaltflächen

Eine ganze Reihe von **Formularelementen** in HTML werden alle nach dem gleichen Schema ausgezeichnet:

```
<input type="..." name="..." value="...">
```

<input type=	Bedeutung
"text"	Einzeiliges Eingabefeld
"password"	Eingabefeld für Passwörter

Löwenstark Digital Group GmbH
Petzvalstr. 38
38104 Braunschweig
Tel.: 0531 21360 5500
Fax: 0531 21360 5795
kontakt@loewenstark.com

<input type=	Bedeutung
"radio"	Radiobutton (ja/nein-Schalter). Von mehreren Radiobuttons mit gleichem name-Attributwert kann immer nur einer ausgewählt werden.
"checkbox"	Checkbox (ja/nein-Schalter)
"button"	Schaltfläche ohne vordefinierte Funktion
"submit"	Schaltfläche zum Absenden der Formulardaten
"reset"	Schaltfläche zum Zurücksetzen des Formulars in den Anfangszustand. Eingaben werden gelöscht.
"image"	Alternative zu „submit“. Ein zusätzliches src-Attribut im <input>-Tag erlaubt das Referenzieren eines Grafik-URI. Die Grafik ist anklickbar und sendet das Formular ab.
"file"	Eingabefeld mit Durchsuchen-Schaltfläche zum Auswählen einer beim Anwender lokal gespeicherten Datei. Die Datei wird zusammen mit den Formulardaten abgesendet („File-Upload“).
"hidden"	Versteckte Information; Element ist nicht sichtbar.

Info Einzeilige Eingabefelder eignen sich für alle Arten von Einzeldaten wie Namen, Mail-Adressen, Telefonnummern, Suchausdrücke usw. Passwortfelder sollten nur für entsprechende Zwecke genutzt werden. Dabei ist es wichtig zu wissen, dass lediglich
die Darstellung am Bildschirm verschleiert wird. Die eingegebenen Daten werden dadurch noch nicht verschlüsselt. Radiobuttons sind nur als Gruppe sinnvoll. Dabei erhalten alle zur Gruppe gehörigen Radiobuttons bei name= den gleichen Wert.
Der Browser reagiert so, dass immer nur einer der Radiobuttons aktivierbar ist. Eine sinnvolle Anwendung sind beispielsweise zwei Radiobuttons zur Auswahl des Geschlechts männlich oder weiblich.

Checkboxen eignen sich vor allem zum Ein-/Ausschalten von Optionen wie etwa

„Groß-/ Kleinschreibung unterscheiden“. Sowohl bei Radiobuttons als auch bei Checkboxen können Sie im <input>-Tag das Standalone-Attribut checked (in XHTML: checked="checked") notieren. Damit erzwingen
Sie, dass das entsprechende Element beim Aufrufen oder Zurücksetzen des Formulars vorausgewählt ist. Bei zusammengehörigen Radiobuttons kann natürlich nur eines der Elemente vorausgewählt werden. Bei den Schaltflächen unterscheidet **HTML** grundsätzlich zwischen funktionslosen Schaltflächen (type="button") und den beiden vordefinierten Schaltflächen zum Absenden der Formulareingaben

Löwenstark Digital Group GmbH
Petzvalstr. 38
38104 Braunschweig
Tel.: 0531 21360 5500
Fax: 0531 21360 5795
kontakt@loewenstark.com

(„OK“-Funktionalität bzw. Submit-Button – `type="submit"`) und zum Zurücksetzen der Formulareingaben („Abbrechen“-Funktionalität bzw. Reset-Button – `type="reset"`). Funktionslose Schaltflächen sind vor allem für clientseitige Interaktionsverarbeitung mit JavaScript gedacht. Die beiden anderen Typen sind dagegen Nachbildungen der Standardschaltflächen von Dialogen, wie man sie auch bei modernen grafischen Benutzeroberflächen kennt. Die `reset`-Schaltfläche ist jedoch, was Usability betrifft, mit Vorsicht zu genießen, vor allem bei umfangreicheren Formularen.
Ein unachtsamer Anwenderklick auf eine solche Schaltfläche und alle zuvor getätigten Formulareingaben sind unwiderruflich weg. Die meisten Formulare bieten daher nur die `submit`-Schaltfläche an. Die Beschriftung der Schaltfläche wird in allen Fällen im `value`-Attribut festgelegt. Die Auszeichnung von Schaltflächen über `<input ...>` ist nur die ältere von zwei Varianten. Für die neuere Variante wurde mit HTML 4.0 das `button`-Element eingeführt. Die folgenden beiden Beispiele bewirken im Browser letztlich das Gleiche:

```
<input type="submit" value="Absenden">
<button type="submit" value="Absenden">
</button>
```

Der Unterschied besteht vor allem darin, dass das `input`-Element ein Standalone-Element ohne End-Tag ist (in XHTML muss es daher in der Form `<input.../>` notiert werden). Das `button`-Element erlaubt dagegen Elementinhalt, und zwar sowohl Text als auch Block- und Inline-Elemente. Auf diese Weise können Sie Buttonbeschriftungen auffällig gestalten, beispielsweise auch mit Grafiken. Die Attribute `name=` und `value=` bestimmen die Daten, die beim Absenden des **Formulars** übertragen werden. Das `name`-Attribut bezeichnet den Namen eines Parameters und das `value`-Attribut dessen Wert. Während das `name`-Attribut bei allen `input`-Typen außer bei `submit` und `reset` obligatorisch ist, ist das `value`-Attribut nicht zwingend erforderlich, da der Wert zumindest bei Eingabefeldern, Radiobuttons, Checkboxen usw. durch den Anwender bestimmt wird. Das `value`-Attribut eignet sich bei Eingabefeldern jedoch dazu, einen Default-Wert vorzugeben. Beispiel:

```
URL-Adresse: <input type="text" name="url" value="http:// ">
```

Info In diesem Fall wird das Eingabefeld mit dem Namen **url** mit der Zeichenfolge `http://` vorbelegt. Bei der Wertzuweisung an das `name`-Attribut gelten gewisse Regeln. Der Parametername muss mit einem Buchstaben A–Z oder a–z beginnen. Als weitere Zeichen sind ebenfalls A–Z und a–z erlaubt sowie die Ziffern 0–9 und die Zeichen für Unterstrich (`_`), den Bindestrich (`-`), Doppelpunkt (`:`) und Punkt (`.`).
In Hinblick auf Programmiersprachen wie Java-Script oder PHP, die auf die Parameter über diese Namen zugreifen, sollten Sie sich jedoch unbedingt auf Buchstaben, Ziffern und Unterstrich beschränken. Groß- und Kleinschreibung werden bei den genannten Programmiersprachen ebenfalls unterschieden. Auf die optische Gestaltung von Elementen wie Eingabefeldern, wozu auch die Bestimmung von angezeigter Breite gehört, gehen wir später darauf ein. Die Breite eines Eingabefelds hat jedoch nichts damit zu tun, wie viele Zeichen in dem Feld eingegeben werden können. Um die Anzahl der möglichen Zeichen in einem Feld zu begrenzen, benötigen Sie das `maxlength`-Attribut. Ein Beispiel:

Löwenstark Digital Group GmbH
Petzvalstr. 38
38104 Braunschweig
Tel.: 0531 21360 5500
Fax: 0531 21360 5795
kontakt@loewenstark.com

```
Vorname: <input type="text" name="given_name" maxlength="30">
```

Info In diesem Beispiel wird durch die Angabe `maxlength="30"` festgelegt, dass der Browser von Eingaben in diesem Feld maximal die 30 ersten Zeichen übertragen soll. Solche Begrenzungsangaben sind wichtig, wenn das verarbeitende
Script die eingegebenen Daten beispielsweise in eine Datenbank schreibt, in der die Feldlängen entsprechend begrenzt sind. Allerdings ist auf die Zuverlässigkeit der Browser bei der Interpretation des maxlength-Attributs kein Verlass.
Stattdessen empfiehlt es sich in jedem Fall, Feldeingaben vor dem Schreiben in eine Datenbank auch auf ihre Länge hin zu prüfen und gegebenenfalls abzuschneiden. Speziellere Formen des `input`-Elements sind `type="image"`, `type="file"` und `type="hidden"`. Die Angabe `type="image"` ersetzt `type="submit"` und ist funktionsgleich. Anstelle einer Standardschaltfläche kann eine frei wählbare Grafik als Absende-Button definiert werden, wie im folgenden
Beispiel:

```
<input type="image" src="ok.png" alt="Absenden">
```

Info Die gewünschte Grafik wird beim `src`-Attribut als URI zugewiesen. Außerdem sollten Sie in diesem Fall auch das alt-Attribut notieren, das den Alternativtext enthält, falls die Grafik nicht angezeigt werden kann.
Die gleiche Funktionalität wie durch `type="image"` erreichen Sie übrigens über das `button`-Element wie folgt:

```
<button type="submit">
<img src="ok.png" alt="Absenden">
</button>
```

Info Versteckte **Formularelemente** (`type="hidden"`) sind nur in Zusammenhang mit Scripting von Bedeutung. Wir werden deshalb erst bei den Scriptsprachen näher darauf eingehen. Nicht ganz unproblematisch ist das `input`-Element
mit der Funktionalität Datei-Upload (`type="file"`). Das serverseitige Script, das auf diese Weise Dateien vom Anwender übertragen bekommt, muss auf jeden Fall eine Reihe von Sicherheitschecks durchführen, um den Server nicht zu gefährden.
Denn Dateien können Viren transportieren, sehr groß sein usw. Die erlaubte Maximalgröße ist zwar in HTML wie auch bei der Zeichenlänge von Eingabefeldern durch das `maxlength`-Attribut begrenzbar. Doch wie schon erwähnt ist kein Verlass
darauf, dass alle Browser das Attribut konsequent beachten. Wenn Sie in ein Formular ein File-Upload-Feld integrieren, sollte das einleitende `<form>`-Tag folgende Angaben erhalten:

```
<form action="..." method="post" enctype="multipart/form-data">
```

Info Die enctype-Angabe ist wichtig, damit serverseitig erkannt werden kann, dass es sich um mehrteilige Daten handelt. Der Wert `multipart/form-data` ist eine Mime-Type-Angabe für solche Zwecke. Als

Löwenstark Digital Group GmbH
Petzvalstr. 38
38104 Braunschweig
Tel.: 0531 21360 5500
Fax: 0531 21360 5795
kontakt@loewenstark.com

Übertragungsmethode sollte im
Fall von File-Upload stets die POST-Methode gewählt werden. Im <input>-Tag ist es außerdem möglich, mithilfe eines accept-Attributs die Dateiauswahl zu begrenzen, z.B.:

```
<input name="Datei" type="file" maxlength="100000" accept="image/*">
```

Info In diesem Beispiel wird dem Anwender erlaubt, eine lokal gespeicherte Grafikdatei (image/) beliebigen Typs (*) mit einer Maximalgröße von 100.000 Byte zum Hochladen auszuwählen.

Versteckte Elemente in Formularen definieren

Sie können Felder in einem **Formular** definieren, die dem Anwender nicht angezeigt werden. Versteckte Felder können Daten enthalten. Beim Absenden des Formulars werden die Daten versteckter Felder mit übertragen. Auf diese Weise
können Sie beispielsweise zusätzliche Informationen an CGI-Scripts übergeben oder erläuternden Text einfügen, der bei der E-Mail-übertragung der Formulardaten in der E-Mail mit enthalten ist.

Auch für JavaScript ist diese Möglichkeit interessant. Da ein JavaScript Formularfelder problemlos auslesen und deren Werte auch ändern kann, ist es auf diese Weise bequem möglich, interne Daten zu speichern, die nicht am Bildschirm angezeigt werden.
So könnte ein JavaScript beispielsweise, nachdem die Seite mit dem Formular beim Anwender geladen ist, Informationen zur Bildschirmauflösung beim Anwender sammeln (siehe Screen-Objekt) und die Ergebnisse in ein verstecktes
Formularfeld schreiben. Die Daten werden dann, wenn der Anwender das Formular abschickt, mit übertragen.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
<title>Versteckte Elemente in Formularen definieren</title>
</head>
<body>
<h1>Feedback</h1>
<form name="Feedback" action="input_hidden.htm">
<p>
<input type="hidden" name="UserBrowser" value="">
Ihr Kommentar:
<br>
<textarea name="UserKommentar" rows="2" cols="20">
</textarea>
<br>
<input type="submit" value="senden">
</p>
<br>
```

Löwenstark Digital Group GmbH
Petzvalstr. 38
38104 Braunschweig
Tel.: 0531 21360 5500
Fax: 0531 21360 5795
kontakt@loewenstark.com

```
<script type="text/javascript">
document.Feedback.UserBrowser.value = navigator.appName;
</script>
</form>
</body>
</html>
```

Info Mit `<input type="hidden">` definieren Sie versteckte Felder in einem Formular (input = Eingabe, hidden = versteckt). Die Daten selbst bestimmen Sie beim Attribut `value` (value = Wert). Im obigen Beispiel
erhält das versteckte **Formularfeld** zunächst keine Daten (`value=""`). Unterhalb des Formulars ist jedoch ein JavaScript notiert. Dieses Script ermittelt den Browser, den der Anwender verwendet, und schreibt den
ermittelten Wert in das versteckte Formularfeld. Wenn der Anwender das Formular absendet, wird also der verwendete Browser als Formularinhalt mit übertragen. Das Ergebnis wird in der Adresszeile sichtbar.