

HTML, XHTML und CSS

Kurzreferenz (PDF-Fassung)

Version 1.0

Datum: 20. Februar 2002

Autor: Johann-Christian Hanke

Inhaltsverzeichnis:

(X)HTML/CSS-REFERENZ: DIE WICHTIGSTEN BEFEHLE:	3
ALLGEMEINE TAGS.....	3
ABSÄTZE, ZEILENUMBRUCH UND <DIV>	3
TAGS ZUR EINFACHEN TEXTGESTALTUNG	4
GRAFIK-DATEIEN EINFÜGEN.....	4
LINIEN GESTALTEN.....	5
LISTEN GESTALTEN.....	5
FONT-TAG.....	5
KLANG-DATEIEN UND VIDEOS EINFÜGEN.....	6
LAUFTEXT EINFÜGEN UND GESTALTEN.....	6
TAGS FÜR FORMULARE	7
ANKER UND HYPERLINKS.....	8
TAGS FÜR TABELLEN.....	8
TAGS FÜR FRAMESETS.....	9
STYLE SHEETS: DIE WICHTIGSTEN ATTRIBUTE VON CSS	10
ATTRIBUTE ZUR STEUERUNG VON SCHRIFTEIGENSCHAFTEN.....	10
ATTRIBUTE ZUR STEUERUNG VON LISTEN.....	11
TEXTAUSRICHTUNG.....	11
GESTALTUNG VON RÄNDERN UND RAHMEN.....	12
FARBE UND HINTERGRUNDGRAFIK.....	13
ABSOLUTE POSITIONIERUNG (WICHTIG FÜR DHTML).....	13
MABEINHEITEN, FARBEN UND SONDERZEICHEN	14
DIE WICHTIGSTEN FARBEN	14
RGB-FARB-CODES FÜR CSS	14
UMSCHREIBUNG VON SONDERZEICHEN.....	14
DIE LÄNDER-CODES FÜR ZEICHENSÄTZE.....	14
HOVER-LINKS	14
MABEINHEITEN FÜR CSS	14

(X)HTML/CSS-Referenz: Die wichtigsten Befehle:

Auf den nächsten Seiten zeige ich die wichtigsten Befehle von (X)HTML mit Erläuterungen. Wenn sinnvoll, habe ich in die rechte Spalte gleich ein Anwendungsbeispiel gesetzt.

Falls das Tag einen internen Schrägstrich besitzt (`<meta />`) bedeutet das, dass das Tag intern abgeschaltet werden muss. Das ist aber nur für den interessant, der schon dem XHTML-Standard folgen möchte. Ansonsten würde ich diesen internen Schrägstrich einfach ignorieren.

Allgemeine Tags

In dieser Tabelle finden sich die bekannten allgemeinen Tags zu HTML und die wichtigsten Attribute. Zur Erinnerung zeige ich zuerst das Grundgerüst für eine HTML-Seite. Nicht vergessen: Jedes Tag benötigt ein End-Tag. HTML bzw. XHTML-Dokumente tragen die Nachsilbe *htm* bzw. *html*.

```
<html>
<head>
  <title>Seitentitel, erscheint in der Titelzeile des Browsers</title>
</head>
  <body>
    Inhalt
  </body>
</html>
```

Auf die `<meta>`-Tags gehe ich hier nicht weiter ein. Sie werden auch nur von wenigen Suchmaschinen ausgewertet, nicht vom Branchenprimus Google. Fireball (www.fireball.de) besitzt einen Meta-Tag-Generator. Einen Artikel zu DOCTYPE veröffentliche ich auf www.jchanke.de/homepage im Bereich zum Heft 168.

Tags	Attribute	mögliche Werte	Erläuterung	Anwendungsbeispiel
<code><html></html></code>			definiert HTML-Datei	
<code><head></head></code>			definiert den Kopf	
<code><title></title></code>			erzeugt Titel der Datei, der in der Titelzeile des Browsers erscheint	<code><title>Meine erste Seite</title></code>
<code><meta /></code>			erzeugt Bereich für Zusatzinformationen	<code><meta name="author" content="Peter Pan" /></code>
<code><body></body></code>			leitet Hauptbereich ein	
	text	hex. Code, Farbname	definiert Textfarbe	<code><body text="#000080"></code>
	background		fügt Hintergrundgrafik ein	<code><body background="picture.gif"></code>
	bgcolor	hex. Code, Farbname	definiert eine Hintergrundfarbe	<code><body bgcolor="silver"></code>
	link	hex. Code, Farbname	Farbe des Links	<code><body link="green"></code>
	alink	hex. Code, Farbname	Farbe des aktiven Links	<code><body alink="#008080"></code>
	vlink	hex. Code, Farbname	Farbe des besuchten Links	<code><body vlink="blue"></code>

Absätze, Zeilenumbruch und `<div>`

Die folgende Tabelle zeigt, wie man Absätze `<p></p>` bzw. einen Zeilenumbruch `
` gestaltet und ausrichtet. Besonders interessant ist der `<div>`-Container, den man für die freie Gestaltung mit Hilfe von Style Sheets verwenden kann. Man kann `<div>` an Stelle von `<p>` verwenden.

Tags	Attribute	mögliche Werte	Erläuterung	Anwendungsbeispiel
<code><p></p></code>			definiert einen Absatz	<code><p>Das ist ein Absatz</p></code>
	align	left, center, right, justify	Ausrichtung	<code><p align="right">Rechtsbündige Ausrichtung des Absatzes</p></code>
	class bzw. id	freier Name	definiert eine eigene Klasse	<code><p class="zitat" id="zitat">Das ist ein Zitat</p></code>
<code>
</code>			Zeilenumbruch	
<code><div></div></code>			„Container“ auf Absatzebene	<code><div>Das ist ein freier Absatz</div></code>
	class bzw. id	freier Name	definiert eine eigene Klasse	<code><div class="pos1">Ein Abschnitt</div></code>

Tags zur einfachen Textgestaltung

In HTML setzt man die zu gestaltenden Passagen in spitze Klammernpaare, die wohlbekannten Tags. Es gibt unzählige dieser Gestaltungs-Symbole. Hier die wichtigsten in einer Schnellübersicht:

Tags	Erläuterung	Anwendungsbeispiel
<h1></h1> bis <h6></h6>	Überschriftsebenen 1 bis 6	<h3>Überschrift 3</h3>
	<i>bold</i> , fett	Das ist eine fette Angelegenheit.
<i></i>	<i>italics</i> , kursiv	Mit <i>kursiv</i> wird's ziemlich schief!
<u></u>	<i>underlined</i> , unterstrichen	Bitte <u>nicht</u> mehr verwenden.
	<i>strong</i> , stark betont, meist fett dargestellt	Bitte kommen Sie nicht.
<code></code>	für Computerquellcode geeignet	<code>function hallo()</code>
<kbd></kbd>	Text, der über die Tastatur eingegeben werden muss	Tippe einfach <kbd> md knowware</kbd>.
<cite></cite>	Literaturangaben, sie werden meist kursiv dargestellt	siehe auch <cite>Maardt 97</cite>.
<dfn></dfn>	<i>definition</i> , Definitionstext	<dfn>p steht für paragraph</dfn>
<samp></samp>	Text wird so ausgegeben wie eingegeben	<samp>§%&\$&</samp>
<q></q>	für kurze Zitate geeignet, wird meist kursiv dargestellt	<q>Hallo</q> sagte sie.
<tt></tt>	<i>Teletype</i> , Fernschreiber, nichtproportionale Schrift	<tt>Text steht meist in Courier</tt>
	definiert einen -Container, dieser ist ideal für Zeichenformatierung mit den Style Sheets (CSS)	Das ist ein rotes Wort.
<strike></strike>	durchgestrichen	<strike>durchgestrichen</strike>
<big></big>	große Schrift, geeignet für Hervorhebungen	<big>E</big>s war einmal.
<small></small>	kleine Schrift	<small>immer klein</small>
	<i>subscript</i> , Tiefstellung	H₂O
	<i>superscript</i> , Hochstellung	10⁵
	<i>emphatic</i> , hervorgehoben, betont, wird kursiv dargestellt	Es ist wirklich wichtig!

Grafik-Dateien einfügen

Wie Grafiken in HTML-Dokumente eingefügt werden, ist sicher bekannt. Trotzdem hier noch die wichtigsten Grundsätze, da immer wieder Fehler auftreten: Nur die Formate GIF, JPEG oder PNG sind gestattet! Die Dateinamen dürfen keine Umlaute und Leerzeichen enthalten. Beachte Groß- und Kleinschreibung. Ein Verweis auf funktioniert nicht, wenn die Grafik selbst *Tower.gif* heißt! Server-Systeme unterscheiden ganz im Gegensatz zu Windows sehr wohl zwischen Groß- und Kleinschreibung!

Die Grafiken sollten entweder im gleichen Ordner wie die HTML-Datei liegen bzw. der Verweis auf den Unterordner muss korrekt gesetzt werden. Absolute Pfade wie *C:\Eigene Dateien\tower.gif* sind verboten!

Tags	Attribute	mögliche Werte	Erläuterung	Anwendungsbeispiel
			fügt Grafikdatei ein	
	src	Quelle der Datei		
	align		Ausrichtung der Grafik	
		left	linksbündig mit Textfluss	
		right	rechtsbündig mit Textfluss	
	width	Pixel	Breite der Grafik	width="100"
	height	Pixel	Höhe der Grafik	width="100" height="20"
	alt	Text	Alternativtext, wird zur QuickInfo	
	border	Pixel	fügt Rahmen ein, (border="0" ist ohne)	
	vspace	Pixel	zusätzlicher Raum oben und unten	vspace="4"
	hspace	Pixel	zusätzlicher Raum links und rechts	hspace="5"

Linien gestalten

Linien lockern HTML-Dokumente auf. Man kann Ausrichtung, Breite, Farbe und Dicke der Linie bestimmen. Da das Linien-Tag `<hr />` kein End-Tag besitzt, wird es in XHTML intern abgeschaltet.

Tags	Attribute	mögliche Werte	Erläuterung	Anwendungsbeispiel
<code><hr /></code>			erzeugt eine Linie	<code><hr /></code>
	size	Pixel	Dicke der Linie, Voreinstellung 1	<code><hr size="4" /></code>
	width	Pixel, %	Breite in Pixel oder Prozent, 100% ist Voreinstellung	<code><hr width="250" /></code>
	align	center	zentriert, Voreinstellung	
		left	linksbündig	<code><hr width="60%" align="left" /></code>
		right	rechtsbündig	<code><hr width="380" align="right" /></code>
	color	hex. Code, Farbname	Farbe, nur Internet Explorer	<code><hr width="320" size="3" color="blue" /></code>
	noshade	noshade	Linie soll keinen Schatten werfen	<code><hr noshade="noshade" /></code>

Listen gestalten

Mit den folgenden Tags erzeugt man schicke Listen. Dabei kann man sich die verschiedenen Formate und Startwerte aussuchen.

Tags	Attribute	mögliche Werte	Erläuterung	Anwendungsbeispiel
<code></code>			<i>unordered list</i> , erzeugt eine Aufzählung mit Bullets (Aufzählungszeichen)	
	type	disc	Voreinstellung, gefüllte runde Bullets	
		square	erzeugt gefüllte, quadratische Bullets	<code><ul type="square"></code>
		circle	erzeugt hohle, kreisförmige Bullets	<code><ul type="circle"></code>
<code></code>			<i>ordered list</i> , erzeugt eine Nummerierung	
	type	1	Typ der Nummerierung, voreingestellt sind arabische Zahlen	
		A	erzeugt Großbuchstaben	<code><ol type="A"></code>
		a	erzeugt Kleinbuchstaben	<code><ol type="a"></code>
		I	große römische Ziffern (großes i)	<code><ol type="I"></code>
		i	kleine römische Ziffern (kleines i)	<code><ol type="i"></code>
	start	Ziffer	Startwert, Voreinstellung ist 1	<code><ol type="1" start="2"></code>
		Wert		<code><ol type="A" start="F"></code>
<code></code>			<i>list</i> , einzelner Listeneintrag	<code>Katzen</code>

Font-Tag

Bis vor kurzem galt das ``-Tag als Wundermittel zum Gestalten von Text. Inzwischen wurde es durch die Style Sheets abgelöst. Das ``-Tag ist in XHTML zwar nicht mehr zulässig, wird aber aus „Kompatibilitätsgründen“ immer noch gebraucht. Schließlich funktioniert es auch mit alten Browsern, was man von Style Sheets leider nicht behaupten kann.

Tags	Attribute	Werte	Erläuterung	Anwendungsbeispiel
<code></code>			legt Schrifteigenschaften fest	
	face	Schriftart	definiert eine Schriftart	<code>Beispiel</code>
	size	1 bis 7 (Normal=3) oder rel. Werte wie -1, -2, +1, +2 (Normal=0)	definiert die Schriftgröße	<code>3 Stufen größer</code>
	color	hex. Wert oder Farbname	definiert die Schriftfarbe	<code>Das Blaue vom Himmel</code>

Klang-Dateien und Videos einfügen

Hier nun verschiedene Möglichkeiten, Mediadateien in HTML einzubinden. Dabei ist es egal, ob es sich um Midi-, Wav-, MP3-Sounds, AVI- oder Quicktime-Filmchen handelt. Das Tag `<bgsound>` zum Einbinden eines Hintergrundsounds ist zwar bequem, wird aber nur vom Internet Explorer unterstützt. Dieses Tag ist in XHTML nicht mehr zulässig. Die Alternative ist `<embed>`. Das folgende Beispiel fügt eine Klangdatei als Hintergrund-sound per `<embed>` ein, Breite und Höhe sind nur als „Platzhalter“ gedacht:

```
<embed src="song.mid" width="150" height="40" autostart="true" loop="false" hidden="true" />
```

Achtung: Media-Dateien kannst du auch per Link `` einbinden!

Tags	Attribute	mögliche Werte	Erläuterung	Anwendungsbeispiel
<code><bgsound /></code>			spielt beim Aufruf des Dokuments eine Sounddatei ab	
	src	Name der Klangdatei	geht nur im Internet Explorer, Tag ist kein offizieller Standard	<code><bgsound src="lied.mid" /></code>
<code><embed /></code>			fügt eine Mediadatei ein	
	src	Dateiname	Quelle der Datei	<code><embed src="lied.mid" /></code>
	width, height	Pixel	Höhe bzw. Breite des Players	<code><embed src="film.avi" width="150" height="75" /></code>
	autostart	true, false	gibt an, ob Stück automatisch gespielt wird	<code><embed src="lied.mp3" autostart="false" /></code>
	loop	true, false	gibt an, ob im Dauermodus abgespielt wird, Voreinstellung ist true	<code><embed src="lied.mp3" autostart="false" loop="false" /></code>
	hidden	true, false	gibt an, ob Player angezeigt oder versteckt wird, z.B. für Hintergrund-sound	<code><embed src="lied.mid" hidden="true" /></code>

Lauf-text einfügen und gestalten

Wer hätte das gedacht: Alles rollt, und das nicht nur auf Inline-Skatern. Das Tag `<marquee>` wird aber vom Netscape-Browser (und von Opera) nicht unterstützt und sollte deshalb nur mit Bedacht eingesetzt werden.

Tags	Attribute	mögliche Werte	Erläuterung	Anwendungsbeispiel
<code><marquee></code> <code></marquee></code>			definiert Lauf-text	<code><marquee>Alles rollt</marquee></code>
	loop	Ziffer	Anzahl der Wiederholungen, die Voreinstellung ist unendlich	<code><marquee loop="5">rollt 5x</marquee></code>
	behavior		definiert das Bewegungsverhalten	
		scroll	Voreinstellung, einfaches Durchlaufen	behavior="scroll"
		alternate	rollt hin und her	behavior="alternate"
		slide	reinrollen und stehenbleiben	behavior="slide"
	direction		gibt die Richtung an	
		left	Voreinstellung, von links	direction="left"
		right	von rechts	direction="right"
		down	nach unten	direction="down"
		up	nach oben	direction="up"
	bgcolor	hex. Wert o. Farbname	Hintergrundfarbe des Lauf-textes	bgcolor="yellow"
	height	Pixel, Prozent	Höhe des Lauf-textes	height="100"
	width	Pixel, Prozent	Breite des Lauf-textes	width="20%"
	scrollamount	Ziffer	gibt an, um wieviel Pixel das Objekt bei jedem Animationsschritt bewegt wird	scrollamount="25"
	scrolldelay	in ms	Dauer der Pause zwischen zwei Animationsschritten in Millisekunden	scrolldelay="1000"

Tags für Formulare

Formulare sind in HTML sehr wichtig. So kann man dem Surfer Informationen entlocken und sich diese zuschicken lassen. Entweder die Infos werden über ein CGI-Programm (PHP, Perl) auf dem Server (Formmailer, Datenbank) abgewickelt oder aber direkt per E-Mail versandt. CGI ist auf jeden Fall das Beste.

Hier die Grundgestalt eines Formulars, welches per E-Mail verschickt wird. Setze statt *email* deine E-Mail-Adresse ein.

Du kannst mit *?subject=Betrefftext* auch einen Betreff eintragen. Das Attribut *enctype="text/plain"* sorgt dafür, dass der Formularinhalt direkt im Body der Mail erscheint und nicht als angehängte Datei.

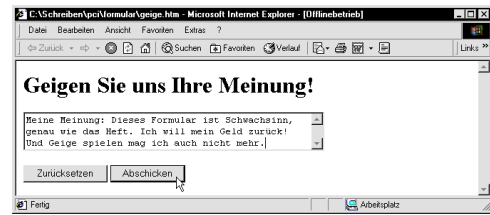
Das folgende Formularbeispiel besteht aus einem Texteingabefeld mit 3 Zeilen und 50 Zeichen je Zeile. Ein Zeilenumbruch ist möglich. Du bekommst den Inhalt des Formulars an deine E-Mail-Adresse geschickt.

Soll das Formular statt dessen per CGI ausgewertet werden, setzt du hinter *action=* den Pfad zum CGI-Skript.

```
<form action="mailto:email@server.de?subject=Betrefftext" method="post" enctype="text/plain">
<textarea name="Kommentar" rows="3" cols="50" wrap="soft" name="Kommentar"></textarea>
<p><input type="reset" value="Zurücksetzen" />
<input type="submit" value="Abschicken" /></p>
</form>
```

Und hier nun eine Übersicht über die Tags, Attribute und Werte, die du in Formularen verwenden kannst.

Tag	Attribute	Werte	Erläuterung	Anwendungsbeispiel
<form> </form>			umschließt ein Formular	
	action	Pfad zu CGI-Skript	CGI-Auswertung (PHP, Perl ...) ist auf jeden Fall die beste Variante	action="http://www.xyz.de/formmail.php"
		E-Mail-Adresse	Schlüsselwort <i>mailto</i> angeben	action="mailto:info@xyz.de"
	method	post oder get	Art der Übermittlung	method="post"
<input />	type		definiert ein Formularelement	
		text	Texteingabefeld	<input type="text" />
		password	Feld zur Passworteingabe	<input type="password" />
		radio	Radioknopf (Einfachauswahl)	<input type="radio" name="a" value="1" />
		checkbox	Kontrollkästchen (Mehrfachauswahl)	<input type="checkbox" name="check" value="0" />
		reset	Reset-Button	<input type="reset" value="Reset" />
		submit	Submit-Button (Abschicken)	<input type="submit" value="Go" />
		button	Schaltfläche	<input type="button" value="Go" onClick="go()" />
		hidden	verstecktes Formularfeld	<input type="hidden" name="1" value="1" />
<textarea> </textarea>			mehrzeiliges Texteingabefeld	
	rows, cols	Wert	Anzahl der Zeilen, Länge	<textarea name="a" rows="2" cols="70"></textarea>
	wrap	soft	Zeilenumbruch, Standard ist leider <i>off</i> (bedeutet kein Umbruch)	<textarea name="a" rows="2" cols="70" wrap="soft"></textarea>
<select> </select>			definiert Auswahlfeld	
	size	Wert	definiert Zahl der anzuzeigenden Felder, Voreinstellung 1 (Klappliste)	<select name="auswahl" size="1">
<option> </option>			Auswahlfeld im <select>-Tag	<option>Seite 1</option>



Anker und Hyperlinks

Über Anker und Hyperlinks bist du sicher schon umfassend informiert. Sie sind praktisch das A und O bei der ganzen „Homepage“. Vergiss nicht, dass du auf diese Weise sogar Grafik- und Multimedia-Dateien in deine Dokumente einbinden kannst. Niemand hindert dich beispielsweise daran, dem Surfer über `Mein MP3-Song` eine MP3-Datei oder ein AVI-Filmchen anzubieten.

Tags	Attribute	Werte	Erläuterung	Anwendungsbeispiel
<code><a></code>	href	Quelle der Datei (URL)	verweist auf eine interne oder externe Datei	<code>...</code>
				<code>...</code>
		Ankername	verweist auf einen internen Anker	<code>...</code>
		mailto:E-Mail-Adresse	erzeugt einen einfachen E-Mail-Link	<code>Mail</code>
	name bzw. id	Ankername	definiert einen internen Anker	<code>...</code>

Tags für Tabellen

Wie gelingt exaktes Layout? Hier das Grundgerüst einer unsichtbaren Tabelle, die 200 Pixel breit ist.

```
<table border="0" width="200">
<tr><td width="150">Zelle 1</td><td width="50">Zelle 2</td></tr>
<tr><td width="150">Zelle 3</td><td width="50">Zelle 4</td></tr>
</table>
```

Und hier die wichtigsten Tabellen-Tags im Überblick.

Tags	Attribute	Werte	Erläuterung	Anwendungsbeispiel
<code><table></table></code>			umschließt und definiert eine Tabelle	
	border	Pixel	definiert Rahmen, Voreinstellung =0	<code><table border="3"></code>
	bgcolor	hex. Wert, Farbname	Hintergrundfarbe	<code><table border="2" background="silver"></code>
	background	Grafikname	bindet Hintergrundgrafik ein	<code><table background="bild3.gif"></code>
	align	left, right, center	Ausrichtung, nur bei left/right findet Textfluss statt, Voreinstellung ist ohne	<code><table align="right"></code>
	width	Pixel, %	Breite absolut oder relativ	<code><table width="70%"></code>
	cellpadding	Pixel	Platz zwischen Zellrand und Zellinhalt, „Polsterung“	<code><table border="1" cellpadding="4"></code>
	cellspacing	Pixel	Abstand zwischen einzelnen Feldern, „Dicke der Wand“	<code><table border="1" cellspacing="3"></code>
<code><tr></tr></code>			definiert eine Zeile	<code><tr><td>Inhalt</td><td>Inhalt</td></tr></code>
<code><td></td></code>			definiert „gewöhnliche“ Zelle	<code><td>5</td></code>
	width	Pixel, %	Breite absolut oder relativ	<code><td width="100"></code>
	align	left, center, right	Ausrichtung, die Voreinstellung ist <i>left</i>	<code><td width="100" align="right"></code>
	valign	top, middle, bottom	Vertikalausrichtung, die Voreinstellung ist <i>middle</i>	<code><td width="100" align="right" valign="top"></code>
	bgcolor	hex. Wert, Farbname	Hintergrundfarbe	<code><td background="#FF0000"></code>
	background	Grafikname	bindet Hintergrundgrafik ein	<code><td background="bild3.gif"></code>
	colspan	Wert	definiert, über wie viele Spalten sich eine Zelle erstreckt	<code><td colspan="2"></code>
	rowspan	Wert	definiert, über wie viele Zeilen sich eine Zelle erstreckt	<code><td rowspan="3"></code>
<code><th></th></code>	siehe <td>		definiert eine Zelle im Tabellenkopf	<code><th>Anzahl</th></code>

Tags für Framesets

Mit Framesets teilst du dein Browserfenster in Rahmen auf. Das können zwei, drei oder auch mehrere Bereiche sein, in die sich jeweils andere HTML-Dokumente einbinden lassen. Baue die Framedefinition in ein HTML-Dokument ein, in dem du `<body></body>` weglässt. Dieses Dokument wird als Frameset bezeichnet.

Der Grundaufbau eines einfachen zweigeteilten Fensters sieht so aus:

```
1 <frameset cols="150,*">
2 <frame src="navi.htm" />
3 <frame src="main1.htm" />
4 </frameset>
```

Zeile 1 legt die Fensteraufteilung fest. Hier sind es Spalten. Die linke Spalte ist 150 Pixel breit, die rechte passt sich dank des *-Symbols der Fensterbreite an.



Beachte, dass dieses Frameset selber nur die „Steuerdatei“ ist. Hier legst du lediglich die Fensteraufteilung fest. Welche Dokumente in welchem Frame angezeigt werden sollen, steht im Beispiel in *Zeile 2* und *3*.

Es sind die Dokumente *navi.htm* bzw. *main1.htm*. Die Seite *navi.htm* würde links, die Seite *main1.htm* dagegen rechts angezeigt werden. Im Beispiel benötigst du also drei Dokumente, das Frameset und die in den Fenstern anzuzeigenden HTML-Dateien!

Die folgende Tabelle gibt dir einen Überblick über die wichtigsten in Frames einzusetzenden Tags und Attribute.

Tags	Attribute	Werte	Erläuterung	Anwendungsbeispiel
<code><frameset></code> <code></frameset></code>			umschließt ein Frameset, nicht die End-Tags vergessen!	
	cols	Pixel, %, *	teilt das Fenster in Spalten (columns) ein	<code><frameset cols="*,50"></code>
	rows	Pixel, %, *	teilt das Fenster in Zeilen (rows) ein	<code><frameset rows="10%,*"></code>
	border	Pixel	gibt Rahmenbreite in Pixeln an, Rahmen werden mit <code>border="0"</code> unterdrückt, Tag ist veraltet, aber trotzdem derzeit noch unentbehrlich	<code><frameset rows="10%,*" border="0"></code>
	frameborder	0, 1 (yes, no)	legt fest, ob Rahmen (mit Effekt) dargestellt werden soll. Ältere Browser verlangen yes und no	<code>frameborder="1"</code>
	bodercolor	hex. Code, Farbname	weist dem Rahmen eine bestimmte Farbe zu	<code><frameset="10,*,20" border="10" bordercolor="red"></code>
<code><frame /></code>			zeigt eine Seite im Frameset an	
	src	Dateiname	gibt Quelle (source) für den Frame an	<code><frame src="navi.htm" /></code>
	scrolling	auto, yes, no	gibt an, ob Rollbalken angezeigt oder unterdrückt werden sollen	<code><frame src="navi.htm" scrolling="no" /></code>
	noresize	noresize	unterdrückt das Anpassen der Rahmengröße durch den Benutzer	<code><frame src="navi.htm" noresize="noresize" /></code>
	marginwidth	Pixel	Randbreite, Abstand zwischen rechten und linken Fensterrand	<code><frame src="navi.htm" marginwidth="0" /></code>
	marginheight	Pixel	Randhöhe, Abstand zwischen oberem und unterem Fensterrand	<code><frame src="navi.htm" marginheight="5" /></code>
<code><noframes></code> <code></noframes></code>			enthält Alternativbeschreibung für nicht-framefähige Browser	<code><noframes>Sorry, dein Browser versteht keine Frames</noframes></code>

Style Sheets: Die wichtigsten Attribute von CSS

Zur Erinnerung: Mit Styles Sheets steuerst du exakt und bequem die einzelnen Eigenschaften der Tags. Du setzt die Style-Sheet-Anweisungen intern zwischen `</title>` und `</head>` oder lagerst sie extern aus. Die Style-Sheet-Syntax ähnelt der von JavaScript, die einzelnen Blöcke werden in geschweifte Klammern eingehüllt, mehrere Eigenschaften durch Komma getrennt. Jede Zeile wird durch ein Semikolon abgeschlossen.

Das folgende Style Sheet ändert die Schriftart von `<h1>` und `<h2>` (Zeile 3) und Schriftart, -größe und Farbe von normalen Textabsätzen `<p>` (Zeile 4). Wie du in Zeile 3 siehst, kannst du mehrere Tags auf einmal „bearbeiten“ (hier `<h1>` und `<h2>`). Reihe diese Eigenschaften lediglich durch Kommas auf.

```
1 <style type="text/css">
2 <!--
3 h1, h2 { font-family: Arial; }
4 p { font-family: Arial; font-size: 12pt; color: blue; }
5 //-->
6 </style>
```

Externe Style Sheets dagegen werden als Textdatei mit der Endung `css` gespeichert. Setze hier nur die Stil-Anweisungen hinein, nimm im Beispiel nur Zeile 3 und 4. Wenn du das externe Style Sheet `standard.css` nennst, sieht der Verweis in der HTML-Datei so aus: `<link rel="stylesheet" type="text/css" href="standard.css">`.

Attribute zur Steuerung von Schrifteigenschaften

Attribute	Werte	gestaltet ...	Anwendungsbeispiel
font-family	Schriftname	Schriftart	font-family: Helvetica;
	serif	Schriftfamilie, Serifenschrift allgemein	font-family: Times, serif;
	sans-serif	Schriftfamilie, serifenlose Schrift	font-family: Arial, sans-serif;
	cursive	Schriftfamilie, kursiv-geschwungene Schrift	font-family: cursive;
	monospace	Schriftfamilie, nichtproportionale Schrift	font-family: monospace;
	fantasy	Schriftfamilie, Phantasieschrift	font-family: Western, fantasy;
font-size	pt, cm usw.	Schriftgröße	font-size: 14pt;
font-style		Schriftschnitt	
	normal	ohne Eigenschaft	
	italic	kursiv	font-style: italic;
	oblique	Hohlschrift	font-style: oblique;
font-weight	100 bis 900	Dicke (gemessen in Hunderterschritten)	font-weight: 500;
	normal	entspricht Wert 400	font-weight: normal;
	bold	entspricht Wert 700	font-weight: bold;
font-variant		Schriftvarianten	
	normal	ohne Eigenschaft	
	small-caps	Kapitälchen (kleine Großbuchstaben)	font-variant: small-caps;
word-spacing	pt, cm	Wortzwischenraum	word-spacing: 5pt;
letter-spacing	pt, cm	Buchstabenzwischenraum (für Sperrschrift)	letter-spacing: 0,2cm;
text-transform		Groß- und Kleinschreibung	
	none	bleibt unverändert	
	uppercase	Großbuchstaben	text-transform: uppercase;
	lowercase	Kleinbuchstaben	text-transform: lowercase;
text-decoration		Hervorhebungen	
	none	schaltet Hervorhebung ab	text-decoration: none;
	underline	unterstrichen	text-decoration: underline;
	overline	Strich darüber	text-decoration: overline;
	line-through	durchgestrichen	text-decoration: line-through;
	blink	blinkender Text	text-decoration: blink;
	capitalize	erster Buchstabe im Wort groß	text-transform: capitalize;

Attribute zur Steuerung von Listen

In Style Sheets stehen dir etliche Optionen zur Gestaltung deiner Listen zur Verfügung. Besonders nett ist beispielsweise die Möglichkeit, eine kleine Grafik als Aufzählungszeichen zu nutzen. Die folgende Zeile bindet die Datei *kreuz.gif* als Aufzählungszeichen ein. Diese sollte dabei nur wenige Pixel breit und hoch sein.

```
ul { list-style-image: url(kreuz.gif); font-family: Arial, Helvetica; font-size: 14pt; }
```

Hier die wichtigsten Möglichkeiten im Überblick:

Attribute	Werte	gestaltet ...	Anwendungsbeispiel
list-style-type		Erscheinungsform der Listen	
(für ol)	decimal	Voreinstellung, (1, 2, 3, ...)	
	lower-alpha	Kleinbuchstaben (a, b, c, ...)	ol { list-style-type: lower-alpha; }
	upper-alpha	Großbuchstaben (A, B, C, ...)	ol { list-style-type: upper-alpha; }
	lower-roman	römische Zahlen (i, ii, iii, ...)	ol { list-style-type: lower-roman; }
	upper-roman	große römische Zahlen (I, II, III, ...)	ol { list-style-type: upper-roman; }
(für ul)	disc	Voreinstellung (gefüllter Kreis)	
	circle	hohler Kreis	ul { list-style-type: circle; }
	square	Quadrat	ul { list-style-type: square; }
list-style-image	url(Dateiname)	bindet Grafik als Aufzählungszeichen ein	ul { list-style-image: url(bild.gif); }
list-style-position	outside	Voreinstellung, Text wird eingezogen	
	inside	Text bleibt bündig am linken Rand, Aufzählungszeichen wird „integriert“	ol { list-style-position: inside; }

Textausrichtung

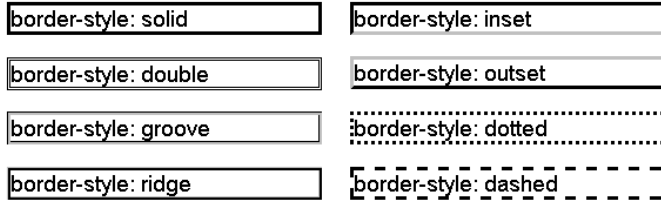
Schau in die nächste Übersicht! Am interessantesten sind sicher die Attribute *text-align*, *text-indent* und *line-height*. Damit justierst du Absatzausrichtung, Erstzeileneinzug und Zeilenhöhe. Mit *line-height: 1.5* hättest du einen eineinhalbfachen Zeilenabstand. Beachte, dass Dezimalbrüche in CSS durch Punkt getrennt werden, nicht durch ein Komma, wie wir Deutschen das so lieben.

Attribute	Werte	gestaltet ...	Anwendungsbeispiel
text-align		Absatzausrichtung	
	left	linksbündig, Normaleinstellung	text-align: left;
	center	zentriert	text-align: center;
	right	rechtsbündig	text-align: right;
	justify	Blocksatz	text-align: justify;
text-indent	pt, cm usw.	Erstzeileneinzug	text-indent: 1cm;
line-height	pt, cm usw.	Zeilenhöhe	line-height: 14pt;
	relativer Wert	Zeilenabstand relativ zur Schriftgröße	line-height: 1.2;
vertical-align		Vertikalausrichtung relativ zur Grundlinie, auf der die Schrift ausgerichtet wird	
	baseline	an Grundlinie, Normaleinstellung	vertical-align: baseline;
	super	Hochstellung	vertical-align: super;
	top	an Oberkante der Zeile	vertical-align: top;
	text-top	an Oberkante des Texts	vertical-align: text-top;
	middle	mittig	vertical-align: middle;
	text-bottom	Unterkante des Texts	vertical-align: text-bottom;
	bottom	Unterkante der Zeile	vertical-align: bottom;
	sub	Tiefstellung	vertical-align: sub;

Gestaltung von Rändern und Rahmen

Besonders viele Möglichkeiten bieten dir Style Sheets bei der Gestaltung von Rändern und Rahmen. Leider werden viele Rahmenformate nur von den allerneuesten Browsern unterstützt. Die Browser, die derzeit alle unter *border-style* aufgelisteten Rahmeneffekte darstellen können, sind z.B. IE und Netscape 6, und Opera 5/6.

(Wie man Scrollbars einfärbt und individuelle Mauscursor nutzt, erfährst du auf www.jchanke.de/homepage.)



Mein Tipp: Probiere es einfach einmal aus:

Attribute	Werte	gestaltet ...	Anwendungsbeispiel
margin	pt, px, cm usw.	Außenabstand, äußerer „Leerraum“	body { margin: 1cm; }
margin-top		Rand von oben	p { margin-top: 10px; }
margin-bottom		Rand von unten	h3 { margin-bottom: 3pt; }
margin-left		Rand von links	img { margin-left: 50px; }
margin-right		Rand von rechts	p.zitat { margin-right: 10pt; }
padding	pt, px, cm usw.	Innenabstand, Füllung bzw. „Polsterung“	p { padding: 2cm; }
padding-top		Innenabstand von oben	body { padding-top: 15px; }
padding-bottom		Innenabstand von unten	h3 { padding-bottom: 1.3cm; }
padding-left		Innenabstand von links	p.neu { padding-left: 5px; }
padding-right		Innenabstand von rechts	img { padding-right: 5cm; }
border-top		Rahmenbreite oben	h1 { border-top: 0.2cm; }
border-bottom		Rahmenbreite unten	h2 { border-bottom: 4px; }
border-left		Rahmenbreite links	p.zitat { border-left: 5px; }
border-right		Rahmenbreite rechts	pre { border-right: 12px; }
border-style		Rahmenstil	
	solid	durchgezogen	p { border-style: solid; }
	double	doppelte Rahmenlinie	h1 { border-style: double; }
	groove	3-D-Effekt mit „Furche“ bzw. „Rille“	p { border-style: groove; }
	ridge	3-D-Effekt mit erhöhtem Rand, mit „Rücken“	p.neu { border-style: ridge; }
	inset	3-D-Effekt mit vertieftem Inhalt	div { border-style: inset; }
	outset	3-D-Effekt mit erhöhtem Inhalt („Schaltfläche“)	span { border-style: outset; }
	dotted	punktierter Rand	h2, h3 { border-style: dotted; }
	dashed	mit Strichen umzogen	pre { border-style: dashed; }
border-top-style	siehe oben	Rahmenstil nur oben	border-style-top: ridge;
border-bottom-style	siehe oben	Rahmenstil nur unten	
border-left-style	siehe oben	Rahmenstil links	
border-right-style	siehe oben	Rahmenstil rechts	
border-width	px, cm usw.	Rahmenbreite (bzw. Abstand zwischen Rahmenlinien), auch hier möglich: border-top-width, usw.	<div style="border style: outset; border-width: 10px;">
border-color	hex. Wert, Name, rgb	Rahmenfarbe, auch hier wieder möglich: border-top-color usw.	p { border: 7px; border-color: rgb(255, 0, 0); }

Farbe und Hintergrundgrafik

Aus der folgenden Tabelle sind sicher *color* (Elementfarbe) und *background-color* (Hintergrundfarbe) am interessantesten! Du möchtest einen eigenen Textabsatz erstellen, der gelb hinterlegt ist? Dann definiere eine eigene Klasse für den Absatz, nenne diese beispielsweise *gelb*.

```
<p class="gelb">Das ist ein gelb hinterlegter Textabsatz.</p>
```

Im Style Sheet selbst füllst du diese Klasse folgendermaßen mit Leben, hier gilt die „Punktnotation“.

```
p.gelb { background-color: yellow; }
```

Außerdem findest du in CSS verschiedene Möglichkeiten, Hintergrundgrafiken einzubinden.

Attribute	Werte	gestaltet ...	Anwendungsbeispiel
color	hex. Wert bzw. Farbname	Farbe eines Elementinhalts, in der Regel Text	body { color: blue; }
background-color	hex. Wert bzw. Farbname	Hintergrundfarbe, z. B. für „Schattierungseffekt“	h1, h2 { background-color: silver; }
background-image	url(Grafikname)	bindet Hintergrundgrafik ein	body { background-image: url(bild.gif); }
background-repeat		legt fest, wie die Grafik ausgerichtet werden soll	
	repeat	Voreinstellung, Kachelung	background-repeat: repeat;
	repeat-x	Wiederholung nur horizontal, „auf der x-Achse“	background-repeat: repeat-x;
	repeat-y	Wiederholung nur vertikal, „auf der y-Achse“	background-repeat: repeat-y;
	no-repeat	keine Wiederholung	background-repeat: no-repeat;
background-attachment		gibt an, ob Hintergrundgrafik beim Scrollen mitbewegt wird	
	scroll	Voreinstellung, Browser bewegt Hintergrundgrafik mit	background-attachment: scroll;
	fixed	Hintergrundgrafik bleibt feststehend	background-attachment: fixed;
background-position	px, cm usw.	Position der Hintergrundgrafik, zuerst links, dann oben	background-position: 100px 10px

Absolute Positionierung (wichtig für DHTML)

Der große Knüller von CSS: Du kannst Elemente (Text, Grafiken usw.) absolut auf der Seite platzieren. Am liebsten packt man die Elemente in `<div></div>`-Container, also „leere“ Tags auf Absatzebene. Du willst einen Text exakt 50 Pixel von oben und 100 Pixel von links ausrichten? Dann schreibe:

```
div { position: absolute; top: 50px; left: 100px; }
```

Im HTML-Dokument selbst nutzt du dann einfach diese `<div></div>`-Tags.

```
<div>Ich bin ganz exakt ausgerichtet</div>
```

Hier eine Übersicht:

Attribute	Werte	gestaltet ...	Beispiel
position		Positionierung auf der Seite	
	static	Voreinstellung, normale Positionierung „im Kontext“	
	absolute	absolute Positionierung in Bezug auf linke obere Ecke	position: absolute; top: 10px; left 20px;
	relative	ermöglicht Positionierung relativ zum vorherigen Element	position: relative; top: 20px;
top	px, pt ...	Abstand vom oberen Rand	
left	px, pt ...	Abstand vom linken Rand	
width	px, pt ...	Breite des Elements	width: 150px;
height	px, pt ...	Höhe des Elements	height: 80px;
visibility		Sichtbarkeit	
	visible	Voreinstellung, Element ist sichtbar	
	hidden	Element wird versteckt	visibility: hidden;
z-index	1, 2, 3 ...	legt Elementreihenfolge fest	z-index: 2;

Maßeinheiten, Farben und Sonderzeichen

Die wichtigsten Farben

engl. Name	dt. Name	hexadezimaler Wert
black	schwarz	#000000
silver	hellgrau	#C0C0C0
gray	grau	#808080
white	weiß	#FFFFFF
maroon	kastanienbraun	#800000
red	rot	#FF0000
purple	lila	#800080
fuchsia	helllila	#FF00FF
green	dunkelgrün	#008000
lime	hellgrün	#00FF00
olive	olivgrün	#808000
yellow	gelb	#FFFF00
navy	dunkelblau	#000080
blue	blau	#0000FF
teal	blaugrün	#008080
aqua	himmelblau	#00FFFF

RGB-Farbcodes für CSS

In CSS gibt es auch eine so genannte RGB-Methode zum Zuweisen der Farbe.

Schließlich setzen sich die Farben am Bildschirm aus den Bestandteilen Rot, Grün und Blau zusammen. Schreibe also einfach nach dem Schema

```
rgb(ROT, GRÜN, BLAU).
```

Dabei sind die Ziffern 0 bis 255 oder auch Prozentwerte erlaubt. Für die Farbe Grün schreibst du

```
rgb(0, 255, 0) bzw. rgb(0%, 100%, 0%).
```

Umschreibung von Sonderzeichen

Zeichen	Kodierung	Zeichen	Kodierung
ä	ä	»	»
Ä	Ä	«	«
ö	ö	„	“
Ö	Ö	“	”
ü	ü	– (Ged.strich)	–
Ü	Ü	€ (EURO)	€
ß	ß	©	©
<	<	®	®
>	>	™	™
&	&	. (Mittelpunkt)	·
"	"	¶	¶
' (Apostroph)	'	• (Bullet)	•
erzw. Leerz.	 	§	§

Die Ländercodes für Zeichensätze

ISO-Standard	Ländercode
UTF-8 (Unicode)	universeller Zeichensatz, weltweit
ISO-8859-1 (Latin-1)	Westeuropa, Lateinamerika
ISO-8859-2 (Latin-2)	Zentral- und Osteuropa
ISO-8859-3 (Latin-3)	Südosteuropa
ISO-8859-4 (Latin-4)	Skandinavien, Baltikum
ISO-8859-5	Kyrillisch
ISO-8859-6	Arabisch
ISO-8859-7	Griechisch
ISO-8859-8	Hebräisch
ISO-8859-9	Türkisch
ISO-8859-10	Lappländisch, Nordisch, Eskimosprachen
EUC-JP <i>oder</i> Shift_JIS	Japanisch

Hover-Links

Hier zeige ich dir noch einmal die vier Pseudo-Klassen für Hyperlinks:

- *A:link* ist der normale Hyperlink
- *A:hover* ist der Hyperlink während des Darüberfahrens mit der Maus,
- *A:active* ist der aktive Link
- *A:visited* ist der besuchte Hyperlink

Du möchtest einen Link während des Darüberstreichens hervorheben? Dann schreibst Du im Style Sheet:

```
A:hover {text-decoration: none; color: red;}
```

Maßeinheiten für CSS

Hier ein Überblick über die Maßeinheiten von CSS.

- **Pixel, Endung *px***
- Millimeter, Endung *mm*
- Zentimeter, Endung *cm*
- Zoll, Endung *in* von inch, 1 Zoll entspricht 2,54 cm
- **typographischer Punkt, Endung *pt***, entspricht 1/72 inch
- typographischer Pica-Punkt, Endung *pc*, entspricht 12 pt
- relativ in Bezug auf übergeordnetes Element, Endung *em*, z.B. *font-size 1.3em*

Die wichtigsten Einheiten sind Pixel (z. B. *14px*), und Punkt (*12pt*), *em* ist nicht zu empfehlen.

Bei ungeraden Werten (0.5) musst du als Dezimaltrenner einen Punkt statt des Kommas setzen.