

HTML



10. Auflage

Stefan Münz / Clemens Gull

Der
neue Münz:
Seit über 15 Jahren
das unerreichte
deutschsprachige
Standardwerk
zu HTML.

HTML5

Handbuch

- So setzen Sie anspruchsvolle Web-Layouts mit HTML5 und CSS um
- Webseiten für jedes Endgerät: Media Queries für Mobile Devices
- JavaScript-Framework jQuery für dynamische Webseiten nutzen
- Umfangreicher Referenzteil für HTML und CSS zum Nachschlagen

Stefan Münz / Clemens Gull

HTML5 Handbuch

Stefan Münz / Clemens Gull

HTML5

Handbuch

- So setzen Sie anspruchsvolle Web-Layouts mit HTML5 und CSS um
- Webseiten für jedes Endgerät: Media Queries für Mobile Devices
- JavaScript-Framework jQuery für dynamische Webseiten nutzen
- Umfangreicher Referenzteil für HTML und CSS zum Nachschlagen

Bibliografische Information der Deutschen Bibliothek

Die Deutsche Bibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte Daten sind im Internet über <http://dnb.ddb.de> abrufbar.

Alle Angaben in diesem Buch wurden von den Autoren mit größter Sorgfalt erarbeitet bzw. zusammengestellt und unter Einschaltung wirksamer Kontrollmaßnahmen reproduziert. Trotzdem sind Fehler nicht ganz auszuschließen. Der Verlag und die Autoren sehen sich deshalb gezwungen, darauf hinzuweisen, dass sie weder eine Garantie noch die juristische Verantwortung oder irgendeine Haftung für Folgen, die auf fehlerhafte Angaben zurückgehen, übernehmen können. Für die Mitteilung etwaiger Fehler sind Verlag und die Autoren jederzeit dankbar. Internetadressen oder Versionsnummern stellen den bei Redaktionsschluss verfügbaren Informationsstand dar. Verlag und die Autoren übernehmen keinerlei Verantwortung oder Haftung für Veränderungen, die sich aus nicht von ihnen zu vertretenden Umständen ergeben. Evtl. beigefügte oder zum Download angebotene Dateien und Informationen dienen ausschließlich der nicht gewerblichen Nutzung. Eine gewerbliche Nutzung ist nur mit Zustimmung des Lizenzinhabers möglich.

© 2014 Franzis Verlag GmbH, 85540 Haar bei München

Alle Rechte vorbehalten, auch die der fotomechanischen Wiedergabe und der Speicherung in elektronischen Medien. Das Erstellen und Verbreiten von Kopien auf Papier, auf Datenträgern oder im Internet, insbesondere als PDF, ist nur mit ausdrücklicher Genehmigung des Verlags gestattet und wird widrigenfalls strafrechtlich verfolgt.

Die meisten Produktbezeichnungen von Hard- und Software sowie Firmennamen und Firmenlogos, die in diesem Werk genannt werden, sind in der Regel gleichzeitig auch eingetragene Warenzeichen und sollten als solche betrachtet werden. Der Verlag folgt bei den Produktbezeichnungen im Wesentlichen den Schreibweisen der Hersteller.

Programmleitung: Dr. Markus Stäuble

Satz: DTP-Satz A. Kugge, München

art & design: www.ideehoch2.de

Druck: C.H. Beck, Nördlingen

Printed in Germany

ISBN 978-3-645-60345-4

Inhaltsverzeichnis

1	Intro	15
1.1	Zum vorliegenden Buch	15
1.1.1	Zielgruppen und Ziele	15
1.1.2	Aufbau und Vorgehensweise.....	15
1.1.3	Verwendete Symbole	16
1.2	Website-Erstellung heute	17
1.2.1	Website-Typologie	18
1.3	Web-Technologien	21
1.3.1	HTML / XHTML.....	21
1.3.2	CSS.....	23
1.3.3	Clientseitiges Scripting	25
1.3.4	Serverseitiges Scripting und Datenbankbindung.....	26
1.3.5	Flash.....	27
1.3.6	XML	27
1.4	Software zum Erstellen von Websites	29
1.4.1	Code-Editoren	30
1.4.2	Web-Browser	33
1.4.3	Desktop-Tools.....	34
1.4.4	Content-Management-Systeme	37
1.5	Backgrounds.....	39
1.5.1	Die Arbeit des W3-Konsortiums.....	39
1.5.2	Quellen, Dokumentation und Hilfe im Web.....	41
2	HTML-Dokumente	43
2.1	HTML editieren.....	43
2.1.1	Grundsätzliche Grammatik von HTML.....	43
2.1.2	Whitespace und interpretierte Bereiche in HTML	47
2.1.3	Kommentare in HTML	49
2.1.4	Dateikonventionen.....	51
2.1.5	Referenzierung.....	52
2.2	Zeichenkodierung in HTML.....	56
2.2.1	Numerische und benannte Entities	57
2.3	Dokument-Grundgerüst.....	58
2.3.1	HTML-Dokumenttypen.....	60
2.3.2	XHTML-Dokumenttypen.....	62
2.4	Backgrounds.....	62
2.4.1	Computer und Zeichenkodierung	62
2.4.2	Quirks-, Standards- und Kompatibilitätsmodus.....	65

3	Dokument-Kopfdaten.....	69
3.1	Elemente für Kopfdaten.....	69
3.1.1	Titel eines HTML-Dokuments	69
3.1.2	Meta-Angaben	70
3.1.3	Logische Verlinkung.....	75
3.1.4	Style- und Script-Einbindung.....	82
3.1.5	Basis-URLs.....	84
3.2	Backgrounds.....	85
3.2.1	RDF (Resource Description Framework)	85
3.2.2	Das Dublin-Core-Metadaten-Set.....	86
3.2.3	RDF, Dublin-Core und HTML.....	87
4	Textstrukturierung	89
4.1	Webseiten-Bereiche (Sectioning)	89
4.1.1	Dokumentkörper	89
4.1.2	Abschnitte	90
4.1.3	Header und Footer	91
4.1.4	Navigation und Seitenleisten	93
4.1.5	Überschriften	95
4.1.6	Artikel	97
4.1.7	Autorenangaben	98
4.2	Gruppierende Elemente (Grouping).....	100
4.2.1	Der Inhaltsbereich.....	100
4.2.2	Textabsätze	101
4.2.3	Zeilenumbrüche.....	102
4.2.4	Geordnete und ungeordnete Listen	103
4.2.5	Beschreibungslisten	108
4.2.6	Blockzitate.....	109
4.2.7	Abbildungen	111
4.2.8	Präformatierte Bereiche (mit Leerraum).....	112
4.2.9	Trennlinien.....	114
4.2.10	Allgemeine Bereiche	115
4.3	Textauszeichnungen (Textlevel-Semantics)	116
4.3.1	Elemente für die Textauszeichnung.....	116
4.4	Elemente für Änderungsmarkierungen	128
4.5	Mathematische Formeln.....	129
4.6	Backgrounds.....	133
4.6.1	Content-Modelle	133
4.6.2	Arbeiten mit unbekanntenen Elementen	136
5	Tabellen.....	141
5.1	Aufbau von Tabellen	141
5.1.1	Einfache Tabelle.....	141
5.1.2	Tabelle mit Bereichen für Kopf, Körper und Fuß.....	143
5.1.3	Spalten und Spaltengruppen	146

5.1.4	Tabellenbeschriftung	148
5.1.5	Tabellenzellen verbinden	149
5.2	Zusätzliche Auszeichnungen für Tabellen.....	152
5.2.1	Zusammenfassung einer Tabelle.....	152
5.2.2	Expliziter Kopfzellenbezug	155
5.3	Backgrounds.....	157
5.3.1	Tabellenlayouts	157
6	Hyperlinks	159
6.1	Aufbau von Hyperlinks	159
6.1.1	Links zu beliebigen Zielen.....	159
6.1.2	Download-Verweise	161
6.1.3	Links zu beliebigen Ressourcen	162
6.1.4	E-Mail-Links	164
6.1.5	Links und Anker	167
6.2	Zusatzinformationen in Hyperlinks.....	170
6.2.1	Angaben zum Linkziel	170
6.2.2	Typisierte Links	171
6.2.3	Links mit Ping-Funktion.....	174
6.3	Backgrounds.....	175
6.3.1	Hyperlinks und Navigation	175
6.3.2	Rechtliche Aspekte von Hyperlinks	179
7	Grafik und Multimedia	181
7.1	Pixelgrafik.....	181
7.1.1	Pixelgrafik-Referenzen	181
7.1.2	Image-Maps (verweissensitive Grafiken)	185
7.2	Vektorgrafik	188
7.2.1	SVG-Grafikreferenzen.....	188
7.2.2	SVG-Grafiken (direkt eingebettet).....	190
7.2.3	Basisfunktionen von SVG	192
7.2.4	Generierte Vektorgrafik mit dem canvas-Element	198
7.2.5	Funktions-Set für Canvas-2D-Rendering-Context	201
7.3	Audio- und Video-Ressourcen	209
7.3.1	Eingebettete Audio-Ressourcen	209
7.3.2	Eingebettete Video-Ressourcen.....	213
7.3.3	Zusätzliche Spuren (Tracks)	217
7.4	Aktive Inhalte (Flash, Java und andere)	222
7.4.1	Aktive Inhalte mit dem embed-Element.....	223
7.4.2	Aktive Inhalte mit dem object-Element	224
7.5	Eingebettete Frames	228
7.5.1	Script-Zugriffe auf Ressourcen von fremden Domains	236
7.6	Backgrounds.....	238
7.6.1	Pixelgrafik-Formate	238
7.6.2	Audio- und Video-Codexs.....	241

8	Formulare und Interaktion.....	245
8.1	Aufbau von Formularen	245
8.1.1	Formularbereiche und Formularversand	246
8.1.2	Strukturierung von Formularen	250
8.2	Formular-Feldtypen	252
8.2.1	Einzeilige Eingabefelder.....	253
8.2.2	Mehrzeilige Eingabefelder.....	255
8.2.3	Auswahllisten	257
8.2.4	Eingabefelder mit Datenlisten (Comboboxen)	260
8.2.5	Radiobuttons und Checkboxes	262
8.2.6	Felder für Datei-Uploads	264
8.2.7	Versteckte Formularfelder	267
8.2.8	Schaltflächen (Buttons)	268
8.2.9	Schlüsselgenerierung für sichere Datenübertragung	273
8.3	Eingabekontrolle und Eingabehilfe.....	275
8.3.1	Feldtypen für kontrollierte Eingaben.....	275
8.3.2	Zusätzliche Angaben für die Feldinterpretation	281
8.3.3	Deaktivieren und Nurlesen von Formularfeldern	283
8.3.4	Eingabehilfen bei Formularfeldern	284
8.3.5	Verwendete Schreibrichtung mitsenden.....	288
8.4	Formularkontrolle.....	288
8.4.1	Felder außerhalb von Formularen.....	288
8.4.2	Formulare mit mehreren Submit-Buttons.....	289
8.5	Informationskontrolle	291
8.5.1	Ausgabefelder für Kalkulationen	291
8.5.2	Fortschrittsanzeige.....	293
8.5.3	Metrische Werte.....	296
8.6	Interaktive Elemente	298
8.6.1	Menüs.....	298
8.6.2	Aufklappbare Details	302
8.6.3	Dialogelemente.....	304
8.7	Formularverarbeitung.....	306
8.7.1	Einfacher Form-Mailer in PHP	306
8.7.2	Formulare in Datenbank speichern (PHP/MySQL)	310
8.8	Backgrounds: Formulare	315
8.8.1	Richtext-Eingaben	315
9	Mikrodaten und globale Attribute	319
9.1	Mikrodaten	319
9.1.1	Aufbau von Mikrodaten.....	320
9.1.2	Referenzen zwischen Mikrodaten.....	322
9.1.3	Typisierte Mikrodaten	323
9.2	Anwendung von Mikrodaten	325
9.2.1	vCard-Anwendung (Kontaktdaten).....	325

9.2.2	vEvent-Anwendung (Terminkalenderdaten)	331
9.2.3	Works-Anwendung (Lizensierung von Werken)	336
9.3	Globale Attribute.....	337
9.3.1	Übersicht der globalen Attribute	337
9.3.2	Beispiele für globale Attribute.....	342
9.4	Backgrounds.....	343
9.4.1	Mikrodaten und Mikroformate.....	344
9.4.2	Mikrodaten und RDFa.....	347
9.4.3	Unbekannte Attribute.....	348
10	HTML und XHTML	349
10.1	Unterschiede zwischen HTML und XHTML	349
10.1.1	Unterschied: MIME-Typen	349
10.1.2	Unterschied: Dateinamen	350
10.1.3	Unterschied: Die XML-Deklaration und die Zeichenkodierung	350
10.1.4	Unterschied: Dokumenttyp-Deklaration	351
10.1.5	Unterschied: HTML-Wurzelement mit Namensraumangabe	351
10.1.6	Unterschied: Kleinschreibung	351
10.1.7	Unterschied: Leere Elemente.....	352
10.1.8	Unterschied: Elemente mit optionalem Abschluss-Tag.....	352
10.1.9	Unterschied: Attributwerte in Anführungszeichen	353
10.1.10	Unterschied: Alleinstehende (leere) Attribute.....	353
10.1.11	Unterschied: Leerraum in Attribut-Wertzweisungen	353
10.1.12	Unterschied: Verweise zu Ankern	354
10.1.13	Unterschied: Das lang-Attribut	354
10.1.14	Unterschied: Inhalt von Script- und Style-Bereichen.....	355
10.1.15	Unterschied: Ausnahmen von Verschachtelungsregeln	355
10.2	Backgrounds: XML-Grundlagen	356
10.2.1	Semantische Auszeichnung beliebiger Daten.....	356
10.2.2	XML, DTDs und Validierung	357
10.2.3	Baumstruktur und Knoten eines XML-Dokuments.....	359
10.2.4	XSL oder CSS – Formatierung für XML-Elemente	360
10.2.5	Transformation mit XSLT	361
10.2.6	XML-Namensräume	361
11	Veraltetes HTML.....	363
11.1	Obsoletes HTML.....	363
11.1.1	Obsolete Elemente.....	363
11.1.2	Obsolete Attribute.....	365
11.2	Proprietäres HTML.....	370
11.2.1	Proprietäre Elemente.....	371
11.2.2	Proprietäre Attribute	372
12	CSS (Cascading Stylesheets)	375
12.1	CSS in HTML einbinden	375

12.1.1	Style-Definitionen in separaten CSS-Dateien	375
12.1.2	Kodierung von externen Stylesheets mittels @charset	377
12.1.3	Style-Bereiche im HTML-Dokumentkopf	377
12.1.4	Inline-Styles in HTML-Elementen	378
12.2	Grundsätzliche Grammatik von CSS	380
12.2.1	CSS-Regeln (Rules).....	380
12.2.2	@import: CSS-Dateien einbinden.....	380
12.2.3	@charset: Zeichenkodierung für externe Stylesheets	381
12.2.4	@media: Formate für bestimmte Ausgabegeräte (Media Queries)	382
12.2.5	@page: Formate für Print-Layouts	384
12.2.6	@font-face: Formatbereich für Web-Schriftarten.....	385
12.2.7	Selektoren	386
12.2.8	Formate für HTML-Elemente	386
12.2.9	Formate für verschachtelte HTML-Elemente	387
12.2.10	Formate für Elemente mit bestimmten Attributen	389
12.2.11	Formate für Elemente mit class-Attribut.....	390
12.2.12	Formate für Elemente mit id-Attribut.....	391
12.2.13	Formate für »Pseudo-Elemente und Pseudo-Klassen«	392
12.2.14	Formate	393
12.2.15	Kaskadierung und Vererbung	394
12.2.16	Spezifität – Gewichtung der Selektoren.....	396
12.3	Das Boxmodell von CSS	398
12.3.1	Das Verhalten von Block-Elementen.....	398
12.3.2	Das Verhalten von Inline-Elementen.....	399
12.3.3	Container für Elemente.....	400
12.4	Wertangaben in CSS	403
12.4.1	Maßeinheiten für numerische Werte	404
12.4.2	Farbwerte und Farbnamen.....	405
12.4.3	Farbnamen.....	409
12.4.4	Netscape-Farbpalette.....	410
12.5	CSS3.....	412
12.5.1	CSS3-Profile.....	423
13	CSS-Eigenschaften	425
13.1	Schriftformatierung.....	425
13.1.1	Schriftart.....	425
13.1.2	Schriftvariante, Schriftgewicht, Schriftstil und Schriftgröße.....	427
13.1.3	Die zusammenfassende font-Eigenschaft.....	429
13.1.4	Zeichenabstand, Wortabstand und Schriftlaufweite	430
13.1.5	Schriftfarbe und Schrift mit Schatteneffekt.....	432
13.1.6	Schriftattribute und automatische Text-Transformationen	434
13.2	Schriftformatierung mit Schriftartendatei	435
13.3	Ausrichtung und Absatzkontrolle	438

13.3.1	Horizontale und vertikale Ausrichtung.....	438
13.3.2	Texteinrückung	440
13.3.3	Zeilenhöhe.....	441
13.3.4	Textumbruchkontrolle	442
13.4	Außenrand und Abstand	444
13.5	Innenabstand	446
13.6	Rahmen, Ecken und Konturen	449
13.6.1	Schmuckrahmen (border)	449
13.6.2	Abgerundete Ecken (border-radius).....	453
13.6.3	Grafische Rahmen (border-image).....	456
13.6.4	Funktionsrahmen (outline).....	459
13.7	Hintergrundfarben und Hintergrundbilder	461
13.7.1	Hintergrundfarben	461
13.7.2	Hintergrundgrafiken.....	463
13.7.3	Zusätzliche Eigenschaften für die Hintergrundgestaltung in CSS3.....	467
13.8	Listenformatierung.....	468
13.8.1	Gestaltungsmöglichkeiten für ungeordnete Listen	468
13.8.2	Gestaltungsmöglichkeiten für geordnete Listen	470
13.8.3	Abstand zu Aufzählungszeichen/Nummerierung.....	473
13.9	Tabellenformatierung.....	474
13.9.1	Typische CSS-Formatierungen für Tabellen.....	474
13.9.2	Rahmenoptionen	477
13.9.3	Anzeige leerer Tabellenzellen.....	480
13.9.4	Ausrichtung der Tabellenbeschriftung.....	481
13.10	Breite und Höhe von Elementen	483
13.11	Positionierte Elemente	485
13.12	Floatende Elemente	489
13.13	Anzeigesteuerung von Elementen	494
13.13.1	Sichtbarkeit von Inhalten	494
13.13.2	Content-Modelle von Elementen umdefinieren.....	497
13.14	Layer-Technik und Transparenz	500
13.14.1	Schichtposition von Elementen.....	500
13.14.2	Transparenz von Elementen	502
13.15	Pseudo-Klassen für Hyperlinks.....	504
13.16	Pseudoklassen und -elemente für Textabsätze.....	507
13.17	Automatische Inhalte und Nummerierung.....	508
13.17.1	Automatische Inhalte vor und nach einem Element.....	508
13.17.2	Automatische Anführungszeichen.....	512
13.17.3	Automatische Nummerierung.....	514
13.18	Cursor-Anzeige.....	516
13.19	Media-Queries für mobile Devices.....	518
13.19.1	Media-Types in CSS2	518
13.19.2	Media-Queries in CSS3	519

14	HTML und Scripting	525
14.1	Event-Handler	525
14.1.1	Event-Handler für Mausereignisse.....	525
14.1.2	Event-Handler für Tastaturereignisse.....	529
14.1.3	Interaktive Elementereignisse	531
14.1.4	Event-Handler für Fenster- und Dokumentereignisse	534
14.2	Script- und Noscript-Bereiche	537
14.2.1	Externe Scripts einbinden	539
14.3	ECMA/JavaScript-Grundlagen.....	541
14.3.1	Sprachbestandteile von ECMAScript/JavaScript	541
14.3.2	DOM-Scripting – Zugriff auf Webseiteninhalte.....	553
14.3.3	Markupstruktur und Knoten	554
14.3.4	Auf Elemente des HTML-Dokuments zugreifen.....	555
14.3.5	HTML-Elementobjekte und HTML-Attribute als Objekteigenschaften.....	558
14.3.6	CSS-Eigenschaften von HTML-Elementen als Objekteigenschaften.....	560
14.3.7	Elemente, Attribute und Text im DOM erzeugen.....	563
14.4	JavaScript: Backgrounds	565
14.4.1	JavaScript-Objekte	565
14.4.2	Das document-Objekt	567
14.4.3	Die verschiedenen DOM-Modelle	567
14.5	JavaScript: Unobtrusive Programming	569
14.5.1	Elemente mit speziellen Eigenschaften ermitteln	569
14.5.2	Event-Handler im Skript anmelden	570
15	Scripting mit jQuery.....	573
15.1	Der Siegeszug eines Frameworks	573
15.1.1	JavaScript-Frameworks.....	573
15.1.2	jQuery-Plug-ins	574
15.2	jQuery einbinden	574
15.2.1	jQuery downloaden und lokal einbinden.....	575
15.2.2	jQuery über Hochverfügbarkeitsserver einbinden.....	575
15.3	Grundsätzliche Funktionsweise von jQuery	576
15.3.1	Das jQuery- bzw. \$-Objekt.....	576
15.3.2	\$(document) und \$(document).ready()	577
15.3.3	Verkettung von Objekten und Objektfunktionen.....	579
15.3.4	CSS-kompatible Selektoren	580
15.3.5	Aus Literalen und Variablen zusammengesetzte Selektoren.....	581
15.3.6	Traversion und this-Selektor	582
15.4	DOM-Manipulationen.....	583
15.4.1	Elementinhalte ermitteln und ändern	583
15.4.2	Attribute ermitteln, ändern und entfernen	584
15.4.3	Optik von Elementen auf CSS-Ebene ermitteln und ändern.....	584

15.4.4	Elemente ein- und ausblenden.....	586
15.4.5	Elemente hinein- und herausblenden.....	588
15.4.6	HTML-Inhalte einfügen und entfernen	589
15.4.7	DOM-Strukturen errichten	590
15.5	Event-Handling	593
15.5.1	Handler-Funktionen	593
15.5.2	Zweck der Funktionen on() und bind()	595
15.5.3	Event-Informationen	596
15.6	Formularmanagement.....	598
15.6.1	Feldwerte lesen und ändern	598
15.6.2	Ajax-unterstützende Funktionen.....	600
15.7	Ajax mit jQuery.....	601
15.7.1	GET-Requests.....	602
15.7.2	POST-Requests	604
15.8	Plug-ins.....	607
A	HTML-Elementreferenz.....	609
A.1	Liste der HTML5-Elemente.....	609
B	HTML-Attributreferenz	673
B.1	Liste der HTML5-Attribute	673
C	HTML-Eventreferenz.....	703
C.1	Allgemeines zu Eventhandlern und Events	703
C.2	Auflistung der HTML5-Events und -Eventhandler	703
D	CSS-Referenz.....	715
E	HTML-Zeichenreferenz	755
E.1	Benannte Zeichen aus dem ASCII-Zeichensatz	755
E.2	Benannte Zeichen aus Latin-1	757
E.3	Benannte Zeichen aus Latin Extended A/B.....	763
E.3.1	Latin Extended A	763
E.3.2	Latin Extended B	769
E.4	Benannte Zeichen für diakritische Zeichen.....	770
E.5	Benannte Zeichen für griechische Buchstaben.....	771
E.6	Benannte Zeichen für kyrillische Buchstaben.....	775
E.7	Benannte Zeichen für Interpunktion	779
E.8	Benannte Zeichen für Zahlen	781
E.9	Benannte Zeichen für Pfeilsymbole	782
E.10	Benannte Zeichen für mathematische Symbole.....	783
E.11	Benannte Zeichen für technische Symbole.....	785
E.12	ASCII-Zeichensatz	786
F	MIME-Typenreferenz.....	789
F.1	Übersicht der MIME-Typen	790

G	Referenz: Sprachenkürzel	795
	Stichwortverzeichnis	799

1 Intro

- Was Sie von diesem Buch erwarten können
- Wie HTML mit dem Gesamtkomplex »Erstellung von Websites« zusammenhängt

1.1 Zum vorliegenden Buch

Das vorliegende Buch vermittelt Ihnen HTML, die Basissprache zum Erstellen von Websites. Der Schwerpunkt liegt dabei auf Anwendern, die bereits früher einmal HTML gelernt haben, aber die neuen Möglichkeiten von HTML5 noch nicht kennen. Selbstverständlich ist das Buch aber auch für Neueinsteiger geeignet.

Dieses Buch behandelt in erster Linie HTML. Es ist daher nicht geeignet, Webdesign zu lernen, denn zum Erstellen von Webseiten-Layouts benötigen Sie heute Cascading Style Sheets (CSS). HTML hat ausschließlich die Aufgabe, Inhalte von Webseiten zu strukturieren. In manchen Beispielen des Buches kommt etwas CSS-Code vor. Dieser ist jedoch nicht der zentrale Gegenstand der Erläuterungen.

Auch Web-Programmiersprachen wie JavaScript/ECMAScript, PHP oder Java werden in diesem Buch nicht behandelt. Nur am Rande kommen einige Programmiersprachenbeispiele vor, da einige HTML5-Elemente nur im Zusammenhang mit Programmierung erklärbar sind.

1.1.1 Zielgruppen und Ziele

Das HTML-Handbuch richtet sich an folgende Zielgruppen:

- Anwender, die eine eigene Homepage oder einen eigenen Webauftritt von Grund auf und »in Handarbeit« erstellen wollen.
- **Webautoren**, die ein zeitgemäßes HTML-Arbeitsbuch für ihre alltägliche Arbeit benötigen.
- **Entwickler und Programmierer**, die Anwendungen im Web-Umfeld erstellen und HTML für dynamische Browser-Ausgaben oder zur Template-Entwicklung benötigen.
- **Blogger, Verkäufer** und andere, die HTML auf webbasierter Ebene benötigen, um Inhalte zu strukturieren.

1.1.2 Aufbau und Vorgehensweise

Das vorliegende Buch ist keine reine Faktenreferenz, sondern legt viel Wert darauf, auch Hintergründe und Zusammenhänge zwischen den beschriebenen Inhalten zu vermitteln. Das Buch folgt jedoch bewusst nicht der aus der amerikanischen Medienwelt

stammenden Tutorial-Philosophie. Stattdessen kommt es eher solchen Lesern entgegen, die eine am Inhalt orientierte Systematik bevorzugen.

Das Buch lässt sich durchaus von vorne bis hinten durchlesen. Doch es ermuntert seine Leser an vielen Stellen auch, Teile erst einmal zu überspringen und an anderen Stellen (weiter vorne oder weiter hinten) weiterzulesen. Denn viele Leser möchten die Reihenfolge, in der sie sich Wissen aneignen, nicht vorgeschrieben bekommen. Eigentlich ist der Autor dieses Buches ja auch ein bekannter Verfechter von Hypertext. Und Hypertext bedeutet: Der Anwender bahnt sich seinen Weg selbst, wobei es durchaus erlaubt ist, ihm vorgegebene Wege anzubieten.

1.1.3 Verwendete Symbole

Mit den folgenden Symbolen werden Befehle, Elemente oder Attribute gekennzeichnet, welche ab der angeführten Version zur Verfügung stehen oder eingeführt wurden.

HTML

Version	2.0	3.2	4.0	5
Symbol				

XHTML

Version	1.0	1.1	5
Symbol			

CSS

Version	1.0	2.0	2.1	3.0
Symbol				

Browser

Mit den folgenden Symbolen werden Befehle, Elemente oder Attribute gekennzeichnet, die von bestimmten Browsern unterstützt werden. Zusätzlich wird die Versionsnummer angeführt, ab welcher die Unterstützung vorhanden ist.

Es werden aber nur die Versionen dargestellt, die noch häufig bei den Benutzern anzutreffen sind.

MS Internet Explorer

Version	alle	7.0	
Symbol			usw. für weitere Versionen

Mozilla Firefox

Version alle 3.0

Symbol   usw. für weitere Versionen**Google Chrome**

Version alle 8

Symbol   usw. für weitere Versionen**Apple Safari**

Version alle 3.0

Symbol   usw. für weitere Versionen**Opera**

Version alle 9.0

Symbol   usw. für weitere Versionen**Hinweise**

Diese Symbole verwenden wir für spezielle Hinweise, welche für Sie wichtig sind.

Bedeutung	Neu in HTML5	Neu in CSS3	Semantische Auszeichnung	Zusätzliche Informationen	Weiterführende Hyperlinks
-----------	-----------------	----------------	-----------------------------	------------------------------	------------------------------

Symbol					
--------	---	---	---	---	---

1.2 Website-Erstellung heute

15 bis 20 Jahre nach den Anfängen des World Wide Webs findet das Produzieren von Inhalten fürs Web auf höchst unterschiedliche Weise statt. Der eine bastelt einzelne Webseiten auf Code-Ebene im HTML-Editor, der nächste entwirft Weblayouts in Photoshop und setzt diese dann mit WYSIWYG-Werkzeugen (What You See Is What You Get) wie Dreamweaver um. Wieder andere editieren online ihre Blogs, erzeugen sogenannten »user generated content« in Foren, Wikis oder Social-Bookmark-Services oder bearbeiten ihre Auktionsseiten bei eBay.

Bei manchen dieser Publikationsformen sind überhaupt keine HTML-Kenntnisse erforderlich. Bei anderen Formen sind solche Kenntnisse zumindest von Vorteil und bei wieder anderen sind sie unabdingbar. Die nachfolgende Website-Typologie gibt einen Überblick über verbreitete Publikationsformen. Dabei wird auch beschrieben, ob und in welchem Umfang HTML-Kenntnisse erforderlich sind.

1.2.1 Website-Typologie

Innerhalb der modernen Website-Landschaft gibt es gewisse Standard-Typen von Websites. Wenn Sie planen, etwas im Web zu publizieren, sollten Sie diese Typen kennen und für sich entscheiden, welchen Website-Typ Sie anstreben.

Homepage



Eine Homepage ist eine private Website mit persönlichen Inhalten und häufig auch mit eigenen Lieblingslinks.

In den Anfangsjahren des Webs, also in den frühen 90er Jahren, hatte jemand, der Webseiten publizierte, einfach eine Homepage. Der Name war kein Zufall. Denn die entsprechenden Leute waren meist Angehörige von Universitäten, wo sie permanent mit dem Internet verbunden waren. Der Server-Rechner, auf dem sie ihre Homepages speicherten, befand sich im eigenen Universitätsnetz. Die Homepage war einerseits die persönliche Startseite zum Surfen, gleichzeitig aber offen zugänglich für alle anderen Web-Benutzer. Homepages boten inhaltlich eine Mischung aus persönlicher Bookmark-Sammlung und lockeren, mitunter humorvoll gestalteten Informationen über ihren Besitzer.

Noch heute pflegen viele Privatleute eine persönliche Homepage im Web, auf der sie ohne professionellen Anspruch persönliche Inhalte veröffentlichen und ihre Lieblingslinks pflegen.

HTML-Kenntnisse sind zum Erstellen einer Homepage in mehr oder weniger großem Umfang erforderlich. Denn Homepages werden meistens noch mit viel Liebe zum Detail »per Hand« erstellt und in Form statischer HTML-Dateien gespeichert.

Web-Auftritt und Web-Präsenz



Ein Web-Auftritt, auch als Web-Präsenz bezeichnet, ist eine repräsentative geschäftliche Website mit Inhalten wie Angebots-Portfolio, Referenzen und Kontaktmöglichkeiten.

Die meisten Unternehmen, Agenturen, Organisationen, Vereine, Verbände, Behörden, Anwälte, Ärzte und Notare, Restaurants, Discos, Bars und Kneipen, Handwerker und andere Selbständige bieten längst eigene Websites an. Bei vielen Anbietern beschränkt sich das Angebot auf einige wenige statische Webseiten. Das genügt, um die eigene Angebotspalette ausführlich vorzustellen, um Referenzen zu nennen, um Kontaktmöglichkeiten und Anfahrtsbeschreibungen, Öffnungszeiten usw. mitzuteilen.

Zu einem geschäftlichen, repräsentativen Webauftritt gehört auch ein eigener Domain-Name. Beim Hosting von Web-Auftritten muss auf eine hohe Verfügbarkeit geachtet werden. Namhafte Anbieter (z. B. größere Unternehmen, Behörden mit einer großen Klientel) oder Anbieter, die mit kurzfristig hohen Besucherzahlen rechnen müssen (z. B. Prominente mit Neigung zu Skandalen) müssen beim Hosting ihres Webauftritts darauf achten, dass die Webserver auch größere Besuchermengen verkraften.

HTML-Kenntnisse sind zum Entwickeln aussagekräftiger individueller Web-Auftritte in hohem Maß erforderlich. Denn professioneller HTML-Code sollte möglichst fehlerfrei, semantisch sinnvoll strukturiert und für Suchmaschinen optimiert sein.

Weblog (Blog)



Ein Weblog (Kurzform: Blog) ist eine primär chronologisch organisierte, dynamisch präsentierte Website, die aus vielen Einzelartikeln besteht.

Um Besucher dauerhaft an eine Website zu binden, sind regelmäßig neue, interessante Inhalte erforderlich. Ein Weblog ist eine Website, die auf der Startseite typischerweise immer die neuesten Beiträge (Artikel) präsentiert. Der neueste Beitrag steht ganz oben. Die Startseite ändert sich also bei jedem neu hinzukommenden Beitrag. Diese Form der Präsentation hat heute die klassischen Homepages weitgehend verdrängt. Empfehlenswert ist sie jedoch nur, wenn das Potenzial vorhanden ist, einigermaßen regelmäßig neue, eigene Inhalte in Form mehr oder weniger kurzer oder langer Artikel zu produzieren.

Auch viele Unternehmen leisten sich längst eigene Blogs, da auch sie auf diese Weise Besucher stärker an ihr Webangebot und damit letztlich an das Unternehmen selbst binden können. Im Unternehmensumfeld spricht man von Corporate Blogs. Der redaktionelle Aufwand für Inhalte, die Besucher wirklich interessieren, ist jedoch nicht zu unterschätzen und kann hohe Personalkosten verursachen.

HTML-Kenntnisse sind beim Betrieb von Weblogs von Vorteil. Weblogs werden normalerweise nicht »von Hand« mit einem HTML-Editor erstellt. Stattdessen wird eine web-basierte-Software (Webanwendung, die im Web-Browser läuft) benutzt, die das Speichern von Artikeln, die Aktualisierung der Startseiten usw. automatisiert und zahlreiche, blogtypische Zusatzfunktionen bereitstellt, wie etwa die sogenannte Ping-Funktion, um die Blogosphäre über neue Beiträge zu informieren. Eine bekannte Anwendung dieser Art ist WordPress. Dazu kommen Hosting-Farmen wie *www.blogger.com*, bei denen eine einfache Registrierung genügt, um ohne spezielle Hosting-Verträge sofort webbasiert ein eigenes Blog zu starten. Der Blogger kann sich dadurch auf die redaktionelle Arbeit konzentrieren. Wer sich auf der Code-Ebene auskennt, kann seine Inhalte jedoch auch bei solchen Services vielseitiger, semantisch sinnvoller und interessanter strukturieren.

Web-Shop



Ein Web-Shop ist eine Website, die primär einen Produktkatalog präsentiert und dem Besucher dabei eine Warenkorbfunktion mitsamt Bestellabwicklung ermöglicht.

Das Web ist bestens geeignet, Produkte online zu verkaufen. Auch die Akzeptanz von Online-Shopping in der breiten Bevölkerung wächst ständig. Das Wachstumspotenzial für Online-Umsätze ist immer noch riesig. Verkäufer können im Web überregional agieren, ohne Filialen zu gründen. Anbieter von sehr speziellen oder selten verlangten Produkten können auf teure Geschäftsanmietungen verzichten.

Webshops lassen sich ohne Hilfe von Webshop-Software nicht sinnvoll betreiben. Die Software ermöglicht dem Anbieter, seinen Produktkatalog zu pflegen, also Informatio-

nen, Bilder, Preise usw. zu einzelnen Produkten zu speichern. Die Datenpflege erfolgt meist webbasiert, also direkt im Browser, wo die Webshop-Software dem Anbieter ein zugangsgeschütztes Backend anbietet. Das Frontend der Shop-Software arbeitet im öffentlich zugänglichen Shop. Dieser Software-Teil verwaltet, welche Produkte der Anwender einkaufen möchte (Warenkorbfunktion), und übernimmt den sensiblen Bestellvorgang. Sensibel ist dieser Vorgang deshalb, weil der Anwender währenddessen persönliche Daten von der Lieferadresse bis hin zu Kreditkartennummern übertragen muss. Gesetzliche Vorgaben wie die Widerrufsbelehrung usw. müssen softwaretechnisch so umgesetzt werden, dass der Shop rechtlich nicht anfechtbar ist.

Webshop-Software kann je nach Lizenzierungsform sehr teuer werden und muss sich entsprechend rentieren. Um die geeignete Lösung zu finden, lohnt ein Software-Vergleich. Viele Hosting-Provider bieten Komplettlösungen an, bestehend aus Hosting und Shop-Software.

HTML-Kenntnisse sind beim Betrieb eines Webshops ähnlich wie beim Betrieb eines Blogs meist nicht zwingend erforderlich, aber dennoch von Vorteil. Mit entsprechenden Kenntnissen und der Möglichkeit, auf Code-Ebene zu arbeiten, lassen sich beispielsweise Produkte besser präsentieren.

Web-Plattform



Eine Web-Plattform ist eine Website mit speziellen Zielgruppen, speziellen Funktionen und häufig mit personalisierten Inhalten für registrierte Benutzer.

Im geschäftlichen Bereich werden Web-Plattformen beispielsweise eingerichtet, um webbasierten Kunden-Support zu leisten oder um registrierten Kunden Ergänzungen oder Online-Features zu erworbenen Produkten anzubieten. Auch Social-Networking-Plattformen, Wiki-Farmen oder Wikis und Online-Office-Angebote gehören zu den Web-Plattformen.

Web-Plattformen arbeiten häufig mit registrierten Benutzern, wobei die Möglichkeit der Selbstregistrierung häufig dazugehört. Denn registrierte Benutzer, die sich mit eigenen Zugangsdaten anmelden, sind die Grundvoraussetzung, um personalisierte Inhalte anzubieten.

Auch moderne Intranets werden eher als Web-Plattform konzipiert. Während Intranets früher nur statische, mitunter sehr verklemmte Präsentationen einzelner Abteilungen waren, stehen bei moderneren Lösungen das kollaborierende Arbeiten an gemeinsamem Know-How und der fachliche Austausch im Vordergrund.

HTML-Kenntnisse reichen zum Entwickeln einer Web-Plattform nicht aus. Hierzu sind Web-Entwickler mit viel Know-How in Web-Programmiersprachen gefragt. Eine bereits entwickelte Web-Plattform kann jedoch ganz verschiedene Arbeitsebenen haben. Je nach Art und technischer Realisierung der Plattform können dann HTML-Kenntnisse zum Erstellen redaktioneller Inhalte erforderlich oder von Vorteil sein.

1.3 Web-Technologien

In diesem Abschnitt werden wir die Sprachen und Technologien ansprechen, die typischerweise zum Erstellen von Webseiten eingesetzt werden.

1.3.1 HTML / XHTML

HTML bedeutet Hypertext Markup Language. Als Auszeichnungssprache (Markup Language) hat HTML die Aufgabe, die logische Grob- und Feinstruktur einer Webseite zu beschreiben. HTML geht aus der Familie der SGML-basierten Sprachen (SGML steht für Standard Generalized Markup Language) hervor.

Vom Web-Gründer Tim Berners-Lee entwickelt, wurde HTML im Zuge des Web-Booms zum erfolgreichsten und am weitesten verbreiteten Dateiformat der Welt. HTML ist eine Sprache zur Strukturierung von Texten, wobei aber auch die Möglichkeit besteht, Grafiken und multimediale Inhalte in Form einer Referenz einzubinden und in den Text zu integrieren.

Mit HTML können Sie Webseiten in Bereiche wie Kopf, Inhalt, Fuß, Navigation und Artikel unterteilen. Weiter können Sie Überschriften, Textabsätze, Listen und Tabellen erzeugen. Sie können anklickbare Verweise auf beliebige andere Webseiten oder Datenquellen im Internet erzeugen. Nicht-textuelle Inhalte können Sie referenzieren. Sie können Formulare in den Text integrieren. Und last but not least bietet HTML Schnittstellen für Erweiterungssprachen wie CSS oder JavaScript an, mit deren Hilfe Sie HTML-Elemente nach Wunsch gestalten oder Interaktionen mit dem Anwender realisieren können.

Das Auszeichnungsschema von HTML geht von einer hierarchischen Gliederung aus. HTML-Dokumente bestehen zunächst aus Kopf- und Inhaltsdaten. Der Inhalt besteht typischerweise aus allgemeinen Elementen für die Bereiche einer Webseite, also z. B. Kopfbereich, Inhaltsspalte, Navigation. Die Inhaltsspalte besteht etwa aus Überschriften 1. und 2. Ordnung, Textabsätzen, Listen, Tabellen und Grafiken. Einige dieser Elemente haben wiederum Unterelemente. So enthält ein Textabsatz zum Beispiel eine als betont oder fett markierte Textstelle, eine Aufzählungsliste besteht aus einzelnen Listenelementen, und eine Tabelle gliedert sich in einzelne Tabellenzellen. HTML ist so konzipiert, dass sich die gesamte Elementstruktur einer Webseite als **Baumstruktur** mit immer feineren Verzweigungen abbilden lässt.



HTML-Versionen

HTML 1.0 (ab 1990) war eine reine Internet-Draft und enthielt Auszeichnungen für Standardelemente wie Überschriften, Textabsätze, für Grafikreferenzen und Verweise.

<http://www.w3.org/MarkUp/draft-ietf-iiir-html-01.txt>

HTML 2.0 (ab November 1995) wurde offizieller Sprachstandard. HTML 2.0 stellt keine wesentliche Weiterentwicklung gegenüber HTML 1.0 dar.

<http://www.w3.org/MarkUp/html-spec/html-pubtext.html>

HTML 3.2 (ab Januar 1997) ging aus der vorangegangenen Entwicklung zu HTML 3.0 hervor. Letzteres war am Browser-Markt vorbei entwickelt worden. Aus heutiger Sicht stellt HTML 3.2 jedoch ein mittleres Unglück dar. Denn in dieser Version wurde HTML als Sprache für optische Effekte etabliert, was jedoch nicht dem Wesen der Sprache entsprach und auch nur halbherzig implementiert war. Viele Bestandteile von HTML 3.2 wurden später als *deprecated* (missbilligt) eingestuft und wieder aus dem Sprachstandard entfernt.

<http://www.w3.org/TR/REC-html32.html>

HTML 4.0 (ab Dezember 1997) wurde 1999 von der überarbeiteten und anschließend gültigen Version 4.01 abgelöst. HTML 4.01 stellte den Versuch dar, sich auf die Kernaufgaben von HTML zurückzubesinnen. Gleichzeitig wurden die Cascading Style Sheets (CSS) als HTML-Ergänzungssprache forciert. Um der Praxis entgegenzukommen, gibt es von HTML 4.01 drei Varianten, nämlich:

- HTML 4.01 strict: entsprach dem eigentlich gewünschten HTML.
- HTML 4.01 transitional: erlaubte etablierte physische Textauszeichnungen und war als Übergangslösung konzipiert.
- HTML 4.01 frameset: Spezialfall der Transitional-Variante für Webseiten, die Frames einsetzen.

http://www.w3.org/TR/HTML_401/

HTML5 (seit 2007 in Entwicklung) ist der neue HTML-Standard. In dieser Version wird die Sprache formal nicht mehr als SGML-Dokumenttyp definiert, sondern als Dokument-Objekt-Modell (DOM). Dadurch kommt die Sprache den Erfordernissen der Programmierung stärker entgegen. Es werden zahlreiche neue, an der Praxis orientierte Elemente eingeführt, die es erlauben, Webseiten semantisch ordentlicher zu strukturieren.

<http://www.w3.org/TR/html5/>

Eine der wichtigsten Eigenschaften von HTML ist die Möglichkeit, Verweise (Hyperlinks) zu definieren. Verweise (Links, Hyperlinks) können zu anderen Stellen der eigenen Website führen, aber auch zu beliebigen anderen Adressen im World Wide Web und sogar zu Internet-Adressen, die nicht Teil des Web sind.

HTML ist ein sogenanntes **Klartextformat**. HTML-Dokumente können Sie mit einem beliebigen Texteditor bearbeiten, der Daten als reine Textdateien abspeichern kann. HTML ist also nicht an irgendein bestimmtes, kommerzielles Softwareprodukt gebunden. Da es ein Klartextformat ist, lässt es sich auch hervorragend mit Hilfe von Programmen oder Scripts generieren. Wenn Sie im Web zum Beispiel einen Suchdienst benutzen und nach einer Suchanfrage die Ergebnisse präsentiert bekommen, dann ist das HTML-Code, der von einem Script generiert wurde. Die Verwendung von HTML ist nicht an Lizenzen gebunden.

XHTML

XHTML löst HTML aus der im Web-Bereich sonst unbedeutenden SGML-Sprachenwelt heraus und bindet die Sprache stattdessen in die modernere und praxisrelevantere

XML-Sprachenwelt ein. XML (Extensible Markup Language) ist wie SGML eine Art Regelwerk zum Definieren von Markup-Sprachen. XHTML kann gegenüber HTML von der XML-Infrastruktur profitieren. Dazu gehört etwa die Möglichkeit, Daten von einer XML-Sprache in eine andere zu übersetzen (mit XSLT) oder Inhalte anderer XML-Sprachen direkt einzubetten (beispielsweise SVG für Vektorgrafiken oder MathML für mathematische Formeln). Die Umsetzung dieser Möglichkeiten ist jedoch bis heute in den Browsern lausig bis gar nicht vorhanden. Wer nicht explizit eine Daten-Anbindung an die XML-Welt benötigt, braucht auch kein XHTML, sondern ist mit HTML besser beraten.



XHTML-Versionen

XHTML 1.0 (seit Januar 2000) ist eine Reformulierung von HTML 4.01 mit Hilfe von XML. Das bedeutet, XHTML enthält genau die gleichen Sprachbestandteile wie HTML 4.01. Die Unterschiede liegen auf anderen Ebenen und sind durch die Unterschiede von SGML und XML bedingt. XHTML 1.0 übernimmt von HTML 4.01 auch die drei Varianten strict, transitional und frameset.

<http://www.w3.org/TR/xhtml1/>

XHTML 1.1 (in Entwicklung seit 2001) ist eine XHTML-spezifische Weiterentwicklung. Die Sprachvarianten transitional und frameset entfallen. Die Version 1.1 soll die Basis schaffen für ein modulbasiertes XHTML. Als Module sind z. B. Struktur, Text, Verlinkung, Listen, Tabellen, Formulare usw. vorgesehen. In der heutigen Praxis der Website-Erstellung spielt XHTML 1.1 kaum eine Rolle.

<http://www.w3.org/TR/xhtml11/>

XHTML 2.0 (in Entwicklung seit 2002) verändert die Sprachbestandteile von XHTML deutlich. Es gibt neue Elemente, etwa für Navigationslisten, und es gibt logische Veränderungen (so kann in XHTML 2.0 jedes Element ein Hyperlink sein, weshalb das herkömmliche HTML-Element für Hyperlinks entfällt). In der heutigen Praxis der Website-Erstellung spielt auch XHTML 2.0 kaum eine Rolle.

<http://www.w3.org/TR/xhtml2/>

XHTML5 ist die Möglichkeit, HTML5 mit Hilfe von XML-gerechter Grammatik und Notation zu erzeugen. XHTML5 ist nur für das Parsen mit echten XML-Parsern gedacht. XHTML5, das von HTML-Parsern verarbeitet wird, gibt es nicht mehr. Es ist dann einfach HTML5, selbst wenn es die XHTML-typische Notation einhält. XHTML5 ist also keine eigenständige Sprache, sondern lediglich eine sogenannte Serialisierungsform des Dokument-Objektmodells von HTML5.

<http://www.w3.org/TR/xhtml5/>

1.3.2 CSS



CSS bedeutet Cascading Stylesheets. Es handelt sich um eine beschreibende Ergänzungssprache für HTML. Sie klinkt sich nahtlos in HTML ein und hat zwei Aufgaben: das Formatieren von Inhalten und das Gestalten von Webseitenlayouts.

Wenn Sie ein HTML-Dokument ohne CSS-Angaben im Browser anzeigen, werden grundsätzliche Strukturen visuell unterscheidbar dargestellt. So werden Überschriften in grafischen Browsern größer und fett angezeigt, Tabellen sind sichtbar, und wichtige Auszeichnungen im Text werden z. B. kursiv, fett oder in nicht-proportionaler Schrift dargestellt. Man spricht in diesem Zusammenhang von einem internen Default-Stylesheet, das Browser verwenden, um HTML-Elemente darzustellen. Die HTML-Spezifikation enthält Empfehlungen für solche Default-Darstellungen.

An diesem Punkt setzen die Cascading Stylesheets (CSS) ein. Mit Hilfe von Stylesheets können Sie beispielsweise festlegen, dass alle Überschriften 1. Ordnung 24 Pixel groß sind, in roter Helvetica-Schrift mit einem Nachabstand von 16 Pixel und einer grünen doppelten Rahmenlinie oberhalb dargestellt werden. Sie können aber genauso gut auch für einen beliebigen Text festlegen, dass nur dieser Text 3 Zentimeter groß sein soll und eine gelbe Hintergrundfarbe erhält.

CSS erlaubt es, zentrale Formate zu definieren, beispielsweise für alle Überschriften erster Ordnung oder für alle Textabsätze mit einem bestimmten Klassennamen oder für hervorgehobenen Text, der innerhalb einer Tabellenzelle vorkommt. Die zentralen Formate können in eine externe Style-Datei ausgelagert werden, die Sie dann in beliebig vielen HTML-Dokumenten einbinden können. So ermöglicht CSS seitenübergreifend einheitliche Layouts und Formatierungen.

CSS ist ebenso wie HTML eine Klartextsprache. Auch für CSS brauchen Sie keine bestimmte Software, es genügt ein Texteditor. CSS ist wie HTML eine offen dokumentierte und vom W3-Konsortium standardisierte Sprache, die Sie frei und ohne Lizenzprobleme verwenden können. Das Standardisierungsverfahren bei CSS folgt den gleichen Regeln wie das von HTML.



CSS-Versionen

CSS 1.0 (seit Dezember 1996) bietet vor allem Eigenschaften für die Standardformatierung (Schriftformatierung, Abstände, Rahmen, Farben usw.) an.

<http://www.w3.org/TR/REC-CSS1/>

CSS 2.0 (seit 1998) führte vor allem die Formateigenschaften ein, die CSS als Sprache zum Entwerfen von Webseitenlayouts befähigen. Außerdem wurde die Selektorensyntax zum Ansteuern von Elementen deutlich erweitert.

CSS 2.1 (seit 2007 Kandidat für den offiziellen Sprachstandard) ersetzt CSS 2.0, weshalb CSS 2.0 gar nicht mehr offiziell einsehbar ist. CSS 2.1 entfernt einige Bereiche, die sich in der Praxis als irrelevant erwiesen haben, aus dem CSS-Standard (z. B. Stylesheet-Angaben zum Steuern von Sprechstimmen) und präzisiert verschiedene Details von CSS 2.0.

<http://www.w3.org/TR/CSS21/>

CSS 3.0 (seit 1999 in Entwicklung) gliedert sich in über 20 Einzelspezifikationen, die jeweils eigene Review-Prozesse durchlaufen, weshalb die Einzeldokumente auch unterschiedliche Entwicklungsstände haben. CSS3 führt etliche Verbesserungen ein, beispielsweise bei der Verwendung von serverseitigen Schriftarten oder bei automatisch generierten Inhalten. Neuere Versionen heutiger Browser übernehmen bereits verschiedene Möglichkeiten von CSS3. Die Implementierungen sind jedoch noch sehr uneinheitlich.

<http://www.w3.org/Style/CSS/current-work>

Es dauerte eine Weile, bis die Entwicklergemeinde das volle Potenzial von CSS zu nutzen lernte. Das betraf vor allem die Möglichkeiten, mit CSS Elemente zu gestalten, die Behälter für ganze Teile einer Webseite sind. Mittels spezieller CSS-Formateigenschaften zum Anordnen von Elementen, zur Hintergrundgestaltung und zu sichtbaren Rahmen wurden moderne, CSS-basierte, mehrspaltige Website-Layouts möglich. Prinzipiell funktionieren solche Layouts in allen heute verwendeten Browsern. Der Teufel steckt jedoch im Detail, weil nicht alle Browser die CSS-Spezifikation identisch umsetzen, was sich gerade beim Seitenlayout unschön bemerkbar machen kann. So entstand eine ganze Fachliteratur rund um das Erstellen von CSS-basierten Webseitenlayouts.

1.3.3 Clientseitiges Scripting

1996 führte Netscape, der damalige Marktführer unter den Browsern, eine Scriptsprache namens JavaScript ein. Netscape ließ diese Sprache als ECMA-Industriestandard mit der Nummer 262 definieren.

Mit Hilfe von JavaScript konnten Web-Autoren das Verhalten von Webseiten und Browsern bis zu einem gewissen Grad beeinflussen. In HTML können Sie beispielsweise Formulare mit Eingabefeldern, Auswahllisten, Buttons usw. definieren. Der Anwender kann ein Formular ausfüllen und über das Web absenden. Doch HTML erlaubt Ihnen als Formularanbieter nicht, die Eingaben des Anwenders vor dem Absenden des Formulars auf Vollständigkeit und Plausibilität zu prüfen. Mit JavaScript ist das hingegen möglich.

Leider wurden die Möglichkeiten von JavaScript in den Anfangsjahren für allerlei Spielereien missbraucht, die dafür sorgten, dass sich JavaScript einen schlechten Ruf einhandelte. Daran änderte auch Microsoft nichts, das den ECMA-262-Sprachstandard unter dem Namen JScript implementierte und durch eigene Ergänzungen ausbaute.

Später gewann JavaScript seine Reputation durch zwei wichtige Entwicklungen zurück:

- Das W3-Konsortium schuf mit dem **Document Object Model (DOM)** ein programmiersprachenunabhängiges Modell für den Programmiersprachenzugriff auf XML-basierte Markupssprachen sowie auf HTML. Browser-Anbieter begannen, das DOM des W3C in ihre Sprachimplementierungen zu übernehmen. Dadurch entstanden viele nützliche, professionellere Anwendungen von JavaScript.
- Ein unscheinbares, spezielles JavaScript-Objekt namens **XMLHttpRequest** (bekannt durch die Verwendung mit **AJAX**, AJAX = Asynchronous JavaScript and XML), das es der Scriptsprache ermöglicht, während eine Webseite im Browser angezeigt wird,

im Hintergrund mit dem Webserver zu kommunizieren, hat das Erscheinungsbild des modernen Web nachhaltig mitgeprägt. Es ermöglicht Webanwendungen, die so flüssig auf Anwender-Interaktionen reagieren wie normale Programme auf dem lokalen Computer.

Wegen zahlreicher Detailunterschiede bei der Implementierung in den einzelnen Browsern halten sich heute viele Entwickler an sogenannte **JavaScript-Frameworks**. Das sind Code-Bibliotheken, die für alle wichtigen und häufig verlangten Aufgaben fertige Funktionen bereitstellen und den Entwickler von Browser-Unterschieden fernhalten.

1.3.4 Serverseitiges Scripting und Datenbankanbindung

Hinter der Mehrzahl der heute im Web aufrufbaren Seiten steckt keine statische, auf dem Webserver gespeicherte HTML-Datei mehr. Stattdessen werden die Seiten beim Aufruf dynamisch generiert. Nicht immer ist das an der URL-Adresse sofort erkennbar. Fast alle Blogs etwa werden mit Blog-Software wie WordPress erzeugt, und die meisten größeren Sites mit Content-Management-Systemen (CMS) wie TYPO3, Drupal oder Joomla!. Das sind Webanwendungen, die auf einem Webserver installiert werden. In der Regel bestehen sie aus einem Backend und einem Frontend.

Das **Backend** ist ein Administrationsbereich mit einer eigenen, öffentlich nicht bekannten Webadresse. Dort eingerichtete Benutzer können sich anmelden und das Backend nutzen. Das Backend ist selbst eine Webanwendung, die vorwiegend aus Formularen besteht. Es ermöglicht das Verwalten und Editieren von Webseiten, aber auch von verwaltungsberechtigten Benutzern.

Das **Frontend** ist eine Webanwendung, die das dynamische Generieren von Webseiten steuert. In der Regel besteht die Aufgabe darin, definierte HTML-Templates zu verwenden und in darin definierten Bereichen Inhalte einzufügen, die aus einer Datenbank oder aus Datendateien geholt werden. Dazu enthalten die HTML-Templates in der Regel Platzhalterzeichenfolgen, die von der Anwendung erkannt und durch generierte Inhalte ersetzt werden.

Backend- und Frontend-Anwendungen müssen programmiert werden. Dazu sind durchaus klassische Programmiersprachen wie C oder Delphi geeignet. Für webtypische Aufgaben besser gerüstet sind jedoch Sprachen wie PHP, Java, Python oder Perl. Die meisten bekannten Blogsysteme und Content-Management-Systeme sind heute in PHP, Java oder Python programmiert. Dabei sind PHP und Python reine Scriptsprachen. Die Anwendungen werden erst zur Laufzeit in Maschinensprache übersetzt. Dank immer schnellerer Rechner und sehr effizienter Script-Interpreter sind Scripts heute im Vergleich zu vorab kompiliertem Code absolut konkurrenzfähig.

Auch wenn Web-Autoren keine eigenen Webanwendungen erstellen, können Kenntnisse in einer webtypischen serverseitigen Scriptsprache wie PHP von Vorteil sein. Viele Aufgaben, egal ob Formular-Mailer, Umfrage, Bildergalerie oder das Ermitteln von Daten aus einer Datenbank, lassen sich damit lösen, auch wenn sonst keine große Webanwendung zum Einsatz kommt.

1.3.5 Flash

Flash bietet die Möglichkeit, Multimedia-Effekte auf Web-Seiten zu bringen, aber auch Anwendungen wie Spiele, Simulationen oder Produkt-Animationen. Im Laufe der Jahre hat es sich zum defacto-Standard für Multimedia im Web entwickelt, und das, obwohl es eine proprietäre Technologie mit einem binären Dateiformat ist, deren Original-Autorensoftware nicht ganz preiswert ist. Zwei wichtige Gründe haben den Erfolg von Flash begünstigt:

- Offene Standard-Alternativen wie SVG (SVG = Scalable Vector Graphics) in Verbindung mit SMIL (SMIL = Synchronized Multimedia Integration Language) und JavaScript sind letztlich zu komplex konzipiert, gehen an der Praxis von Multimedia-Designern vorbei, und bis heute lässt die Browser-Unterstützung dafür zu wünschen übrig.
- Das Flash-Plugin wurde irgendwann mit jedem Browser von Haus aus mit ausgeliefert, sodass Flash bedenkenlos eingesetzt werden kann und fast überall funktioniert.

Ohne Flash wären erfolgreiche Multimedia-Sites wie etwa YouTube gar nicht denkbar. Insgesamt hängt viel davon ab, wie Flash eingesetzt wird. Als Basis für einfach integrierbare Video- oder Music-Player ist es allgemein akzeptiert und beliebt. Nicht durchgesetzt haben sich dagegen reine flashbasierte Websites, also Websites, bei denen der Webserver nicht HTML-Dokumente mit eingebetteten Flashmovies sendet, sondern gleich nur noch Flashmovies.

HTML5 enthält jedoch einige Ansätze, die genau in den Einsatzbereich von Flash zielen. Dazu gehören etwa die neuen Elemente `video` und `audio` oder das Script-gesteuerte Element `canvas`. Diese Elemente könnten dazu beitragen, den Einfluss von Flash in Zukunft zu verringern. Besonders die nicht vorhandene Unterstützung von Flash bei den iDevices des Unternehmens Apple macht HTML5 hier zu einer echten Alternative.

Flash wurde ursprünglich von der Firma Macromedia entwickelt. Heute wird es von Adobe vertrieben.

1.3.6 XML



XML bedeutet Extensible Markup Language. Es handelt sich um ein Regelwerk für Markup Sprachen.

HTML ist eine Auszeichnungssprache zur Strukturierung gewöhnlicher Webseiten. Es gibt jedoch bereits im engeren Zusammenhang mit Webseiten Daten, die sich gar nicht oder nicht sinnvoll mit Hilfe von HTML beschreiben lassen: Vektorgrafiken zum Beispiel oder in Feedreadern abonmierbare Feeds von News- oder Blog-Artikeln. Dabei ist es jedoch von Vorteil, wenn auch andere beschreibende Sprachen den gleichen syntaktischen Grundregeln wie etwa HTML folgen. Ein Vorteil ist beispielsweise, dass ein Parser, also eine Software, die Markup-Sprachen verarbeitet, immer die gleichen Algorithmen zur Auflösung der notierten Auszeichnungen verwenden kann.

Nun wurde HTML bis einschließlich Version 4.01 mit Hilfe von SGML definiert. SGML ist ein allgemeines Regelwerk zur Definition von Markup-Sprachen wie HTML. Aller-

dings ist SGML sehr komplex und für die meisten Anwendungsfälle überdimensioniert. Deshalb hatte sich das W3-Konsortium bereits in den 90er Jahren entschlossen, ein leichter zu handhabendes und für die praktische Datenverarbeitung zweckmäßigeres Regelwerk zu entwickeln. Das Ergebnis ist XML. XML ist jedoch nichts, was von den Grundregeln von SGML völlig abweicht. Es ist vielmehr selbst ein sogenanntes »SGML-Profil«. Aber ebenso wie SGML ist XML selbst keine Anwendungssprache, sondern eben ein Regelwerk zur Definition von Anwendungssprachen.

Mit Hilfe von XML können Sie also eigene, neue Markup-Sprachen entwickeln, die sich an bestimmte, definierte Muster und Regeln halten. XML ist dabei das »Regelwerk«, das Schema. Beim Definieren eigener Markup-Sprachen mit XML können Sie eigene Namen vergeben. So können Sie zum Beispiel Elemente definieren, um Bereiche eines Dokuments als Lexikonartikel auszuzeichnen, als Musiknote oder als Bestandteil einer chemischen Formel. Sie müssen bei der Definition der Elemente jedoch exakt festlegen, welche Eigenschaften das Element hat, zum Beispiel, innerhalb welcher anderen Elemente es vorkommen kann und innerhalb welcher nicht. Wenn Sie dann ein XML-Dokument mit der Definition einer eigenen Sprache erstellt haben, können Sie Dateien in dieser Sprache erstellen und im Dateikopf angeben, auf welche XML-Definitionen Sie sich in dieser Datei beziehen.

Auszeichnungssprachen, die Sie mit XML definieren, sind zunächst noch nichts, was irgendwelche Dinge auf den Bildschirm zaubert. Dazu ist Software erforderlich, die eine Markup-Sprache parst und aus der daraus gewonnenen Datenstruktur irgendetwas macht, beispielsweise eine Bildschirmausgabe.

XML bietet einerseits die Möglichkeit, eigene Datenformate zu erzeugen, die einem standardisierten Schema folgen und daher Kompatibilität garantieren. Andererseits lassen sich mit XML aber auch Datenformate erzeugen, die für eine breitere Verwendung im Software-Markt geeignet sind. Das W3-Konsortium arbeitet an mehreren XML-basierten Auszeichnungssprachen. Für jede dieser Auszeichnungssprachen gibt es eine eigene Arbeitsgruppe, die sich um eine saubere Spezifikation und eine praxisgerechte Weiterentwicklung der jeweiligen Sprache kümmert. Aber auch andere Hersteller und Anbieter versuchen, XML-basierten Sprachen zu einem Standard-Status zu verhelfen. Die folgenden Sprachen sind Beispiele solcher Standardsprachen.



Bekannte XML-Sprachen

XHTML (Extensible Hypertext Markup Language)

XML-basierte Variante von HTML (siehe weiter oben).

SVG (Scalable Vector Graphics)

Auszeichnungssprache zur Beschreibung zweidimensionaler Vektorgrafik. Damit lassen sich grafische Daten beschreiben, wie sie von Programmen wie Corel Draw erzeugbar sind.

<http://www.w3.org/Graphics/SVG/>

MathML (Mathematical Markup Language)

MathML hat die Aufgabe, technisch-wissenschaftliche Inhalte, die mathematische Formelsprache zur Darstellung benötigen, abzubilden.

<http://www.w3.org/Math/>

RDF (Resource Description Framework)

Sprache zur Beschreibung von Metadaten für Dokumente. Metadaten können verschiedene Aufgaben haben: etwa als Input für Suchmaschinen dienen oder Rahmendaten für die Aufnahme des Dokuments in das Katalogsystem einer bestimmten elektronischen Bibliothek liefern.

<http://www.w3.org/RDF/>

RSS (Really Simple Syndication)

Sprache zur Beschreibung von Feeds für Blog-Artikel oder Newsticker.

<http://www.rssboard.org/rss-specification>

Atom

Sprache zur Beschreibung von Feeds für Blog-Artikel oder Newsticker.

<http://www.atompub.org/>

XSLT (Extensible Stylesheet Language, Transformationskomponente)

Ermöglicht das Übertragen von einer XML-basierten Markup-Sprache in eine andere. Dabei lässt sich exakt definieren, welche Elementinhalte und Attributwerte der Ausgangssprache in welche Strukturen der Zielsprache übertragen werden sollen.

<http://www.w3.org/TR/xslt>

XPath (XML Path)

Ermöglicht das gezielte Adressieren von Daten in einer XML-basierten Markup-Sprache. Wird beispielsweise von XSLT benötigt.

<http://www.w3.org/TR/xpath>

1.4 Software zum Erstellen von Websites

Das Wichtigste, was es zum Thema Software in Bezug auf das Entwickeln von Webseiten zu sagen gibt, ist, dass es **die Software** nicht gibt und hoffentlich auch nie geben wird. Internet-typische Technologien sind bewusst Software-unabhängig ausgelegt. Das hat gute Gründe. Im Internet treffen alle nur erdenklichen Betriebssysteme und Rechner-typen aufeinander, und Technologien, die für alle verfügbar sein sollen, haben keine großen Chancen, wenn sie systemspezifisch sind oder spezielle Anforderungen stellen, die Teile der Internet-Nutzer ausschließen.

1.4.1 Code-Editoren

Ohne einen guten Code-Editor kommt praktisch kein Webentwickler aus. Ein Universaleditor mit speziellen Funktionen für die Webentwicklung unterstützt gleichermaßen beim Editieren von HTML, CSS, XML, JavaScript oder serverseitigen Sprachen wie PHP. Zu den Eigenschaften, über die ein leistungsstarker Code-Editor verfügen sollte, gehören:

- **Syntax-Highlighting:** Schlüsselwörter, Klammern, Sonderzeichen, Element- oder Funktionsnamen in Sprachen wie HTML und XML, CSS, JavaScript, PHP, Java, Perl, Python, C, SQL sollte ein Editor farblich hervorheben. Dadurch werden Inhalte wesentlich leichter erfassbar.
- **Zeilenkontrolle:** Es sollte je nach Bedarf umschaltbar sein, ob lange Zeilen für die Ansicht umbrochen werden oder nicht. Ebenso sollten Zeilennummern angezeigt werden können. Denn viele Fehlermeldungen von Validatoren oder Programmiersprachen-Interpretern beziehen sich auf Zeilennummern.
- **Code-Folding:** Dieses Feature erlaubt es, strukturierte Codes (z. B. größere Bereiche eines HTML-Dokuments oder Funktionen eines Scripts) nach Bedarf ein- und auszublenken. Bei umfangreicheren Code-Dateien erleichtert das die Übersicht.
- **Code-Editier-Hilfen:** Gute Code-Editoren kennen beispielsweise alle HTML-Element- und -Attributnamen, CSS-Eigenschaften oder PHP-Funktionen. Während des Bearbeitens bieten sie entsprechende Autovervollständigung an. Außerdem existiert eine kontextsensitive Hilfe für Parameter, welche von einer Funktion erwartet werden.
- **Unterstützende Präzisionsfunktionen:** Das betrifft etwa eine ausgereifte Suchen-Ersetzen-Funktion, die auch viele Dateien gleichzeitig durchläuft und beispielsweise die Möglichkeit bietet, reguläre Ausdrücke zu verwenden. Auch sprachentypische Konvertierwerkzeuge wie das Umwandeln von höherwertigen Zeichen in ihre HTML-Entity-Namen oder in numerische HTML-Entities gehören dazu.
- **Flexible Erweiterbarkeit:** Viele Code-Editoren speichern ihre Informationen für Syntax-Highlighting und Code-Editier-Hilfen in eigenen Dateien. Diese Dateien haben ein dokumentiertes Format. Dadurch ist es möglich, Syntax-Highlighting-Anweisungen auch für Sprachen zu erstellen, die der Editor bislang nicht unterstützt.
- **Integration in andere Software-Umgebungen:** Ein guter Code-Editor ermöglicht es, individuelle Hilfedokumente zu integrieren. Für Entwickler ist es sinnvoll, wenn der Editor direkt mit einem Versionenkontrollsystem wie Subversion zusammenarbeitet, oder wenn er es ermöglicht, Dateien direkt via FTP/SFTP von einem Webserver-Rechner zu laden und wieder dort zu speichern.
- **Zeichenkodierung:** Gute Code-Editoren unterstützen alle verbreiteten Formen von Zeichenkodierung, also z. B. ANSI, ISO-8859-x, UTF-8, UTF-16, Unicode-2-Byte, Unicode-4-Byte, usw. Dazu gehört auch die Fähigkeit, Inhalte von einer Zeichenkodierung in eine andere zu konvertieren.
- **Performance:** Code-Editoren sind Programme, von denen erwartet wird, dass sie die Ressourcen eines Rechners nicht nennenswert belasten, dass sie sehr schnell laden

und sehr robust laufen. Sie sind selbstverständliche Arbeitswerkzeuge, die fast ständig geöffnet sind, auch während anderer Arbeiten.

Universaleditoren für MS Windows (Auswahl)

- **1stPage** (Shareware)
<http://www.evrsoft.com/>
- **40tudeHTML** (Shareware)
<http://www.40tude.com/>
- **Boxer** (Shareware)
<http://www.boxersoftware.com/>
- **CoffeeCup** (Shareware)
<http://www.coffeecup.com/>
- **EditPlus** (Shareware)
<http://www.editplus.com/>
- **EmEditor** (Shareware)
<http://www.emeditor.com/>
- **HTML-Kit** (Shareware)
<http://htmlkit.com/>
- **Homesite** (Trial-Ware)
<http://www.adobe.com/products/homesite/>
- **NoteTab** (Shareware)
<http://www.notetab.com/>
- **Notepad++** (OpenSource – GNU GPL)
<http://notepad-plus.sourceforge.net/>
- **SciTE** (OpenSource – Eigenlizenz)
<http://www.scintilla.org/SciTE.html>
- **Scribe!** (Freeware)
<http://www.scribe.de/>
- **SuperHTML** (Shareware)
<http://www.superhtml.de/>
- **TSW WebCoder** (Shareware)
<http://www.tsware.net/>
- **Text Hawk** (Shareware)
<http://www.texthawk.com/>
- **TextPad** (Shareware)
<http://www.textpad.com/>
- **UltraEdit** (Shareware)
<http://www.ultraedit.com/>
- **Weaverslave** (Freeware – Eigenlizenz)
<http://www.weaverslave.ws/>
- **Webocton Scriptly** (Freeware)
<http://scriptly.webocton.de/>

- **Zeus** (Shareware)
<http://zeusedit.com/>

Universaleditoren für Apple Mac OS X (Auswahl)

- **BEdit** (nur käuflich zu erwerben)
<http://www.barebones.com/products/bbedit/>
- **Bluefish** (OpenSource – GNU GPL)
<http://bluefish.openoffice.nl/>
- **Coda** (Shareware)
<http://www.panic.com/coda/>
- **Espresso** (Shareware)
<http://macrabbit.com/espresso/>
- **Pagespinner** (Shareware)
<http://www.optima-system.com/>
- **Smultron** (Freeware)
<http://smultron.softonic.de/mac>
- **SubEthaEdit** (Shareware)
<http://www.codingmonkeys.de/subethaedit/>
- **TextMate** (Shareware)
<http://macromates.com/>
- **TextWrangler** (Freeware)
<http://www.barebones.com/products/TextWrangler/>
- **skEdit** (Shareware)
<http://www.skti.org/skedit/>

Universaleditoren für Linux (Auswahl)

- **Bluefish** (OpenSource – GNU GPL)
<http://bluefish.openoffice.nl/>
- **gEdit** (OpenSource – GNU GPL)
<http://projects.gnome.org/gedit/>
- **Kate** (OpenSource – LGPL)
<http://kate-editor.org/>
- **Quanta** (Freeware)
<http://quanta.kdwebdev.org/>
- **SciTE** (OpenSource – Eigenlizenz)
<http://www.scintilla.org/SciTE.html>
- **Screem** (Freeware)
<http://www.screem.org/>
- **Vi** (Unix-Standardeditor)
<http://de.wikipedia.org/wiki/Vi>
- **WebMaker** (OpenSource)
http://www.pro-linux.de/t_wm/webmaker.html

Plattformunabhängige Universaleditoren (Auswahl)

- **Eclipse PDT** (OpenSource – Eclipse Public License)
<http://www.eclipse.org/pdt/>
- **Emacs** (OpenSource – GNU GPL)
<http://www.gnu.org/software/emacs/>
- **jEdit** (OpenSource – GNU GPL)
<http://www.jedit.org/>
- **Vim** (Charity-Ware – GNU-GPL-kompatibel)
<http://www.vim.org/>

1.4.2 Web-Browser

Als Entwickler von Webseiten sollten Sie nicht nur einen Browser auf Ihrem Rechner installiert haben, sondern verschiedene Browser, auch ältere Versionen. Mit welchem Browser Sie persönlich am liebsten surfen, bleibt Ihnen überlassen. Aber die Besucher Ihrer Seiten verwenden nun mal unterschiedliche Produkte, und deshalb sollten Sie Ihre Seiten stets mit mehreren Produkten überprüfen. Denn auch wenn Sie sich an die Webstandards von HTML halten, gibt es leider immer noch viele kleine, ärgerliche Unterschiede zwischen den Browsern und Browser-Versionen im Detail.

Verantwortlich für die Unterschiede ist letztlich die Layout-Engine oder Rendering-Engine des jeweiligen Browsers. Bei allen modernen Browsern ist dieses Software-Kernstück von den übrigen Software-Funktionen (z. B. HTTP-Kommunikation oder Cache-Verwaltung) getrennt. Führende Rendering-Engines, wie etwa die Gecko-Engine oder die Webkit-Engine, werden sogar unabhängig von jeder Browser-Software weiterentwickelt. Die Rendering-Engine ist für das Parsen von HTML zuständig, für das Auflösen von CSS-Definitionen sowie für das Anzeigen von Grafiken.

<i>Web-Browser</i>	<i>Rendering-Engine</i>	<i>Noch relevante Versionen</i>
Firefox	Gecko	ab ca. 12.0
Internet Explorer	Trident	8.0, 9.0
Google Chrome	WebKit	ab ca. 20.0
Safari	WebKit	4.x, 5.x, 6.x
Opera	Presto	10.x, 11.x, 12.x

Neben modernen grafischen Browsern empfiehlt sich auch die Installation des Text-Browsers Lynx. Wenn Sie Webseiten mit diesem Browser testen, bekommen Sie eher ein Gefühl dafür, wie Webseiten-Inhalte z. B. in Spezial-Web-Clients mit Braille-Zeilen oder mit akustischer Sprachausgabe wiedergegeben werden. Gemeint sind damit Aspekte wie die Wiedergabereihenfolge von Webseitenbereichen oder die Wirkung von Navigationen.

Wichtige Web-Browser

- **Firefox** (OpenSource – MPL/GPL/LGPL für Windows, Mac, Linux)
<http://www.mozilla.org/de/firefox/fx/>

- **Microsoft Internet Explorer** (proprietäre Software für Windows)
<http://windows.microsoft.com/de-DE/internet-explorer/products/ie/home>
- **Google Chrome** (OpenSource – BSD-Lizenz für Windows, Mac und Linux)
<http://www.google.com/chrome>
- **Safari** (proprietäre Software mit OpenSource-Teilen für Mac und Windows)
<http://www.apple.com/de/safari/>
- **Opera** (proprietäre Software, Freeware für Windows, Mac und Linux)
<http://de.opera.com/>
- **Lynx** (OpenSource – GNU GPL für MS DOS und andere Textumgebungen)
<http://lynx.isc.org/>

Browser-Statistik

Eine sehr gute und unabhängige Übersicht über Verbreitung und Marktanteile einzelner Browser bietet die englische Wikipedia. Dort werden die jeweils aktuellen Ergebnisse verschiedener bekannter Browser-Statistik-Anbieter wiedergegeben.

 <small>Links</small>	Browser-Statistik von Wikipedia (Englisch) <i>http://en.wikipedia.org/wiki/Browser_statistics</i>
	W3Schools – monatlich aktualisierte Browser-Statistik (Englisch) <i>http://www.w3schools.com/browsers/browsers_stats.asp</i>
	Browser-Marktanteile nach Typ und Version gegliedert <i>http://www.browser-statistik.de/statistiken/versionen/</i>

1.4.3 Desktop-Tools

Es gibt eine Menge von Programmen und Software-Typen, die Web-Entwickler sinnvoll einsetzen können. Nachfolgend eine Übersicht.

FTP/SFTP/SSH-Programme

Um statische Dateien eines Webprojekts auf einen Server ins Web zu laden, brauchen Sie Software, die das File Transfer Protokoll (FTP) bzw. die Secure-Variante SFTP unterstützt. Mit solchen Programmen können Sie Ihre Projektdateien hochladen und auf dem Serverrechner verwalten (umbenennen, löschen usw.). Für weiterreichende Arbeiten auf dem Server benötigen Sie einen Remote-Zugang. Über einen SSH-Client können Sie eine Remote-Shell auf einem entfernten Rechner starten, sofern dort ein SSH-Server läuft. Auch per VPN getunnelte Remote-Desktop-Verbindungen sind geeignet.

Wichtige FTP/SFTP/SSH-Programme

- **FileZilla** (FTP/SFTP, OpenSource GPL für Windows, Mac, Linux)
<http://filezilla-project.org/>
- **PuTTY** (SSH, OpenSource GPL für Windows, Linux)
<http://www.chiark.greenend.org.uk/~sgtatham/putty/>

- **SpeedCommander** (Dateimanager mit FTP/SFTP, Shareware für Windows)
<http://www.speedproject.de/speedcommander/>
- **Total Commander** (Dateimanager mit FTP/SFTP, Shareware für Windows)
<http://www.ghisler.com/deutsch.htm>

Grafikprogramme

Grafikprogramme sind eine wichtige Ergänzung zum Erstellen aussagekräftiger Webseiten. Es gibt unzählige Programme zum Erstellen, Konvertieren und Nachbearbeiten von Grafiken. Auf jeden Fall müssen die eingesetzten Programme die Dateiformate GIF, JPEG und PNG unterstützen, denn das sind die verbreiteten Grafikformate im Web. Zum Entstehungsprozess einer Grafik tragen jedoch häufig auch andere Programme bei, etwa vektororientierte Programme oder Raytracing-Programme.

Wichtige Grafikprogramme

- **Adobe Photoshop** (Kauf-Software – Windows, Mac)
<http://www.adobe.com/products/photoshop/>
- **Adobe Illustrator** (Kauf-Software – Windows, Mac)
<http://www.adobe.com/de/products/illustrator/>
- **CorelDRAW Graphics Suite** (Kauf-Software für Windows)
<http://www.corel.com/>
- **Gimp** (OpenSource GNU-GPL für Windows, Mac, Linux)
<http://www.gimp.org/>
- **IrfanView** (Freeware für Windows)
<http://www.irfanview.com/>
- **Google Picasa** (Freeware für Windows, Mac)
<http://picasa.google.de/>
- **Ulead PhotoImpact** (Kauf-Software für Windows)
<http://www.ulead.com/pi/>

Webserver und Server-Tools

Ein lokal betriebener Webserver ermöglicht es, Webseiten in HTTP-Umgebung am eigenen PC oder im eigenen LAN zu testen. Der Webserver wiederum ist die Voraussetzung für den Einsatz von serverseitiger Software, wie PHP, Datenbankanbindung, etwa mit MySQL, oder den Einsatz von Content-Management-Systemen. Doch selbst wenn Sie nur mit statischen HTML- und CSS-Dateien arbeiten, kann der Einsatz eines Webserver sinnvoll sein – beispielsweise im Hinblick auf die Referenzierung von Grafiken und Multimedia im HTML-Code.

Es ist nicht einmal eine Internet-Verbindung erforderlich, um einen Webserver zu betreiben. Es müssen lediglich TCP/IP-Sockets unterstützt werden, was heute bei jedem Betriebssystem der Fall ist. Auch ein Domain-Name-Service ist für einen lokalen Betrieb eines Servers nicht erforderlich.

Wichtige Server-Software

- **XAMPP Apache Webserver inklusive PHP u. MySQL** (OpenSource für Windows, Linux und Solaris)
<http://www.apachefriends.org/de/>
- **MAMP Apache Webserver inklusive PHP u. MySQL** (Freeware für Mac)
<http://www.mamp.info/de/>
- **Apache Webserver** (OpenSource für Windows oder Unix-Sourcen)
<http://httpd.apache.org/>
- **LightHTTP Webserver** (OpenSource, Unix-Sourcen)
<http://www.lighttpd.net/>
- **Xitami Webserver** (OpenSource für Windows oder Unix-Sourcen)
<http://www.xitami.com/>
- **PHP** (Scriptsprache) (OpenSource für Windows oder Unix-Sourcen)
<http://www.php.net/>
- **Perl** (Scriptsprache) (OpenSource für Windows, Mac, Linux)
<http://www.perl.com/>
- **Python** (Scriptsprache) (OpenSource für Windows, Mac, oder Unix-Sourcen)
<http://www.python.org/>
- **MySQL** (Datenbanksystem) (OpenSource für Windows, Mac, Linux)
<http://www.mysql.com/>
- **PostgreSQL** (Datenbanksystem) (OpenSource für Windows, Mac, Linux)
<http://www.postgresql.org/>

Diverse Tools

Rund um die Realisierung von Webprojekten gibt es, abhängig von Größe, Anzahl und Zusammensetzung der beteiligten Personen, eine Menge anderer Programme und Software-Typen, die Web-Entwickler sinnvoll einsetzen können.

Empfehlenswerte Tools

- **HMTLValidator** (Shareware für Windows)
Umfassendes Validier- und Webseiten-Check-Tool
<http://www.htmlvalidator.com/>
- **XenuLink** (Freeware für Windows)
Rekursiv arbeitender Link-Checker für ganze Websites
<http://home.snafu.de/tilman/xenulink.html>
- **HTML Tidy** (OpenSource für Windows, Unix-Sourcen)
Tool zur Bereinigung und Optimierung von HTML-Code
<http://tidy.sourceforge.net/>

Je nach Art des Webprojekts kann es auch sinnvoll sein, serverseitige Software-Lösungen einzusetzen, da nur so eine effiziente Zusammenarbeit zwischen größeren Teams möglich ist.

1.4.4 Content-Management-Systeme

Fast alle größeren Webprojekte werden heutzutage mit Hilfe von Content-Management-Systemen (CMS) erstellt und gepflegt. Die Gründe sind folgende:

- Ein CMS übernimmt die gesamte Datenspeicherung. Redakteure kümmern sich nur noch um Struktur und Inhalte einer Website, aber nicht mehr darum, wo und wie die ganzen Daten gespeichert werden. Server-Zugänge via FTP/SFTP oder SSH entfallen, da die gesamte Bearbeitung der Web-Inhalte webbasiert im Browser erfolgt.
- Ein CMS erleichtert umfangreiche Änderungen und Umstrukturierungen bei Websites. Da solche Systeme mit Templates arbeiten und Inhalt, Struktur und Layout konsequent trennen, sind Änderungen beim Webdesign unproblematisch und wirken sich automatisch auf alle Seiten aus. Das Gleiche gilt für Umstrukturierungen bei der Navigation.
- Ein CMS verfügt in der Regel über verschiedene Berechtigungs- und Freischaltstufen. Dadurch lassen sich Prozesse innerhalb größerer Redaktionsteams abbilden.

Ein CMS hält Redakteure in der Regel von der Code-Ebene fern. Viele Systeme bieten integrierte WYSIWYG-Editoren oder eine vereinfachte Eingabesyntax an. Dadurch werden einerseits Code-Fehler durch ungeübtes Personal vermieden, und andererseits wird das Einschleusen von problematischem Code vermieden, z. B. das direkte Notieren von JavaScript-Code oder gar Steuersequenzen für das CMS.

Auch Blog-Software gehört zur Kategorie der Content-Management-Systeme. Manche Systeme bieten auch eine integrierte Blog-Funktionalität an. Wenn eine Website vorwiegend aus einem Blog bestehen soll, ist eine Blog-Software wie z. B. WordPress in den meisten Fällen die beste Wahl. Anders sieht es aus, wenn z. B. ganz unterschiedliche Sites unter einer einheitlichen Oberfläche gepflegt werden sollen. In solchen Fällen sind allgemein ausgerichtete Systeme wie TYPO3 oder Drupal von Vorteil.

All diesen Systemen ist gemeinsam, dass die eigentliche Bearbeitung in einem sogenannten **Backend** stattfindet. Das bedeutet, es wird in der Regel eine eigene Domain oder Subdomain eingerichtet, die öffentlich nicht bekannt ist und unter der sich Entwickler, Webdesigner und Redakteure in das System einloggen können. Websites, die mit dem System bearbeitet werden, sind unter anderen eigenen Domains oder Subdomains aufrufbar. Anders verhält es sich mit einer Spezialform von Content Management, nämlich mit Wikis. Bei Wikis findet die Bearbeitung der Inhalte innerhalb der Website, also direkt im **Frontend** statt. Das ist Absicht und unterstreicht den Workshop-Charakter von Wikis. Für viele Websites sind Wiki-Systeme durchaus eine geeignete Wahl, vor allem dann, wenn mit weniger Hierarchien innerhalb von Redaktionen gearbeitet wird, und wenn Web-Inhalte häufig geändert, erweitert oder aktualisiert werden müssen.

Content-Management-Systeme müssen in einer Webserver-Umgebung installiert werden und erfordern je nach Produkt Scriptsprachen wie etwa PHP oder Python und möglicherweise Datenbanksysteme wie etwa MySQL oder PostgreSQL. Die entsprechenden Produkte müssen in den meisten Fällen installiert sein, bevor das CMS installiert wird.

Allgemeine CMS

- **Contao** (OpenSource, Script-Sourcen)
PHP mit MySQL
<http://www.contao.org/>
- **Drupal** (OpenSource, Script-Sourcen)
PHP mit MySQL
<http://drupal.org/>
- **Joomla!** (OpenSource, Script-Sourcen)
PHP mit MySQL
<http://www.joomla.de/>
- **Mambo** (OpenSource, Script-Sourcen)
PHP mit MySQL
<http://mambo-foundation.org/>
- **TextPattern** (OpenSource, Script-Sourcen)
PHP mit MySQL
<http://textpattern.com/>
- **TYPO3** (OpenSource, Script-Sourcen)
PHP mit MySQL und andere
<http://typo3.org/>

Blog-Software

- **Serendipity** (OpenSource, Script-Sourcen)
PHP mit MySQL oder anderen Datenbanksystemen
<http://www.s9y.org/>
- **WordPress** (OpenSource, Script-Sourcen)
PHP mit MySQL
<http://de.wordpress.org/>

Wiki-Software

- **DokuWiki** (OpenSource, Script-Sourcen)
PHP (ohne Datenbank)
<http://www.dokuwiki.org/>
- **MediaWiki** (OpenSource, Script-Sourcen)
PHP mit MySQL
<http://www.mediawiki.org/>
- **Wikidot** (OpenSource, Script-Sourcen)
PHP mit PostgreSQL u. LightHTTP
<http://www.wikidot.org/>



Eine umfangreiche Liste von CMS-Produkten, unterteilt nach Software-Umgebung, findet sich in der englischsprachigen Wikipedia unter:
http://en.wikipedia.org/wiki/List_of_Content_Management_Systems

1.5 Backgrounds

Dieser Abschnitt enthält Hintergrundwissen darüber, wie die maßgeblichen Spezifikationen des W3-Konsortiums zu HTML zustande kommen. Entwickler und Webautoren, die intensiv mit HTML zu tun haben, sollten diese Hintergründe kennen, doch nicht nur wegen der Informationen zum W3-Konsortium, sondern auch wegen der Bemerkungen zu weiteren Quellen sowie zum Geben und Nehmen im Internet, was freiwillige Hilfeleistungen etwa in Fachforen betrifft.

1.5.1 Die Arbeit des W3-Konsortiums

Ende 1994 traf ein erstes Beratungs-Komitee zusammen, um die Grundlagen für eine Institution zu schaffen, die sich um die technischen Grundlagen und Standards im Web kümmern sollte. Denn das CERN, die Geburtsstätte des Web, war mit dieser Aufgabe, die nicht seinem eigentlichen Beschäftigungsgegenstand entsprach, überfordert.

Im Sommer 1995 traten namhafte Firmen dem W3-Konsortium bei. Gleichzeitig eröffnete die europäische Präsenz des Konsortiums ihre Pforten. Mitglieder des W3-Konsortiums sind Firmen oder Organisationen, keine Einzelpersonen. Sie unterschreiben einen 3-Jahres-Vertrag und zahlen Mitgliedsbeiträge, aus denen sich das Gremium finanziert. Als Gegenleistung erhalten sie Zugang zu nichtöffentlichen Informationen und können an der Entwicklung der vom W3C betreuten Standards wie HTML, CSS, XML usw. mitwirken.

Die Arbeit des W3-Konsortiums unterteilt sich in sogenannte **Aktivitäten (Activities)**. Es gibt mehrere Dutzend solcher Aktivitäten. So gibt es Aktivitäten für HTML, XML, CSS usw. Für jede der Aktivitäten gibt es **Arbeitsgruppen (Working Groups)** und **Interessensgruppen (Interest Groups)**. Während die Interessensgruppen eher einflussnehmenden Charakter haben, befassen sich die Arbeitsgruppen mit der eigentlichen Ausarbeitung der Inhalte. Beide Gruppenarten setzen sich aus Mitgliedern des W3C zusammen. In den Arbeitsgruppen sitzen also auch viele Vertreter von Software-Herstellern. Das unabhängige Kern-Team des W3-Konsortiums überwacht die Aktivitäten.

Über die Einstiegsseite des W3-Konsortiums können Sie alle Aktivitäten der Organisation mitverfolgen. Es gibt auch ein deutsches Büro des W3-Konsortiums, das in deutscher Sprache über die Arbeit des W3-Konsortiums berichtet.

Recommendations

Recommendations (Empfehlungen) des W3-Konsortiums sind Dokumente mit dem höchsten im Web verfügbaren Verbindlichkeits-Charakter. Es handelt sich um technische Beschreibungen einzelner Web-Technologien wie beispielsweise HTML, CSS oder XML. Recommendations haben die Aufgabe, die jeweilige Technologie vollständig und möglichst unmissverständlich normativ zu beschreiben. Zielgruppen dieser Beschreibungen sind einerseits Web-Autoren, andererseits aber auch Software-Hersteller, z. B. von Webbrowsern, die angehalten sind, diese Standards so genau und vollständig wie möglich in ihren Produkten zu implementieren.

Der Anspruch der Exaktheit und Vollständigkeit macht die Recommendations für Anfänger nicht unbedingt zu einer leichten Lektüre. Denn nicht selten leidet die Allgemeinverständlichkeit unter den Zwängen der Terminologie. Das W3-Konsortium ist

sich aber bewusst, dass diese Dokumente zum Teil auch von weniger spezialisierten Leuten und Nicht-Informatikern gelesen werden oder gelesen werden müssen. Die neueren Recommendations sind daher reichhaltiger erläutert als die früheren. Auch mit erhellenden Beispielen wird nicht mehr so gespart.

Review-Prozesse

Jede Recommendation durchläuft ein längeres Verfahren, in dessen Verlauf das Dokument mehrmals von verschiedenen Seiten gereviewt (zu Deutsch: geprüft) wird. Aus losen technischen Notizen entstehen zunächst **Working Drafts (Arbeitsentwürfe)**, von denen es einen oder mehrere geben kann. Arbeitsentwürfe lassen das geplante End-Dokument bereits gut erkennen. Verbindlich sind sie jedoch noch nicht. Nach diversen Reviews innerhalb der Arbeitsgruppe und nach Abstimmung mit der Interessensgruppe erlangt das Dokument schließlich den Status einer **Candidate Recommendation (Empfehlungskandidat)**. In diesem Stadium wird noch Feedback aus der Öffentlichkeit erwartet und gegebenenfalls eingearbeitet. Daraus entsteht die **Proposed Recommendation (Empfehlungsvorschlag)**. Nach einem abschließenden Review erhält das Dokument schließlich den Status einer offiziellen Recommendation.

Die Entwicklung der jeweiligen Technologie ist damit allerdings nicht abgeschlossen. Es handelt sich lediglich um festgeschriebene Entwicklungsstadien, damit Entwickler Fixpunkte haben, an die sie sich halten können. Die Recommendations erhalten aus diesem Grund Versionsnummern, ähnlich wie es bei Software-Produkten üblich. So gibt es beispielsweise HTML 4.01, CSS 2.0 oder XML 1.0. Für neue Dokumentversionen wird das Review-Verfahren ganz oder teilweise wiederholt.



Wichtige Web-Adressen des W3-Konsortiums:

Übersichtsseite aller Technical Reports (technische Berichte, Recommendations und Dokumente auf dem Weg dorthin).

<http://www.w3.org/TR/>

Übersichtsseite aller Technical Reports, für die Übersetzungen vom Englischen in andere Sprachen existieren.

<http://www.w3.org/Consortium/Translation/>

Einstiegs- und Übersichtsseite des W3-Konsortiums zu HTML.

<http://www.w3.org/MarkUp/>

Einstiegs- und Übersichtsseite des W3-Konsortiums zu CSS.

<http://www.w3.org/Style/CSS/>

Alle wichtigen Recommendations des W3-Konsortiums liegen im HTML-Format vor und können von den Webseiten des W3-Konsortiums in komprimierter Form heruntergeladen werden. Auf diese Weise kann sich jeder Web-Entwickler lokale Kopien derjenigen Dokumente besorgen, die er für seine tägliche Arbeit benötigt. Wenn Sie sich ernsthaft mit HTML, CSS, XML usw. beschäftigen wollen, ist es dringend zu empfehlen, sich diese Dokumente zu besorgen und sich damit zu beschäftigen.

Die Rolle der WHATWG

Was die Entwicklung von HTML betrifft, so hatte das W3-Konsortium keine besonders glückliche Hand: erst eine einseitige Ausrichtung auf wissenschaftlich-technische Dokumente (HTML 2.0, HTML 3.0), dann eine Anbiederung an schlechtes Defacto-Markup in führenden Browsern (HTML 3.2) und schließlich ein komplizierter, für viel Verwirrung sorgender Kompromiss (HTML 4.01 in drei verschiedenen Varianten, plus XHTML, das zudem als separate Sprache weiterentwickelt werden sollte). Das alles führte schließlich dazu, dass sich eine Aktivisten-Gruppe aus dem W3-Konsortium abgespaltete und mit der WHATWG ein eigenes Standardisierungsgremium gründete.

	<p>WHATWG steht für Web Hypertext Application Technology Working Group. Die Mitglieder der 2004 gegründeten Organisation sind vor allem Interessensvertreter von Browser-Anbietern, also Angestellte der Mozilla Foundation, von Opera Software, Apple Inc. und Google.</p>
	<p>Homepage der WHATWG. http://www.whatwg.org/</p>

Die Entwicklung von HTML5 geht maßgeblich auf die WHATWG zurück. Allerdings geht es der WHATWG nicht darum, das W3-Konsortium zu unterwandern. Die Organisation versteht sich als Zuarbeiterin für das W3-Konsortium, speziell im Bereich der für die Webpraxis relevanten Technologien. In der Praxis sieht das so aus, dass auf den Seiten der WHATWG immer die neueste HTML5-Spezifikation zu finden ist, während das W3-Konsortium daraus festgeschriebene und definierte Dokumentversionen entwickelt.

Im Sommer 2012 kam es zu einem Zerwürfnis zwischen Ian Hickson, dem federführenden Kopf hinter der WHATWG, und dem W3-Konsortium. Seitdem ist HTML auf dem Weg, in zwei Standards zu zerfallen: auf der einen Seite der sogenannte Living Standard der WHATWG, in dem es keine HTML-Versionen mehr geben soll, und auf der anderen Seite der **in Versionen festgeschriebene** Standard des W3-Konsortiums. So gibt es beim W3-Konsortium mittlerweile Arbeiten zu einer HTML-Version 5.1, in die sogar neue Sprach-Features einfließen, die nicht mehr aus dem Living Standard der WHATWG stammen – etwa ein neues HTML-Element namens `template`. Spätestens damit zerfällt HTML in ein WHATWG-HTML und ein W3C-HTML.

Gegenwärtig ist es allerdings eher Ian Hickson mit seinem Living Standard, der HTML sein Gepräge gibt. Auf ihn und die WHATWG gehen fast alle wichtigen Neuerungen in HTML5 zurück, und der große Erfolg von HTML5 unter Webentwicklern beweist, dass Ian Hickson trotz seiner umstrittenen Machtposition ein hervorragendes Gespür für eine sinnvolle Weiterentwicklung der Sprache hat. Auch das vorliegende Handbuch orientiert sich am Living Standard. Trotz des immer weiter um sich greifenden Erfolgs von HTML5 muss man die Weiterentwicklung der Sprache derzeit kritisch beobachten.

1.5.2 Quellen, Dokumentation und Hilfe im Web

Neben den Dokumenten des W3-Konsortiums und der WHATWG gibt es zahllose weiterführende Orte im Web, an denen Web-Entwickler und Webdesigner Antworten auf Fragen oder Inspiration finden.

Verschiedene Informationsquellen im Web

Im heutigen Web gibt es ganz unterschiedliche Quellen für fachliche Informationen. Neben systematischen Dokumentationen gibt es auch Online-Magazine und einflussreiche Fach-Blogs, in denen sich Web-Entwickler und Webdesigner informieren. Da diese Publikationen artikelorientiert sind, empfiehlt es sich, die Quellen regelmäßig aufzusuchen oder deren angebotene Feeds in einem FeedReader zu abonnieren.

 <small>Links</small>	Bekannte Dokumentationen SELFHTML http://de.selfhtml.org Einführung in XHTML, CSS und Webdesign http://jendryschik.de/wsdev/einfuehrung/ W3 Schools http://www.w3schools.com/
	Bekannte Magazine und Blogs Dr.Web http://www.drweb.de/ A List Apart http://www.alistapart.com/ Smashing Magazine http://www.smashingmagazine.com/ WebReference http://www.webreference.com/

Support im Web

Freie Diskussionsorte im Netz sind vorwiegend für Probleme mit softwareunabhängigen Technologien wie HTML oder für Probleme mit OpenSource-basierten Technologien gedacht. Wenn Sie Software käuflich erworben haben und Probleme mit dem Produkt haben, ist immer noch der Produkthanbieter die erste Anlaufstelle für Fragen. So unterhalten beispielsweise große Software-Anbieter wie Microsoft oder Adobe neben den üblichen Call-Centern eigene Diskussionsforen auf ihren Web-Seiten oder in Form von Newsgroups.

 <small>Links</small>	Einige für Web-Entwickler wichtige Diskussionsorte SELFHTML-Forum http://forum.de.selfhtml.org/ Strohalm.org http://www.strohalm.org/ XHTML-Forum http://xhtmlforum.de/ Newsgroup de.comm.infosystems.www.authoring.misc http://groups.google.de/group/de.comm.infosystems.www.authoring.misc/
---	--

6 Hyperlinks

- *Wie Sie Seiten Ihres Web-Projekts untereinander verlinken*
- *Wie Sie auf beliebige Inhalte im Netz verlinken*
- *Wie Sie Hyperlinks mit zusätzlichen Informationen versehen*

6.1 Aufbau von Hyperlinks

Mit Hilfe von Hyperlinks (Verweisen) strukturieren Sie Ihr Projekt. Wenn Sie beispielsweise eine HTML-Datei mit einer Projekt-Einstiegsseite und verschiedenen HTML-Dateien für Themenseiten planen, brauchen Sie zumindest in der Einstiegsseite Verweise zu allen Themenseiten und in jeder Themenseite einen Rücksprungverweis zur Einstiegsseite. Erst dadurch wird aus der losen Dateisammlung eine zusammenhängende Website. Hyperlinks dienen aber ebenso der Vernetzung eigener Inhalte mit anderen Inhalten im Netz.

Für Hyperlinks in HTML ist das `a`-Element zuständig. In diesem Abschnitt werden die möglichen Ziele für Verweise und deren Adressierungsmöglichkeiten behandelt.

6.1.1 Links zu beliebigen Zielen

Sie können Links zu anderen Webseiten des eigenen Projekts definieren (projektinterne Links), Links zu fremden Websites oder zu beliebigen URLs innerhalb anderer Websites (sogenannte Deeplinks).

Beispiel

```
<p>
  In <a href="../kapitel-3/c.html">Kapitel <b>3.c</b></a> wird
  der Beutelteufel näher behandelt. Auch der Artikel
  <a href="http://de.wikipedia.org/wiki/Beutelteufel" target="_blank">
  Wikipedia: Beutelteufel</a> empfiehlt sich als weitere Lektüre.
</p>
<footer>
  <a href="/index.html">
    
  </a>
</footer>
```

Erläuterung

Hyperlinks in HTML werden durch `<a>...` markiert (`a` = *anchor* = Anker). Im `href`-Attribut geben Sie das Verweisziel an (`href` = *hypertext reference* = Hypertext-Refere-

renz). Erlaubt ist als Verweisziel alles, was im Buchabschnitt über Referenzierung (siehe 2.1.5 Referenzierung) beschrieben ist. Der Elementinhalt von `<a>...` ist für den Anwender anklickbar und führt beim Anklicken zum angegebenen Verweisziel.

Das obige Beispiel enthält drei Links: Der erste führt zu einer anderen HTML-Datei im aktuellen Projekt, wobei relative Adressierung zu einem anderen Verzeichnis verwendet wird. Der zweite Link führt zu einem fremden Web-Projekt (Wikipedia). Der dritte Link schließt als Elementinhalt anstelle eines Textes eine Grafikreferenz ein.

Mit Hilfe des Attributs `target` können Sie veranlassen, dass das Verweisziel nicht im aktuellen Browser-Fenster geöffnet wird, sondern in einem neuen Fenster/Tab bzw. in einem anderen Kontext (z. B. in der Sidebar, in einem Popup-Fenster oder in einem eingebetteten Frame-Fenster). Der häufigste Fall bei gewöhnlichen Hyperlinks dürfte es sein, den Link in einem neuen Browser-Fenster/Tab zu öffnen. Dazu notieren Sie wie im obigen Beispiel `target="_blank"` (mitsamt dem Unterstrich).



Bild 6.1: Darstellung von Hyperlinks, a-Elementen, im Browser

Weitere Hinweise

Der Verweistext kann neben Text auch andere HTML-Elemente enthalten, so wie im obigen Beispiel etwa auch Grafikreferenzen (in dem Fall wird die Grafik anklickbar). Im Gegensatz zu HTML 4.01 darf ein a-Element in HTML5 sogar gruppierende Elemente wie `p`, `ul`, `blockquote` usw. umfassen.



Das `target`-Attribut gehört mit HTML5 wieder vollumfänglich zum Standard.

In HTML 4.01 war es in der Strict-Variante nicht mehr erlaubt. Es wird jedoch empfohlen, sparsam davon Gebrauch zu machen. Denn normalerweise ist es besser, dem Anwender die Entscheidung zu überlassen, ob er einen Link in einem anderen Fenster öffnen möchte oder nicht. Neben dem Wert `"_blank"` sind auch die festen Werte `"_self"` (aktuelles Fenster), `"_parent"` (Elternfenster), `"_top"` (oberste Fensterebene) sowie Namen von Fenstern erlaubt, die mit JavaScript geöffnet und deren Namen in JavaScript vergeben wurden.

Die anklickbaren Verweistexte werden von Web-Browsern normalerweise unterstrichen und in unterschiedlichen Farben dargestellt, je nachdem, ob Sie das Verweisziel zuvor

schon einmal aufgerufen haben oder nicht. All diese Eigenschaften können Sie mit Hilfe von CSS bzw. CSS-Eigenschaften nach Wunsch beeinflussen.

Verweise auf Ihrer Website lenken die Aufmerksamkeit des Anwenders auf sich. Wenn Sie Verweise mitten im Text notieren, ist es besser, als Verweistext inhaltlich beschreibende Wörter anzubieten als Wörter ohne Inhalt.

Schlecht ist: zurück (wohin eigentlich?)

Besser ist: zurück zu Kapitel 3

Schlecht ist: Für weitere Information klicken Sie hier (was gibt es denn »hier«?)

Gut ist: Weitere Information können Sie ebenfalls aufrufen.

Beachten Sie im Zusammenhang mit Links auf Ziele außerhalb des eigenen Webangebots auch den Buchabschnitt über rechtliche Aspekte von Links (siehe Abschnitt 6.3.2 *Rechtliche Aspekte von Hyperlinks*).

Referenzinformationen

Elementreferenz <a>,

Attributreferenz href:



Attributreferenz target:



6.1.2 Download-Verweise

HTML5 führt die Möglichkeit ein, Links unabhängig vom MIMEType des Linkziels als Download-Verweis anzubieten. Damit wird eine seit jeher häufig nachgefragte, aber bislang nicht befriedigend gelöste Funktionalität auf einfache Weise mit Hilfe von HTML gelöst.

Beispiel

```
<h1>Aktuelle Downloads</h1>
<ul>
  <li><a download href="meeting-2013-06-28.pdf">Report
    vom Meeting 28.06.2013</a></li>
  <li><a download="weekly-overview.mp3" href="/podcasts.php?current">
    Aktuelle Wochenschau</a></li>
</ul>
```

Erläuterung

- Indem Sie im einleitenden `<a>`-Tag eines Hyperlinks das Attribut `download` notieren, weisen Sie einen Browser, der das Attribut kennt, an, das Verzweiziel in jedem Fall zum Download anzubieten, auch wenn er es selber anzeigen könnte oder ein Plugin oder ein definiertes Programm kennt, das normalerweise für solche Ziele aufgerufen wird.
- Wenn die verlinkte Ressource einen Dateinamen hat, der in dieser Form auch zum Download angeboten werden kann, genügt es, das Attribut als Standalone-Attribut zu notieren, so wie im ersten Fall im obigen Beispiel. Wenn das verlinkte Ziel keinen brauchbaren Dateinamen hat oder wenn Sie einen bestimmten Dateinamen als vorgeschlagenen Dateinamen zum Download anbieten möchten, weisen Sie ihn dem `download`-Attribut als Wert zu. Für den zweiten Link wird in obigem Beispiel davon Gebrauch gemacht. Wenn Sie XHTML-Kodierung verwenden, bietet es sich an, in jedem Fall einen Dateinamen als Wert zuzuweisen.

Weitere Hinweise

- Für Browser, die das `download`-Attribut noch nicht kennen, sollten Sie Alternativen anbieten. Es gibt Dateitypen, die (fast) jeder Web-Browser so interpretiert, dass er dem Anwender anbietet, die Datei lokal zu speichern. Das bekannteste Dateiformat dafür ist heute das ZIP-Format (**.zip*). ZIP-Dateien sind Archivdateien, die mehrere andere Dateien enthalten können, sogar ganze Verzeichnisstrukturen. Die enthaltenen Dateien werden außerdem komprimiert. Der Anwender muss die ZIP-Datei nach dem Download mit einem geeigneten Programm entpacken (Unzip-Programm).
- Attributreferenz `download`:



6.1.3 Links zu beliebigen Ressourcen

Außer dem `download`-Attribut gibt es in HTML keine spezifische Notation, um für Dateien beim Anklicken ein bestimmtes Verhalten anzuweisen. Stattdessen entscheidet der Browser auf Grund der in seiner Konfiguration hinterlegten »Handler« für bestimmte Inhaltstypen (*MimeTypes*), wie er mit einer Ressource des entsprechenden Typs verfahren soll. Als Inhaltsanbieter haben Sie darauf keinen Einfluss. Generell ist es sinnvoll, Links nur auf Ressourcen mit einigermaßen verbreiteten Dateiformaten zu setzen. Bei Dokumenten aller Art ist das vor allem das PDF-Format, das im Gegensatz etwa zu Microsoft-Office-Dokumenten plattformunabhängig und auch nicht so einfach editierbar ist.

Beispiel

```
<h1>Ressourcen aller Art</h1>
<ul>
```

```

<li><a href="katzenposter.pdf">Katzenposter</a> [PDF]</li>
<li><a href="katze-in-aufruhr.avi">Katze in Aufruhr</a> [AVI]</li>
<li><a href="katzenportrait.psd" type="image/x-photoshop">
  Katze in Aufruhr</a> [AVI]</li>
</ul>

```

Erläuterung

Verweise auf beliebige Dateitypen unterscheiden sich prinzipiell nicht von anderen Verweisen. Das Verweisziel ist einfach die gewünschte Datei. Es sind alle Möglichkeiten der Referenzierung möglich.

Bei verbreiteten Formaten wie PDF oder AVI können Sie in der Regel davon ausgehen, dass die Browser der Anwender wissen, wie sie mit solchen Ressourcen umgehen sollen. Wenn es sich um speziellere Formate handelt, etwa um proprietäre Formate wie eine Photoshop-Datei (PSD; letzter Link im obigen Beispiel), ist es zweckmäßig, dem Browser den Internet-MimeType des Formats explizit mitzuteilen. HTML bietet jedoch die Möglichkeit an, im einleitenden `<a>`-Tag ein `type`-Attribut zu notieren. Darin können Sie den MIME-Type der Zieldatei angeben. Diese Information ist für den Web-Browser (und andere Web-Clients) hilfreich. Eine Liste bekannter MIME-Typen finden Sie in der Referenz der MIME-Typen.



Bild 6.2: Darstellung eines Hyperlinks zum Speichern einer zip-Datei

Weitere Hinweise

Wenn Sie Dateien anbieten, die nur für bestimmte Rechnerumgebungen gedacht sind, können Sie natürlich auch Formate verwenden, die speziell für diese Umgebungen gedacht sind. Für DOS/Windows-Umgebungen können das beispielsweise selbstentpackende EXE-Archive sein oder für Apple HQX-Archive. Wenn der Web-Browser mit der Dateiendung nichts anfangen kann, sollte er den Anwender im Dialog entscheiden lassen, was er mit der Datei tun möchte – dabei sollte auch die Download-Möglichkeit angeboten werden.

Bei größeren Dateien, deren Download bei niedrigeren Bandbreiten eine Weile dauert, ist die Angabe der Größe sinnvoll.



Bild 6.3: Hyperlinks zum Speichern verschiedener MIME-Typen

Weitere Hinweise

Dateitypen wie reine Textdateien (*.txt) kann der Web-Browser selbst anzeigen.

Viele Web-Browser haben eine Plugin-Schnittstelle. Wenn der Anwender ein Plugin zur Darstellung des Dateityps besitzt, kann der Browser die Datei mit Hilfe des Plugins selbst anzeigen bzw. abspielen.

Wenn der Anwender ein Programm besitzt, das den Dateityp verarbeiten kann, und dem Web-Browser die Verknüpfung zwischen Dateien mit der Endung des Verweisziels und einem Programm bekannt ist, das solche Dateien verarbeitet, dann kann der Browser das Programm starten. Wenn das Betriebssystem, der Web-Browser und das andere Programm den dynamischen Datenaustausch zwischen Programmen erlauben, kann das Anzeigefenster des Fremdprogramms in das Browser-Fenster eingebettet werden. Es gibt jedoch keine Möglichkeit, solche Dinge in HTML zu beeinflussen.

Wenn der Browser mit dem Dateityp gar nichts anfangen kann, sollte er dem Anwender einen Dialog anbieten, um zu entscheiden, was mit der Datei geschehen soll. Der Anwender sollte die Datei z. B. downloaden können.

6.1.4 E-Mail-Links

Sie können auch Links zu E-Mail-Adressen setzen. Wenn der Anwender auf den Verweis klickt, kann er eine E-Mail an den betreffenden Empfänger absetzen. Eine Garantie dafür, dass dies funktioniert, gibt es jedoch nicht.

Beispiel

```
<h1>Impressum</h1>
<p>Verantwortlich für den Inhalt dieser Seiten ist:<br>
  <a href="mailto:fritz.eierschale@example.org">Fritz
  Eierschale, fritz.eierschale@example.org</a>
</p>
```

Erläuterung

Links zu E-Mail-Adressen funktionieren nach dem gleichen Schema wie alle anderen Verweise. Beim href-Attribut des einleitenden <a>-Tags wird das Verweisziel angegeben. E-Mail-Verweise beginnen dabei immer mit `mailto:` (**ohne // dahinter!**).

Der Verweis ist bei Anwendern nur ausführbar, wenn der Web-Browser das Erstellen und Absenden von E-Mails unterstützt oder wenn bei solchen Verweisen automatisch ein E-Mail-Programm gestartet wird.



Bild 6.4: Darstellung eines Hyperlinks, zum Versand einer E-Mail

Weitere Hinweise

Es ist sinnvoll, im Verweistext die E-Mail-Adresse noch einmal explizit zu nennen, so wie im obigen Beispiel, damit Anwender, bei denen der E-Mail-Verweis nicht ausführbar ist, auf Wunsch separat eine E-Mail senden können.

Beachten Sie, dass das Angeben einer E-Mail-Adresse im Web generell dazu führt, dass eine solche Mailadresse vermehrt mit Spam zu rechnen hat. Eine entsprechende Adresse sollte deshalb über gute, vorgeschaltete Spam-Filter verfügen.

Optionen bei E-Mail-Verweisen

Sie können:

- E-Mail-Verweise an mehrere Empfänger senden, wahlweise `cc` (*carbon copy*, sichtbare Kopienempfänger) oder `bcc` (*blind carbon copy*, unsichtbare Kopienempfänger),
- anstelle einer einfachen E-Mail-Adresse ein vollständiges Empfängerschema angeben von der Art `Fritz Eierschale <fritz.eierschale@example.org>`
- ein `Subject` vordefinieren, sodass beim Öffnen des E-Mail-Editors das `Betreff`-Feld bereits mit einem Text Ihrer Wahl ausgefüllt ist,
- einen `Body`-Inhalt vordefinieren, sodass beim Öffnen des E-Mail-Editors bereits Text im Nachrichtentext der E-Mail steht, z. B. eine Anrede,
- alle Optionen kombinieren.

Mail an einen Hauptempfänger und einen sichtbaren Kopienempfänger:

```
mailto:beispiel@example.org?cc=heidi.bratze@example.org
```

Mail an einen Hauptempfänger und einen unsichtbaren Kopienempfänger:

`mailto:beispiel@example.org?bcc=heidi.bratze@example.org`

Mail mit vordefiniertem Subject (Betreff):

`mailto:beispiel@example.org?subject=eine%20Mail%20von%20deinen%20Web-Seiten`

Mail mit vordefiniertem Body (Nachrichtentext):

`mailto:beispiel@example.org?body=Hallo%20Fritz,%20%0D%0A%0D%0Aich%20wollte%20nur%20sagen,%20dass%20`

Mail mit kombinierten Optionen:

`mailto:fritz.eierschale@example.org?cc=heidi.bratze@example.org&subject=Hallo%20Fritz,%20hallo%20Heidi`

Mail an zwei Hauptempfänger:

`mailto:beispiel@example.org,%20heidi.bratze@example.org`

Mail mit vollständigem Adressierungsschema:

`mailto:Fritz%20Eierschale%20%3Cfritz.eierschale@example.org%3E`

Die Optionen werden in dem Wert notiert, der dem `href`-Attribut zugewiesen wird. Es gibt also keine speziellen HTML-Attribute für diese Optionen, und deshalb ist die Handhabung auch etwas umständlich. Zuerst wird wie üblich der Empfänger notiert. In den obigen Beispielen (mit Ausnahme des letzten) geschieht das jeweils durch `mailto:beispiel@example.org`. Dahinter wird ein Fragezeichen `?` notiert. Das ist in der URI-Syntax das übliche Zeichen, um Parameter an eine aufgerufene Adresse zu übergeben. Hinter dem Fragezeichen folgen die Parameter. Jede Option ist so ein Parameter und besteht aus einem Namen, einem Istgleichzeichen (`=`) und einem zugewiesenen Wert. Als Optionsnamen sind erlaubt:

- `cc` (sichtbarer Kopienempfänger)
- `bcc` (unsichtbarer Kopienempfänger)
- `subject` (Betreff)
- `body` (Nachrichtentext)

Ein Konstrukt wie `cc=heidi.bratze@example.org` ist also ein vollständiger Parameter und bedeutet: sichtbare Kopie an `heidi.bratze@example.org`.

Bei den Wertzuweisungen an die Parameter können Zeichen vorkommen, die nicht zu einer gültigen URI gehören. Damit die URI gültig bleibt (andernfalls wäre das Dokument auch kein gültiges HTML mehr), müssen diverse Zeichen maskiert werden. Die Maskierung besteht darin, ein Prozentzeichen `%` zu notieren, gefolgt von der hexadezimal ausgedrückten Zeichenummer des gewünschten Zeichens.

Die folgende Tabelle listet Zeichen auf, die maskiert werden müssen, weil sie innerhalb von URIs nicht vorkommen dürfen oder eine bestimmte Bedeutung haben. Links steht das Zeichen, rechts die Zeichenkette, mit der Sie das Zeichen maskieren müssen:

<i>Zeichen</i>	<i>Maskierung</i>	<i>Zeichen</i>	<i>Maskierung</i>
[neue Zeile]	%0A	*	%2A
[Wagenrücklauf]	%0D	/	%2F
[Leerzeichen]	%20	<	%3C
!	%21	>	%3E
#	%23	?	%3F
%	%25		

Ferner müssen Sie alle Zeichen maskieren, die nicht in der ASCII-Zeichentabelle vorkommen, also z. B. deutsche Umlaute und das scharfe S (ß). Die folgende Tabelle listet die wichtigsten Zeichen und ihre Maskierung auf:

<i>Zeichen</i>	<i>Maskierung</i>	<i>Zeichen</i>	<i>Maskierung</i>
Ä	%C4	ä	%E4
Ö	%D6	ö	%F6
Ü	%DC	ü	%FC
ß	%DF		

Wenn Sie mehrere Optionen kombinieren wollen, müssen Sie die Optionen durch ein kaufmännisches Und (&) voneinander trennen. Maskieren Sie dieses Zeichen jedoch durch das benannte Zeichen &.

6.1.5 Links und Anker

HTML ist seiner ganzen Struktur nach durchaus so ausgelegt, dass ein einzelnes HTML-Dokument auch umfangreichere Texte, beispielsweise eine komplette Diplomarbeit, enthalten kann. Für Querverweise innerhalb des Dokuments oder für Verweise von außen auf Textstellen innerhalb des Dokuments bietet HTML die Möglichkeit, Zielanker für Hyperlinks an gewünschten Textstellen zu setzen. Links können solche Anker adressieren.

Beispiel

```
<header>
  <a href="#kapitel1">Kapitel 1</a> |
  <a href="#abschnitt1">Abschnitt 1</a> |
  <a href="#abschnitt2">Abschnitt 2</a> |
  <a href="#next">Seitenende</a>
</header>
<hr>
</header>
<hgroup>
  <h1><a id="kapitel1">Kapitel 1</a></h1>
  <h2><a id="abschnitt1">Abschnitt 1</a></h2>
</hgroup>
<p style="height: 200px;">Viel Inhalt</p>
```

```

<h2><a id="abschnitt2">Abschnitt 2</a></h2>
<p style="height: 200px;">Wieder viel Inhalt</p>
<footer>
  <hr>
  <a href="#top">Seitenanfang</a> |
  <a href="#kapitel1">Kapitel 1</a> |
  <a href="#abschnitt1">Abschnitt 1</a> |
  <a href="#abschnitt2">Abschnitt 2</a> |
  <a id="next" href="k02.html#kapitel2">Kapitel 2</a>
</footer>

```

Erläuterung

Ein Anker wird genau wie ein Verweis mit Hilfe des a-Elements erzeugt. Der Unterschied besteht darin, dass kein Attribut href notiert wird, sondern stattdessen das Universal-Attribut id. Ein kompletter Anker sieht also so aus:

```
<a id="ankername">...</a>
```

Um innerhalb einer Datei einen Verweis zu einem im gleichen HTML-Dokument vorhandenen Anker zu notieren, gilt folgendes Schema:

```
<a href="#ankername">...</a>
```

Auch Anker innerhalb anderer HTML-Webseiten sind auf diese Weise adressierbar. Beispiele:

```

<a href="andere-seite.html#ankername">...</a>
<a href="http://example.org/verzeichnis/seite.html#ankername">...</a>

```

a-Elemente sind zugleich Verweis als auch Anker für Verweise, wenn sie sowohl ein href- als auch ein id-Attribut haben, so wie das letzte a-Element im obigen Beispiel.

Den Ankernamen bei id können Sie frei vergeben. Er darf jedoch keine Leerzeichen enthalten. Vergeben Sie möglichst kurze, aber prägnante und lesbare Namen. Beschränken Sie sich auf lateinische Buchstaben, arabische Ziffern sowie den Unterstrich (_). Das erste Zeichen des Namens sollte ein Buchstabe sein. Damit sind Ihre ID-Namen auch gleich kompatibel für den Zugriff via JavaScript und CSS. Groß- und Kleinschreibung werden unterschieden: Ankername wird nicht gefunden, wenn auf #ankername verlinkt wird.

Die Angaben style="height: 200px" in den p-Elementen des obigen Beispiels müssen Sie an dieser Stelle nicht verstehen. Es handelt sich um Style-Angaben, um »Entfernung« zwischen den Ankern zu schaffen.



Bild 6.5: Darstellung der Hyperlinks als Navigation

Weitere Hinweise



Das `id`-Attribut übernimmt erst seit HTML5 die Aufgabe des Ankernamens (wobei das `id`-Attribut selbst mit HTML 4.0 eingeführt wurde).

Bis HTML 4.01 wurde für Anker das Attribut `name` eingesetzt. Im Hinblick auf sehr alte Browser ist es sogar sicherer, mit `name` anstelle von `id` zu arbeiten (Netscape 4.x konnte beispielsweise noch keine Links zu `id`-Ankern ausführen). Der HTML-Standard benötigt jedoch keine Doppelgleisigkeit von `name` und `id` mehr beim `a`-Element. Standardkonform ist deshalb nur noch das `id`-Attribut.

Referenzinformationen

Elementreferenz `<a>`:



Attributreferenz `id`:



6.2 Zusatzinformationen in Hyperlinks

Hyperlinks lassen sich mit zusätzlichen Anweisungen für Suchmaschinen-Robots und Browser versehen, und Informationen über Verweisziele lassen sich vorab beschreiben.



Neu in HTML5 ist darüber hinaus die Möglichkeit, Ping-Benachrichtigungen für Hyperlinks zu definieren.

6.2.1 Angaben zum Linkziel

Sie können bei Links angeben:

- welche Sprache das verlinkte Ziel verwendet,
- zu welchem MIME-Type (Dateiformat) das verlinkte Ziel gehört,
- für welche Medientypen (z. B. Bildschirm, Handheld, Drucker, Screenreader) das verlinkte Ziel gedacht ist.

Beispiel

```
<ul>
  <li>
    <a href="http://www.aawsat.com/" hreflang="ar">Asharq Al-Awsat</a>
    <br>Saudi-Arabische Zeitung in arabischer Sprache
  </li>
  <li>
    <a href="http://www.example.com/spektakel.mp4"
      media="screen, 3d-glasses, print and resolution &gt; 90dpi"
      type="video/mp4">
      Spektakel (der Film)</a><br>
      Spektakulärer 3D-Film (3D-Brille erforderlich)
  </li>
</ul>
```

Erläuterung

Die Angaben zum Linkziel werden in Form von Attributen im einleitenden `<a>`-Tag notiert.

Mit `hreflang` bestimmen Sie die Sprache des Verweisziels. Als Angabe ist ein Sprachenkürzel erlaubt.

Mit `media` können Sie Angaben zu Medien machen, für die das Verweisziel geeignet ist. Die Angabe sollte eine kommasetrennte Aufzählung geeigneter Medientypen, das Schlüsselwort `all` (steht für »alle Medientypen«) oder eine komplexere Medienangabe wie im obigen Beispiel sein.

Mit `type` können Sie den MIME-Type des Linkziel-Dateiformats angeben. Sinnvoll ist diese Angabe in allen Fällen, in denen das Linkziel kein HTML-Dokument ist, also beispielsweise bei Links auf PDF-Dokumente, Flashmovies, MP3-Tracks, usw. Als Angabe ist ein bekannter MIME-Type erlaubt. Siehe Anhang: »MIME-Typen«.

Sie können alle diese Attribute in einem `<a>`-Tag kombinieren.



Bild 6.6: Zusatzinformationen bei Hyperlinks

Weitere Hinweise

Die hier vorgestellten Angaben haben in grafischen Browsern keine sichtbare Wirkung. Dennoch können sie den Browser unterstützen. Beispielsweise betreiben die meisten modernen Browser sogenanntes Prefetching. Sie versuchen also, während eine Webseite angezeigt wird, den darin enthaltenen Links zu folgen und deren Inhalte bereits ganz oder teilweise zu laden. Die hier vorgestellten Angaben zum Linkziel können einem Browser bei der Entscheidung helfen, welche Links beim Prefetching bevorzugt oder ausgeschlossen werden können. Ein anderer Anwendungsfall ist die Formatierung mit CSS. Mit Hilfe sogenannter CSS-Attributselektoren lassen sich Hyperlinks, welche die hier beschriebenen Angaben verwenden, beispielsweise automatisch mit bestimmten Icon-Grafiken versehen (siehe dazu den Buchabschnitt 6.1 *Aufbau von Hyperlinks*).

Referenzinformationen

Elementreferenz `<a>`:



Attributreferenz `hreflang`:



Attributreferenz `type`,

Attributreferenz `media`:



Auf Browser-Angaben wurde hier verzichtet, da nicht oder nicht zuverlässig nachweisbar.

6.2.2 Typisierte Links

HTML bietet die Möglichkeit an, bei Hyperlinks die Art der Beziehung zum Linkziel anzugeben. Das funktioniert genauso wie bei logischer Verlinkung im Kopfbereich des Dokuments. Ebenso wie dort, wird die Beziehung über das `rel`-Attribut hergestellt. Die Linktypen sind jedoch nicht ganz deckungsgleich.

Beispiel

Zum Beispiel ein Formular und dazu ein Link zu einer Seite mit
`Hilfe`.

Erläuterung

Mit dem Attribut `rel` können Sie im `a`-Element, genauso wie im `link`-Element, den Typ der Verlinkung bestimmen (`rel = relationship = Verwandtschaft`).

Das Set möglicher Linktypen weicht beim `a`-Element, gegenüber dem vom `link`-Element leicht ab. Die nachfolgende Tabelle zeigt, welche Linktypen gleich sind und welche unterschiedlich.

<i>rel</i> -Angabe	<i>link</i> -Element	<i>a</i> -Element
<code>rel="alternate"</code>	ja	ja
<code>rel="archives"</code>	ja	ja
<code>rel="author"</code>	ja	ja
<code>rel="bookmark"</code>	nein	ja
<code>rel="external"</code>	nein	ja
<code>rel="first"</code>	ja	ja
<code>rel="help"</code>	ja	ja
<code>rel="icon"</code>	ja	nein
<code>rel="index"</code>	ja	ja
<code>rel="last"</code>	ja	ja
<code>rel="license"</code>	ja	ja
<code>rel="next"</code>	ja	ja
<code>rel="nofollow"</code>	nein	ja
<code>rel="noreferrer"</code>	nein	ja
<code>rel="pingback"</code>	ja	nein
<code>rel="prefetch"</code>	ja	nein
<code>rel="prev"</code>	ja	ja
<code>rel="search"</code>	ja	ja
<code>rel="stylesheet"</code>	ja	nein
<code>rel="sidebar"</code>	ja	ja
<code>rel="tag"</code>	ja	ja
<code>rel="up"</code>	ja	ja

Linktypen für Hyperlinks

Die Linktypen, die im Zusammenhang mit dem `link`-Element im Dokument-Kopfbereich erlaubt sind, sind im Buchabschnitt über logische Verlinkung (siehe 3.1.3 Logische Verlinkung) beschrieben. An dieser Stelle werden deshalb nur die Linktypen beschrieben, die ausschließlich im `a`-Element vorkommen dürfen.

```
<a rel="bookmark">...</a>
```

Mit dieser Angabe definieren Sie, dass der Link zum Anfang des nächsthöheren Abschnitts, Kapitels oder Bereichs führt, zu dem der aktuelle Inhalt gehört (*bookmark* = Lesezeichen). Gedacht ist diese Auszeichnung etwas schwammig zur Rückverlinkung in umfangreicheren, zum Beispiel wissenschaftlichen oder technischen Dokumenten. Die Angabe "bookmark" hat jedoch auch die Bedeutung, sogenannte Permalinks auszuzeichnen, und dafür wird sie in der Praxis durchaus auch verwendet.

Beispiel

```
<a rel="bookmark" href="http://de.wikipedia.org/w/index.php?title=Stefan_M%C3%BCnz&oldid=4115167">Älteste Version des Artikels</a>.
```

Durch die Kennzeichnung als bookmark-Link lässt sich in einem solchen Fall die Tatsache markieren, dass der Link auf eine bestimmte Version eines Artikels führt.

	<p>Permalinks</p> <p>In einigen Content-Management- und Blog-Systemen, besonders aber in Wiki-Systemen, wird jeder Speicherzustand einer Seite bzw. eines Artikels dauerhaft aufgehoben. Jeder einzelne ältere Zustand ist auch über eine URL-Adresse direkt aufrufbar. Hyperlinks, die nicht auf die Seite oder den Artikel verlinken, sondern direkt auf eine bestimmte Version davon, werden als Permalinks bezeichnet.</p>
	<p>Bestes Beispiel sind die Artikel in Wikipedia, zum Beispiel der Artikel über Permalinks: http://de.wikipedia.org/wiki/Permalink</p>

```
<a rel="external">...</a>
```

Diese Angabe erlaubt es, Hyperlinks mit der Bedeutung »dies ist ein Link zu einem anderen Web-Angebot« auszuzeichnen. In Verbindung mit sogenannten Attribut-Selektoren in CSS ist es möglich, solche Links optisch zu kennzeichnen, beispielsweise durch ein spezielles Icon oder durch andere Farben oder Formatierung.

Wenn Sie beispielsweise in einem zentralen Stylesheet-Bereich oder einem externen Stylesheet so etwas definieren wie:

```
a[rel=extern] { color:red; }
```

Dann wird ein Link wie

```
<a href="http://www.example.com" rel="extern"> Link</a>
```

rot dargestellt.

```
<a rel="nofollow">...</a>
```

Mit dieser Angabe weisen Sie Suchmaschinen-Robots an, dem so ausgezeichneten Link nicht zu folgen. Gedacht ist diese Angabe für Umgebungen, in denen Anwender selber Inhalte beitragen und dabei auch anklickbare Hyperlinks erzeugen können. Da es in solchen Fällen immer wieder zu eigentlich unerwünschten Werbe- und Spam-Links kommt, wird die Attraktivität solcher Links zumindest reduziert, wenn Suchmaschinen-

Robots ihnen nicht folgen. Da Google und andere Suchmaschinen diese Angabe tatsächlich beachten, ist sie in der Praxis wirksam.

Eher traurige Berühmtheit erlangte diese Angabe dadurch, dass sich 2007 auch die Wikipedia entschloss, sämtliche externen Links mit dieser Angabe zu versehen und damit Autoren zu unterstellen, sie würden innerhalb von Wikipedia-Artikeln vornehmlich Links zu eigenen Inhalten und damit aus eigennützigen Interessen setzen.

```
<a rel="noreferrer">...</a>
```

Diese Angabe weist den Browser des Anwenders an, beim Anklicken des Links keine Referrer-Adresse zu übermitteln. Das bedeutet: der Webserver der Zieladresse soll keine Information darüber erhalten, woher der Anwender kommt.

Zum Redaktionszeitpunkt ist dies in keinem Browser implementiert.

Referenzinformationen

Elementreferenz `<a>`,

Attributreferenz `rel`:



6.2.3 Links mit Ping-Funktion

Sie können veranlassen, dass der Browser beim Anklicken eines Links nicht nur den Link ausführt, sondern zusätzlich eine oder mehrere andere Webadressen über das Anklicken des Links benachrichtigt. Hinter solchen »anderen Webadressen« verbergen sich dann beispielsweise Scripts, die sogenannte User-Tracking-Informationen sammeln.

Beispiel

```
<a href="produkte.html" ping="http://webstats.example.com/">
  Produkte
</a>
```

Erläuterung

Die Ping-Funktion wird im einleitenden `<a>`-Tag eines Hyperlinks in Form eines `ping`-Attributs notiert. Während die URL-Adresse beim `href`-Attribut wie üblich das Linkziel ist, das beim Anklicken des Links aufgerufen wird, wird die URL-Angabe beim `ping`-Attribut im Hintergrund aufgerufen.

Beim `ping`-Attribut können Sie im Gegensatz zum `href`-Attribut auch mehrere URL-Adressen angeben. Trennen Sie die URL-Adressen in diesem Fall durch Leerzeichen voneinander.

Weitere Hinweise

Die HTTP-Requests, die der Browser an die beim `ping`-Attribut angegebene(n) Adresse(n) versendet, werden in Form von HTTP-POST-Requests versendet.

Aus Anwendersicht bereitet diese Funktion möglicherweise Unbehagen, da ohne sein Wissen zusätzliche Inhalte aufgerufen werden. Es kann sein, dass Browser, die diese Funktion unterstützen, dem Anwender in den Browser-Optionen die Möglichkeit anbieten, keine ping-URLs in Hyperlinks auszuführen. Auch ein sichtbarer Hinweis auf die zusätzlich informierten Adressen ist denkbar.

Referenzinformationen

Elementreferenz `<a>`:



Attributreferenz `ping`:



6.3 Backgrounds

Alle Hyperlinks in HTML haben den gleichen strukturellen Aufbau. Aber Links haben völlig verschiedene Aufgaben, und sie haben als Essenz jeglicher Vernetzung im Web auch nicht-technische Aspekte. In den Backgrounds zu Hyperlinks soll es zum einen darum gehen, wie eine Website letztlich um ihre Navigation herum entsteht. Zum anderen soll ein leidiger Aspekt von Hyperlinks kurz angesprochen werden: die rechtlichen Aspekte von Verlinkung.

6.3.1 Hyperlinks und Navigation

Wenn ein Web-Angebot aus mehr als einer Webseite besteht, was die Regel ist, dann müssen diese Webseiten dem Anwender als zusammenhängendes Angebot zugänglich gemacht werden. Das wichtigste Mittel dazu ist eine Navigation, die auf allen einzelnen Unterseiten wiederkehrt. Nur so (und durch ein gleichbleibendes Layout) entsteht für den Anwender der Eindruck einer Website.

Eine Navigation ist nichts anderes als ein festes Set aus Hyperlinks zu Unterseiten eines Web-Angebots. Die Hyperlinks werden dabei wie üblich durch das `a`-Element markiert. Darüber hinaus benötigen Sie jedoch eine geeignete Markierung des zusammenhängenden Sets.

Aus Sicht von HTML sind Listen das semantisch korrekte Mittel der Wahl, um solche Sets auszuzeichnen. Solange die Einträge der Navigation keine bestimmte Reihenfolge vorgeben, sondern eine freie Auswahl darstellen, sollten Sie ungeordnete Listen (`ul`-Element) verwenden. Vergessen Sie dabei einfach, dass eine ungeordnete Liste in HTML ohne weitere Formatierung nur eine simple, senkrechte Bullet-Liste ist, die optisch nicht viel mit Menüs moderner Websites zu tun hat. Mit Hilfe von CSS können Sie aus `ul`-Listen und darin enthaltenen Links jedoch beliebige vertikale und horizontale Auswahlmenüs mit Mouseover-Effekten und vielem mehr machen.

Vergessen Sie außerdem nicht, die gesamte Navigation in ein `nav`-Element einzuschließen. Das umschließende `nav`-Element ist ein semantisch wichtiges Signal, das eine Aufzählungsliste innerhalb einer Site-Navigation darstellt.

Beispiel einer einfachen Navigation aus HTML-Sicht

```
<nav>
  <ul>
    <li><a href="index.html">HOME</a></li>
    <li><a href="schulbuch.html">Schulbücher</a></li>
    <li><a href="lernsoftware.html">Lernsoftware</a></li>
    <li><a href="musikalien.html">Musikalien</a></li>
    <li><a href="fortbildung.html">Fortbildung</a></li>
    <li><a href="shop.php">Shop</a></li>
    <li><a href="impressum.html">Impressum</a></li>
  </ul>
</nav>
```

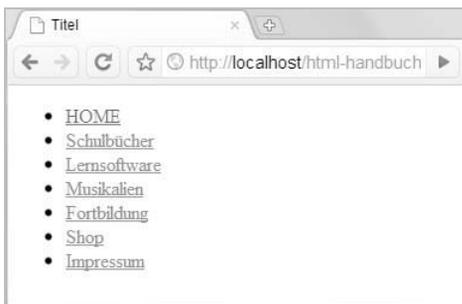


Bild 6.7: Ohne CSS noch schmucklos, aber semantisch sinnvoll: Navigation in HTML

Beispiel einer verschachtelten Navigation aus HTML-Sicht

```
<nav>
  <ul>
    <li><a href="index.html">HOME</a></li>
    <li><a href="schulbuch.html">Schulbücher</a>
    <ul>
      <li><a href="schulbuch-deutsch.html">Deutsch</a></li>
      <li><a href="schulbuch-englisch.html">Englisch</a></li>
      <li><a href="schulbuch-mathe.html">Mathematik</a></li>
    </ul>
    </li>
    <li><a href="lernsoftware.html">Lernsoftware</a>
    <ul>
      <li><a href="lernsoftware-deutsch.html">Deutsch</a></li>
      <li><a href="lernsoftware-englisch.html">Englisch</a></li>
      <li><a href="lernsoftware-mathe.html">Mathematik</a></li>
    </ul>
    </li>
    <li><a href="musikalien.html">Musikalien</a>
```

```

<ul>
  <li><a href="chor.html">Chormusik</a></li>
  <li><a href="instrumental.html">Instrumentalmusik</a></li>
</ul>
</li>
<li><a href="fortbildung.html">Fortbildung</a></li>
<li><a href="shop.php">Shop</a></li>
<li><a href="impressum.html">Impressum</a></li>
</ul>
</nav>

```

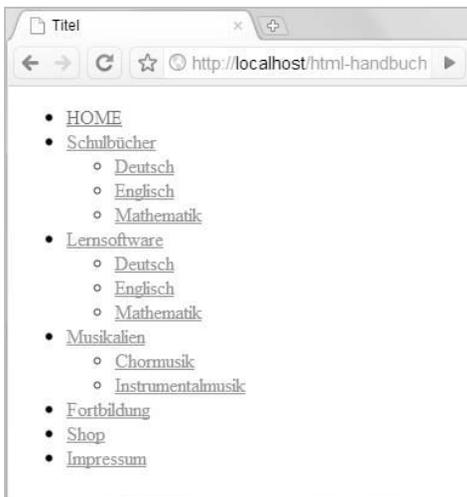


Bild 6.8: Mehrstufige Navigation als verschachtelte Liste in HTML

Erläuterung

Die gesamte Navigation wird innerhalb von `<nav>...</nav>` notiert. Eine ungeordnete Liste (`...`) schließt die Navigationspunkte ein. Jeder Navigationseintrag ist ein Listenpunkt, markiert durch `...`. Der Inhalt der `li`-Elemente ist dann der jeweilige Link, markiert durch `...`.

Im Fall einer verschachtelten (mehrstufigen) Navigationsstruktur enthalten `li`-Elemente, an denen sich ein »Untermenü« anschließt, im Anschluss an den Link, der das Untermenü auslösen soll, eine vollständige `ul`-Liste für das Untermenü. Diese Liste hat wieder den gleichen Aufbau: `li`-Elemente für die Menü-Einträge, welche die `a`-Elemente für die eigentlichen Links enthalten.

Weitere Hinweise

Da die Navigation auf allen Unterseiten wiederholt werden soll, muss der entsprechende HTML-Code in allen Unterseiten wiederholt werden. Wenn Sie mit einem Content-Management-System arbeiten, das die einzelnen Webseiten zur Laufzeit dynamisch generiert, ist das kein Problem. Falls Sie jedoch mit statischen HTML-Dateien arbeiten, haben Sie keine derartigen Möglichkeiten.

Die einfachste Lösung, nämlich den HTML-Code der gesamten Navigation in alle Unterseiten zu kopieren, ist sehr unflexibel. Denn jede Änderung in der Navigation muss dann auf allen Unterseiten nachgezogen werden.



HTML5 sieht eine ebenso simple, wie naheliegende Lösung für dieses Problem vor, die im Gegensatz zu anderen Lösungen keine Serverumgebung und keine andere Technologie als HTML erfordert: nämlich die Navigation in eine eigene HTML-Datei auszulagern und diese einfach in alle gewünschten Webseiten einzubinden. Dazu wird in der HTML5-Spezifikation die folgende Syntax vorgeschlagen.

Zunächst der relevante Code-Ausschnitt der Webseite, in der die Navigation erscheinen soll:

```
<nav>
  <iframe seamless src="navigation.html"></iframe>
</nav>
```

Und dann der Inhalt der darin eingebundenen Datei, hier *navigation.html* genannt:

```
<ul>
  <!-- Inhalt der Navigation -->
</ul>
```

Leider ist das in den meisten Browsern noch nicht so wie gedacht umsetzbar, da diese das *seamless*-Attribut noch nicht umsetzen und stattdessen ein gewöhnliches *iframe*-Fenster anbieten. Das *seamless*-Attribut sollte dafür sorgen, dass der eingebundene Inhalt direkt in den HTML-Code des einbindenden Dokuments integriert wird.

Dennoch ist die Lösung mit dem *iframe*-Element durchaus praktikabel. Auch ohne *seamless*-Attribut lassen sich eingebettete Frames optisch nahtlos in Webseiten integrieren. Mehr dazu finden Sie unter 7.5 Eingebettete Frames.

Ansonsten sind, um eine solche Einbindung zu erreichen, derzeit nur serverseitige, scriptbasierte Lösungen praxistauglich. Wenn die Webseiten beispielsweise auf einem Webserver betrieben werden, auf dem PHP verfügbar ist, müssen die HTML-Dateien mit den Inhalten in **.php*-Dateien umbenannt werden, also so, dass z. B. *impressum.html* zu *impressum.php* wird. Dann genügt an der gewünschten Stelle folgender Code zur Einbindung einer Datei wie der *navigation.html*:

```
<nav>
  <?php
    include("navigation.html");
  ?>
</nav>
```

In diesem Fall findet die Einbindung bereits serverseitig statt, also bevor das gesamte HTML-Dokument überhaupt an den Browser ausgeliefert wird. Deshalb funktioniert eine solche Lösung in allen Browsern.

6.3.2 Rechtliche Aspekte von Hyperlinks

Das Web hat zahlreiche neue Realitäten mit sich gebracht. Deren Integration in bestehende Rechtssysteme ist mühsam und dauert immer noch an. Hyperlinks spielen dabei eine besonders kritische Rolle, da sie den wichtigsten Mehrwert im Web schaffen: nämlich die Nähe aller Inhalte.

Links auf fremde Web-Angebote

Im Normalfall dürfen Sie ungefragt Verweise auf fremde Web-Angebote setzen. Sie brauchen also keine E-Mail an den Anbieter mit Bitte um Genehmigung zu schreiben, wenn Sie auf sein Angebot einen Link setzen wollen. Jeder Anbieter, der mit seinem Web-Projekt online geht, muss sich im Klaren darüber sein, dass er Teil eines weltweiten Hypertext-Systems ist, in dem er nicht allein ist.

Es gibt jedoch Ausnahmen von der Regel. Wenn Sie beispielsweise selbst ein sehr stark frequentiertes Web-Angebot haben und auf dessen Einstiegsseite einen Verweis auf die Homepage eines kleinen, unbekanntes Anbieters setzen, dann sollten Sie ihn vorher fragen. Der Grund: Durch die vielen zu erwartenden Besucher, die über Ihren Verweis auf das fremde Angebot finden, wird dort plötzlich sehr viel Traffic (Besucherverkehr und Datenübertragung) erzeugt. Viele Anbieter haben bei ihrem Provider eine Volumenbegrenzung, und wenn diese überschritten wird, entstehen den Anbietern unkalkulierbare Kosten. Auch könnte es sein, dass der fremde Server nicht sehr belastungsfähig ist und durch die vielen plötzlichen Besucher zusammenbricht.

Eine andere Ausnahme ist, wenn Sie den Verweis in einem negativen Kontext setzen. Wenn Sie also auf einen Anbieter verweisen, nur weil Sie ihn auf Ihren eigenen Seiten heftig kritisieren, dann ist es sicherer, sich dort eine Bestätigung einzuholen, dass ein solcher Verweis gebilligt wird. Juristisch sicher ist eine solche Bestätigung aber nur, wenn sie mit Unterschrift in dokumentenechter Form von einer dafür zuständigen Person geleistet wird.

Deeplinks

Eine Weile lang wollten einige Web-Anbieter verhindern, dass Links von extern in die Tiefen ihres eigenen Web-Angebots führten. Sie wollten erzwingen, dass Links nur zur repräsentativen Startseite führen dürften. Doch das ist letztlich genauso, als dürfe man aus einem Buch nicht zitieren und keine Seitenzahlen nennen.

Deeplinks sind aus heutiger Sicht völlig in Ordnung. Der deutsche Bundesgerichtshof hat diese Auffassung in einem wichtigen Grundsatzurteil aus dem Jahr 2003 bestätigt. Wenn ein Anbieter keine solchen Links will, muss er technische Vorkehrungen treffen, um sie zu verhindern.

Linkhaftung

Die wohl prominenteste und für den Fortbestand des Webs gefährlichste rechtliche Problemzone ist die Frage der Linkhaftung. Obwohl Tim Berners-Lee, der Gründervater des WWW, klar und deutlich Stellung dazu bezogen hat, was ein Link ist, im Gegensatz etwa zu eingebetteten Fremdinhalten, tut sich die Rechtsprechung zuweilen noch schwer

damit. Immer noch wird vor Gericht die »Zeigerfunktion« eines Verweises mit dem Zueigenmachen fremder Inhalte verwechselt.

Argumentationshilfe aus fachlicher Sicht

Ein Link besteht aus zwei Teilen:

- der Erwähnung des anderen Inhalts – sie ist im Verweistext formuliert, und
- einer Adressierungs-Information, nämlich zum Fundort des anderen Inhalts im Internet ("URL").

Das gilt sowohl für Links innerhalb des eigenen Angebots als auch für Links zu fremden Angeboten. Alle Links haben den gleichen Aufbau.

In Bezug auf fremde Inhalte ist ein Link nichts anderes, als wenn ein fremder Inhalt vom Typ »Buch« erwähnt wird und dabei die Informationen für Erscheinungsdatum, Verlag und ISBN hinzugefügt werden.

Den Zugang zum fremden Inhalt stellt nicht der Anbieter des Links her, sondern der Anwender. Und zwar nicht zwangsläufig, sondern willentlich und nur dann, wenn er auf den Verweis klickt und damit die verlinkte Zieladresse aufruft. Das ist nichts anderes, als wenn der Anwender sich entschließt, das erwähnte Buch zu erwerben, dazu Titel, Verlag und ISBN auf einem Zettel notiert, sich in eine Buchhandlung oder Bibliothek begibt und das Buch dort erwirbt oder bestellt. Das Anklicken des Links führt nur zeitlich schneller zum Ziel, stellt aber qualitativ keinen Unterschied zum Gang in die Buchhandlung oder Bibliothek dar.

Wenn sich ein Angebot durch einen Link auf ein fremdes Angebot tatsächlich den fremden Inhalt zu eigen machen würde, dann hätte dies unter anderem folgende wichtige Konsequenzen:

Eine große Suchmaschine, die das Suchen nach beliebigen Begriffen erlaubt und die Suchtreffer in Form von Links zu Fundorten auflistet, würde sich mehr oder weniger das gesamte Web zu eigen machen. Und selbst wenn man Suchmaschinen ausnehmen würde, weil sie weitgehend automatisiert arbeiten, würde die Aussage zumindest auf alle großen redaktionell gepflegten Web-Verzeichnisse zutreffen.

Wer einen Link auf ein fremdes Angebot anbietet, das wieder auf ein anderes fremdes Angebot linkt, das wieder auf ein fremdes anderes Angebot linkt, würde sich im Sinne der infiniten teleologischen Reduktion alle weiteren Inhalte zu eigen machen. Das ist aber offenkundiger Unsinn.

Disclaimer zur Linkhaftung

Manche Web-Anbieter platzieren auf ihren Webseiten einen sogenannten Disclaimer (Haftungsausschluss), in dem jegliche Haftbarmachung für Hyperlinks zu fremden Websites ausgeschlossen wird. Solche Disclaimer sind juristisch gesehen jedoch bedeutungslos, da sie durch keine gesetzliche Regelung gedeckt sind.

Stichwortverzeichnis

Fett gesetzte Seitenzahlen führen zur Referenz bzw. zur grundlegenden Definition des Stichwortes.

&

­ 102

A

Abbildungen 111

abgerundete Ecken 453

ABNF production 280

Absatzkontrolle 438

Absatzschaltung 48

Abschluss-Tag 352

Abschnitt 90

absolute

Positionierung 485

Abstand 444

zu Aufzählungszeichen 473

zu Nummerierung 473

Activities 39

AJAX 25, 245, 248, 551, 566

Aktivitäten 39

alleinstehende Attribute 353

Almost-Standards-Modus 67

Alternativtext 184

Änderungsmarkierungen 128

Änderungszeitpunkt 129

Anführungszeichen 353

Anker 167

anklickbare Grafiken 160

Anweisungen

JavaScript 541

Anzeige leerer Tabellenzellen

480

Anzeigesteuerung 494

Arbeitsentwürfe 40

Arbeitsgruppen 39

Array 555

Array-Objekt 565

Artikel 97

ASCII-Kodierung 63

assoziierter Knoten 554

Atom 29, 75

attributbedingte Formate 389

Attribute 44

abbr 155, 390, 673

abbr (obsolet) 366

above (proprietär) 372

accept 266, 673

accept-charset 248, 674

iso-8859-1 248

utf-8 248

accesskey 286, 674

action 246, 247, 248, 249,

271, 273, 306, 533, 674

align (obsolet) 366

alink (obsolet) 366

alleinstehende 45, 353

allowfullscreen 234, 674

alt 183, 187, 271, 675

archive (obsolet) 366

async 540, 675

autocomplete 249, 287, 675

autofocus 274, 286, 675

autoplay 210, 214, 676

axis (obsolet) 366

background 372

background (obsolet) 366

behavior (proprietär) 372

below (proprietär) 372

bgcolor (obsolet) 366

bgproperties (proprietär) 372

border 143, 184, 231

border (obsolet) 366

bordercolor (proprietär) 372

bordercolordark (proprietär)

372

bordercolorlight (proprietär)
372

cellpadding 143

cellpadding (obsolet) 366

cellspacing (obsolet) 367

challenge 274, 676

char (obsolet) 367

charoff (obsolet) 367

charset 71, 539, 676

charset (obsolet) 367

checked 264, 531, 676

cite 110, 124, 129, 676

class 101, 122, 344, 347, 390,

408, 477, 499, 510, 569, 677

dtstart 346, 347

location 346

vevent 346, 347

classid 227, 366

classid (obsolet) 367

clear 367

clip (proprietär) 372

code 367

codebase 366, 367

codecs

flac 211

speex 211

codetype 367

color 364

color (obsolet) 367

cols 256, 677

colspan 151, 677

command 301

compact (obsolet) 367

content 71, 73, 74, 88, 322,

677

noindex 73

- contenteditable 317, 318, 340, 677
- contextmenu 301, 678
- controls 210, 214, 678
- coords 186, 678
- crossorigin 236, 678
- data 225, 678
- datetime 126, 129, 679
- declare (obsolete) 368
- default 219, 679
- defer 539, 679
- dir 122, 679
 - ltr 122
 - rtl 342
- direction (proprietary) 372
- dirname 288, 679
- disabled 274, 283
- download 162, 680
- draggable 517, 680
- dropzone 338, 680
- dynsrc 373
- dynsrc (proprietary) 372, 373
- einleitende
 - in Tags 44
- encoding 350
- enctype 248, 265, 681
- event (obsolete) 368
- face (obsolete) 368
- fill 195, 198
- font-family 198
- font-size 198
- font-weight 198
- for 251, 681
- for (obsolete) 368
- form 274, 289, 681
- formaction 290, 681
- formenctype 291, 682
- formmethod 290, 682
- formnovalidate 291, 682
- formtarget 291, 682
- frame (obsolete) 368
- frameborder 231
- frameborder (obsolete) 368
- framespacing (proprietary) 372
- Groß- und Kleinschreibung 45
- gutter (proprietary) 373
- headers 683
- height 182, 190, 193, 200, 215, 227, 230, 233, 683
- hidden 683
- high 297, 683
- href 76, 84, 159, 168, 174, 187, 376, 560, 683
- hreflang 170, 187, 684
- hspace (obsolete) 368
- http-equiv 67, 71, 72, 74, 322, 684
 - expires 74
- icon 301, 684
- id 126, 156, 168, 196, 200, 201, 251, 260, 261, 266, 301, 323, 338, 354, 386, 391, 396, 398, 492, 493, 523, 530, 531, 533, 544, 555, 558, 559, 569, 571, 684
- inert 339, 685
- inputmode 281, 685
- internal (proprietary) 373
- ismap 685
- itemid 325, 685
- itemprop 320, 322, 323, 324, 333, 334, 346, 686
- itemref 323, 686
- itemscope 320, 322, 323, 324, 325, 327, 328, 333, 346, 686
- itemtype 324, 325, 328, 333, 336, 346, 686
- keytype 274, 687
 - rsa 274
- kind 218, 687
- label 218, 258, 301, 687
- lang 354, 390, 513, 554, 687
- leere 353
- left (proprietary) 373
- leftmargin (proprietary) 373
- link (obsolete) 368
- list 261, 687
- longdesc 184
- longdesc (obsolete) 368
- loop 211, 214, 688
- loop (proprietary) 373
- low 297, 688
- lowsrc (proprietary) 373
- manifest 688
- marginheight (obsolete) 368
- marginwidth (obsolete) 369
- max 278, 279, 295, 297, 688
- maxlength 254, 256, 257, 266, 688
- media 170, 187, 211, 216, 689
- mediagroup 214, 690
- method 246, 247, 248, 265, 306, 690
- min 278, 279, 297, 690
- mit freier Wertzuweisung 45
- mit freier Wertzuweisung ohne weitere Konventionen 45
- mit Wertzuweisung 44
- multiple 259, 260, 266, 690
- muted 214, 690
- name 71, 169, 186, 226, 231, 247, 256, 259, 260, 268, 272, 278, 292, 322, 354, 569, 691
 - application-name 72
 - author 73
 - creator 73
 - description 72
 - generator 73
 - publisher 74
 - robots 73
- nohref (obsolete) 369
- noresize (obsolete) 369
- noshade (obsolete) 369
- novalidate 249, 691
- nowrap (obsolete) 369
- open 691
- optimum 297, 691
- pattern 278, 281, 692
- ping 174, 692
- placeholder 286, 692
- pluginspage 224
- pluginurl 224
- points 194
- poster 215, 692
- preload 211, 215, 692
- profile (obsolete) 369
- prompt (obsolete) 369
- property 88

- pubdate 693
 - quality 223
 - radiogroup 301, 693
 - readonly 283, 693
 - rel 76, 171, 344, 369, 376, 570, 693
 - alternate 79
 - archives 79
 - author 77
 - bookmark 173
 - first 78
 - help 77
 - icon 80
 - index 77
 - last 78
 - license 80
 - next 78
 - pingback 81
 - prefetch 78
 - prev 78
 - sidebar 79
 - stylesheet 82
 - tag 79
 - up 78
 - required 694
 - rev 344
 - rev (obsolete) 369
 - reversed 107, 694
 - rows 256, 694
 - rowspan 151, 695
 - rules (obsolete) 369
 - sandbox 232, 233, 695
 - scheme (obsolete) 370
 - scope 156, 695
 - col 156
 - colgroup 156
 - row 156
 - rowgroup 156
 - scoped 82, 695
 - scollamount (proprietär) 373
 - scrolldelay (proprietär) 373
 - scrolling 231
 - scrolling (obsolete) 370
 - seamless 178, 233, 234, 696
 - selected 260, 696
 - shape 186, 696
 - circle 186
 - polygon 187
 - rectangle 186
 - size 254, 259, 266, 364, 696
 - size (obsolete) 370
 - sizes 80, 697
 - span 697
 - spellcheck 697
 - src 182, 190, 210, 211, 213, 216, 218, 223, 230, 234, 271, 373, 390, 540, 697
 - srcdoc 234, 698
 - srclang 698
 - srclang= 218
 - standby (obsolete) 370
 - start 107, 698
 - start (proprietär) 373
 - step 278, 698
 - stroke 194, 198
 - stroke-width 194
 - style 116, 168, 260, 272, 379, 386, 396, 398, 426, 451, 474, 480, 481, 532, 562, 698
 - summary 153, 699
 - tabindex 286, 699
 - target 84, 160, 227, 249, 417, 699
 - _blank 85, 160, 249
 - _parent 85, 249
 - _self 85, 249
 - _top 85, 249
 - text (obsolete) 370
 - title 76, 82, 121, 184, 278, 343, 511, 558, 699
 - top (proprietär) 373
 - topmargin (proprietär) 374
 - transform 198
 - translate 341, 700
 - type 79, 80, 82, 83, 163, 170, 187, 211, 215, 223, 226, 253, 265, 268, 271, 277, 300, 301, 348, 364, 378, 390, 538, 540, 700
 - typemustmatch 226, 701
 - unbekannte 344, 348
 - usemap 186, 226, 354, 701
 - value 107, 254, 259, 261, 263, 264, 266, 268, 270, 271, 272, 278, 279, 292, 295, 297, 300, 318, 370, 558, 564, 701
 - valuetype (obsolete) 370
 - version 350
 - version (obsolete) 370
 - vlink (obsolete) 370
 - vspace (obsolete) 370
 - width 143, 147, 182, 190, 193, 200, 215, 227, 230, 233
 - wrap 256, 701
 - hard 256
 - soft 256
 - wrap (proprietär) 374
 - xml:lang 354
 - xmlns 59, 190, 192, 351, 362
 - z-index (proprietär) 374
 - Attributknoten 554
 - Attributnamen 45
 - Attribut-Selektor 173
 - Attributwerte 353
 - aufklappbare Details 303
 - Aufklappmenüs 298
 - Aufzählungszeichen 104
 - Ausgabefelder 291
 - Ausrichtung 438
 - horizontale 438
 - vertikale 438
 - Außenrand 444
 - Auswahllisten 257, 300
 - Auszeichnungssprachen 356
 - Autofocus 284
 - Automatische Übersetzung 341
 - Autorenangaben 98
 - Autoren-Stylesheets 394
 - Autovervollständigung 249, 287
- B**
- Backend 26
 - Background 62
 - Balkengrafiken 192
 - Basis-URL 84
 - Baumstruktur 360
 - Bedingter Trennstrich 102
 - Benannte Zeichen 57

- Benutzer-Stylesheet 394
 - Beschreibungslisten 108
 - Betreff 166
 - bidirektionaler Text 423
 - Bildherkunft 241
 - Block-Element 48, 133, 398, 498
 - Blocksatz 440
 - Blockzitat 109
 - Blog 19
 - Bookmarks 69
 - Boolean-Objekt 565
 - Boxmodell 398
 - Breite
 - der Grafik 182
 - Browser
 - Google Chrome 34
 - Lynx 34
 - Microsoft Internet Explorer 34
 - Mozilla Firefox 33
 - Opera 34
 - Safari 34
 - Browser-Stylesheet 394
 - Browsing-Context 84, 226, 231
 - Bruchzahlen 404
 - Buttons 268
 - BY-Lizenz 81
 - BY-NC-Lizenz 81
 - BY-NC-ND-Lizenz 81
 - BY-NC-SA-Lizenz 81
 - BY-ND-Lizenz 81
 - BY-SA-Lizenz 81
- C**
- Candidate Recommendation 40
 - CanvasRenderingContext2D 198
 - Canvas-Schnittstelle 567
 - Cascading Stylesheets 23, 360, 375
 - CDATA 49, 355
 - Checkboxes 262
 - Client-seitiges Scripting 25
 - CMS 26, 37
 - Backend 37
 - Contao 38
 - Drupal 38
 - Frontend 37
 - Joomla! 38
 - Mambo 38
 - Serendipity 38
 - TextPattern 38
 - Typo3 38
 - WordPress 38
 - Codec 241
 - Code-Editier-Hilfe 30
 - Code-Editor 30
 - Code-Folding 30
 - Code-Fragmente 233
 - inkludieren 233
 - Codetabelle 63
 - Comboboxen 260
 - Conditional Comments 50
 - Container für Elemente 400
 - content
 - nofollow 73
 - Content-Management-Systeme 37
 - Content-Modell 133, 494
 - in HTML 4.01 133
 - in HTML5 134
 - umdefinieren 497
 - Copyright 241
 - Copyright-Angaben 543
 - CORS 236
 - CreativeCommons 80
 - CreativeCommons-Lizenz 80
 - Cross-Origin Resource Sharing 236
 - CSS 23, 360, 375
 - Deklarationen 394
 - Deklarationsblock 393
 - Farbangaben 195
 - in HTML einbinden 375
 - Klasse 408
 - Medieneigenschaften
 - color 383
 - color-index 384
 - device-aspect-ratio 383
 - device-max-width 383
 - grid 384
 - landscape 383
 - max-weight 383
 - min-width 383
 - monochrome 384
 - orientation 383
 - portrait 383
 - resolution 384
 - scan 384
 - Medientyp 395
 - Medientypen
 - all 382
 - braille 382
 - embossed 382
 - handheld 382
 - print 382
 - projection 382
 - screen 382
 - speech 383
 - tty 383
 - tv 383
 - Selektor
 - Attribute 390
 - Syntax 421
 - Universalselektor 397
 - Version 1.0 24
 - Version 2.0 24
 - Version 2.1 24
 - Version 3.0 25
 - Wertangaben 403
 - CSS3-Module
 - 2D Transforms 421
 - 3D Transforms 421
 - Animations 412
 - Backgrounds and Borders 413
 - Basic Box Model 414
 - Basic User Interface 413
 - Behavioral Extensions to CSS 414
 - Cascading and Inheritance 414
 - Color 415
 - Flexible Box Layout 415
 - Fonts 415
 - Generated and Replaced Content 416

- Generated Content for Paged Media 416
- Grid Positioning 416
- Hyperlink Presentation 417
- Image Values and Replaced Content 417
- Line Box 418
- Lists and Counters 418
- Marquee 418
- Multi-Column-Layout 419
- Namespaces 419
- Paged Media 419
- Presentation Levels 420
- Ruby 420
- Speech 420
- Syntax 421
- Template Layout 421
- Text 421
- Transitions 422
- Values and Units 422
- Writing Modes 423
- CSS3-Profile 423
 - Mobile-Profil 423
 - Print-Profil 423
 - TV-Profil 424
- CSS-Eigenschaften
 - alignment 418
 - appearance 715
 - background 407, 463, 466, 715
 - background-attachment 372, 463, 466, 716
 - background-clip 467, 716
 - background-color 366, 380, 400, 407, 455, 462, 463, 477, 716
 - background-image 366, 463, 716
 - background-origin 717
 - background-position 463, 717
 - background-repeat 463, 717
 - background-size 413, 466, 467, 718
 - binding 414
 - bold 428
 - bolder 428
 - border 184, 366, 368, 369, 380, 401, 413, 460, 477, 718
 - border- 449
 - border-bottom 718
 - border-bottom-left-radius 723
 - border-bottom-right-radius 723
 - border-bottom-style 719
 - border-bottom-width 719
 - border-collapse 452, 478, 479, 720
 - border-color 449, 720
 - border-image 449, 456, 457, 720
 - border-image- 413
 - border-image-outset 720
 - border-image-repeat 721
 - border-image-slice 721
 - border-image-slicing 458
 - border-image-source 458, 721
 - border-image-width 722
 - border-left 407, 722
 - border-left-color 722
 - border-left-style 722
 - border-left-width 723
 - border-radius 413, 449, 454, 723
 - border-right 724
 - border-right-color 724
 - border-right-style 725
 - border-right-width 725
 - border-shadow 449
 - border-spacing 143, 367, 479, 480
 - border-style 479, 725
 - border-top 368, 726
 - border-top-color 726
 - border-top-left-radius 724
 - border-top-right-radius 724
 - border-top-style 726
 - border-top-width 727
 - border-width 449, 458, 479, 727
 - bottom 488, 727
 - box-align 415
 - box-decoration-break 727
 - box-direction 415
 - box-orientation 415
 - box-sizing 413, 728
 - caption-side 728
 - clear 367, 491, 728
 - clip 372, 729
 - close-quote 513
 - color 364, 367, 370, 380, 415, 433
 - column-count 419, 729
 - column-gap 419, 729
 - column-rule 419, 730
 - column-rule-color 419, 730
 - column-rule-style 419, 730
 - column-rule-width 419, 731
 - columns 419, 732
 - column-span 419, 731
 - column-width 419, 731
 - condensed 432
 - content 510, 514, 516, 732
 - counter-increment 516, 732
 - counter-reset 516, 732
 - cursor 413, 517, 733
 - direction 733
 - display 400, 494, 497, 498, 499, 734
 - empty-cells 481, 734
 - expanded 432
 - extra-condensed 432
 - extra-expanded 432
 - filter 504
 - float 391, 399, 444, 491, 734
 - font 430, 436, 735
 - font- 415, 427
 - font-family 364, 365, 368, 380, 381, 385, 403, 425, 426, 429, 436, 477, 735
 - font-size 364, 370, 380, 404, 428, 429, 441, 477, 736
 - font-size:large 364
 - font-size:small 365
 - font-stretch 432
 - font-style 428, 429, 434, 736
 - font-variant 428, 429, 736
 - font-weight 428, 429, 434, 737

- grid-columns 416
- grid-rows 416
- height 233, 318, 483, 484, 493, 737
- icon 737
- italic 428
- large 428
- larger 428
- left 373, 488, 493
- letter-spacing 367, 380, 421, 432, 738
- line-height 404, 429, 442, 738
- line-stack 418
- list-style 418, 470, 738
- list-style-image 468, 470, 739
- list-style-position 468, 470, 739
- list-style-type 468, 469, 470, 474, 739
- lowercase 435
- margin 371, 445, 493, 740
- margin-bottom 445, 740
- margin-left 368, 439, 445, 473, 474, 740
- margin-right 439, 445, 740
- margin-top 370, 445, 500, 741
- marquee- 418
- max-height 483, 741
- max-width 483, 741
- medium 428
- min-height 483, 741
- min-width 483, 742
- move-to 416
- ms-filter 504
- narrower 432
- nav-down 742
- nav-index 742
- nav-left 742
- nav-right 743
- nav-up 743
- normal 428
- oblique 428
- opacity 408, 415, 502, 504, 743
- open-quote 513
- orphans 743
- outline 459, 460, 744
- outline- 413, 449
- outline-color 449, 460, 744
- outline-offset 449, 459
- outline-style 449, 460, 744
- outline-width 449, 460, 745
- overflow 231, 318, 370, 483, 484, 745
- overflow-style 418
- overflow-x 745
- overflow-y 746
- padding 366, 368, 369, 380, 401, 448, 463, 477, 746
- padding-bottom 448, 746
- padding-left 441, 448, 473, 474, 746, 747
- padding-right 448
- padding-top 448, 747
- page-break-after 747
- page-break-before 748
- page-break-inside 748
- position 371, 488, 748
- position:fixed 364
- quotes 416, 513, 749
- resize 749
- right 488, 749
- rotation 414
- ruby- 420
- semi-condensed 432
- semi-expanded 432
- size 385
- small-caps 428
- smaller 428
- src 385, 436
- table-layout 749
- target- 417
- text-align 403, 421, 439, 477, 750
 - center 364, 366
 - left 403
- text-align-last 750
- text-blink 371
- text-decoration 393, 434, 750
- text-decoration:line-through 365
- text-decoration:underline 365
- text-indent 441, 750
- text-justify 751
- text-outline 421
- text-shadow 421, 433
- text-transform 434, 435, 751
- top 373, 488, 751
- transition 422
- transition-delay 422
- transition-duration 422
- transition-property 422
- transition-timing-function 422
- ultra-condensed 432
- ultra-expanded 432
- unicode-bidi 751
- uppercase 435
- vertical-align 440, 477, 752
- visibility 494, 497, 752
- white-space 371, 421, 443, 752
- whitespace:nowrap 369
- wider 432
- widows 753
- width 147, 227, 233, 399, 446, 483, 493
- word-spacing 421, 432, 753
- x-large 428
- xx-large 428
- xx-small 428
- z-index 372, 374, 501, 753
- CSS-Elemente
 - caption-side 482
- CSS-Formate 386
- CSS-Maßeinheit
 - % 404
 - cm 404
 - em 404
 - ex 404
 - in 404
 - mm 404
 - pc 404
 - pt 404
 - px 404
- CSS-Methoden
 - attr() 511
 - counter() 516
 - hsl() 415

- hsla() 415
 - image() 417
 - rgb() 407, 415, 452
 - rgba() 408, 415
 - url() 417, 458, 465, 470, 510, 518
 - CSS-Präfix
 - moz 414
 - CSS-Pseudoelemente
 - :after 513, 516
 - :before 513, 514, 516
 - :first-letter 508
 - :first-line 508
 - CSS-Pseudoklassen
 - :after 416
 - :before 416
 - :first-child 508
 - :in-range 413
 - :invalid 413
 - :lang 513
 - :marker 418
 - :out-of-range 413
 - :valid 413
 - a:hover 366
 - a:link 368
 - a:visited 370
 - CSS-Regel
 - @charset 377
 - CSS-Regeln 380
 - !important 396
 - @charset 377, 381
 - @font-face 385, 436
 - @import 376, 380, 398
 - @keyframes 412
 - @media 380, 382, 518
 - @namespace 419
 - @page 380, 384
 - CSS-Reset 492
 - CSS-Schlüsselwörter
 - active 393
 - hover 393
 - link 393
 - visited 393
 - CSS-Schriftfamilien
 - cursive 426
 - fantasy 426
 - monospace 426
 - sans-serif 426
 - serif 426
 - CSS-Sprites 417
 - CSS-Stylesheets 82
 - Cue Components 221
 - Cue Settings 221
 - Cue-IDs 220
 - Cues 219
 - Cursor-Anzeige 516
- D**
- Dateiauswahldialog 266
 - Dateikonventionen 51
 - Dateinamen 52, 350
 - Dateinamenerweiterung 350
 - Datei-Uploads 264
 - Datenbank 310
 - Datenbankanbindung 26
 - Datenbanksoftware
 - MySQL 36
 - PostgreSQL 36
 - Datenfeld 326
 - Datenlisten 260
 - Datenzellen 141
 - Date-Objekt 565
 - Datumsangaben 279
 - Datums-Zeitnotation 74
 - DCMI 86
 - Deaktivieren 283
 - Deeplinks 159, 179
 - Definition
 - DTD 358
 - DELETE 247
 - Details aufklappbare 303
 - deutlicher 238
 - Dezimalzeichen 404
 - Dialogelemente 304
 - Diashow 420
 - Doctype 351
 - Doctype-Switch 65
 - Document Object Model 22, 25, 46, 550, 553, 565
 - Document Type Definition 513, 516
 - document-Objekt 567
 - Dokumentelement 360
 - Dokumentereignisse 534
 - Dokument-Grundgerüst 58
 - Dokumentkörper 58, 89
 - Dokumenttyp 58, 60
 - HTML 4.01 60
 - HTML5 60
 - XHTML 1.1 62
 - Dokumenttyp-Definition 61, 351, 355, 357
 - Dokumenttyp-Deklaration 351
 - DOM 22, 25, 46, 553, 565, 568
 - Eigenschaft 558
 - innerHTML 560
 - Schnittstelle 412
 - this 527
 - innerHTML 528
 - Versionen 568
 - DOM-Abbildung 46
 - DOM-Inspector 46
 - DOM-Knoten 46
 - assoziierter Knoten 554
 - Attributknoten 554
 - Ausgangsknoten 554
 - Elementknoten 554
 - Kindknoten 554
 - Textknoten 554
 - DOM-Scripting 137, 553
 - Elemente erzeugen 563
 - Dots per Inch 405
 - Downloads
 - EXE-Dateien 163
 - HQX-Archive 163
 - Download-Verweise 161
 - DPI-Dichte 239
 - Drag-and-Drop 338
 - DTD 61, 351, 356, 357
 - Dublin Core 325
 - Dublin-Core-Metadaten-Set 86
 - durchgestrichen 125, 434
- E**
- EBCDIC-Kodierung 63
 - Ecken 449, 477
 - ECMA-262 565
 - ECMAScript 537
 - Eigenschaften
 - document

- charset 567
- cookie 567
- URL 567
- style
 - left 562
 - position 562
 - top 562
- Eingabehilfen 275, 284, 337, 341
- Eingabekontrolle 275
- Eingebettete Frames 228
- Einzeilige Eingabefelder 253
- Einzug 438
- Element
 - <div> 523
- Elementbreite 483
- Elemente 43
 - <!doctype html> 136
 - <a> 118, 121, 159, 168, 170, 172, 174, 177, 331, 334, 335, 336, 355, 367, 393, 395, 400, 506, 513, 555, 560, 570, 609
 - <abbr> 44, 118, 121, 363, 610
 - <acronym> (veraltet) 363
 - <address> 99, 400, 611
 - <applet> 367
 - <applet> (obsolet) 363
 - <area> 186, 187, 369, 611
 - <article> 97, 320, 444, 612
 - <aside> 612
 - <audio> 209, 210, 212, 214, 217, 223, 226, 331, 613
 - 118, 121, 388, 389, 613
 - <base> 232, 614
 - <basefont> 367, 368, 370
 - <basefont> (obsolet) 364
 - <bdi> 118, 121
 - <bdo> 118, 121, 423, 614
 - <bgsound> (proprietär) 371
 - <big> (obsolet) 364
 - <blink> 434
 - <blink> (proprietär) 371
 - <blockquote> 110, 160, 381, 391, 451, 510, 615
 - <body> 59, 89, 90, 157, 366, 368, 370, 372, 373, 374, 387, 401, 405, 425, 444, 465, 466, 488, 492, 534, 536, 540, 562, 571, 616
 -
 44, 102, 367, 443, 470
 - <break> 616
 - <button> 270, 271, 272, 317, 617
 - <canvas> 188, 198, 200, 618
 - <caption> 148, 153, 481, 482, 618
 - <center> (obsolet) 364
 - <circle> 192, 194, 205
 - <cite> 118, 122, 619
 - <code> 113, 118, 122, 620
 - <col> 147, 620
 - <colgroup> 147, 621
 - <command> 300
 - <datalist> 261, 621
 - <dd> 109, 497, 622
 - <def> 196
 - 128, 622
 - <details> 303, 304, 623
 - <dfn> 118, 122, 624
 - <dialog> 304, 625
 - <dir> 367
 - <div> 115, 125, 233, 317, 318, 322, 323, 388, 389, 408, 444, 446, 458, 488, 492, 493, 499, 519, 522, 544, 562, 564, 571, 624
 - <dl> 108, 496, 626
 - <dt> 109, 497, 626
 - <ellipse> 194, 206
 - 118, 123, 387, 399, 554, 627
 - <embed> 223, 226, 363, 373, 627
 - <entry> 98
 - <fieldset> 251, 628
 - <figcaption> 111, 629
 - <figure> 111, 184, 629
 - 367, 368, 370
 - (obsolet) 364
 - <footer> 92, 492, 630
 - <form> 227, 246, 248, 249, 256, 257, 265, 271, 287, 288, 293, 306, 311, 318, 533, 558, 631
 - <frame> (obsolet) 364
 - <frameset> (obsolet) 364
 - <h[1-6]> 96, 631
 - <h1> 43, 90, 91, 323, 387, 388, 399, 406, 437, 448, 484, 506, 516, 517, 554, 558
 - <h2> 516
 - <head> 59, 71, 75, 322, 376, 377, 520, 524, 571, 572, 632
 - <header> 92, 401, 444, 492, 632
 - <hgroup> 96, 135, 633
 - <hr> 114, 369, 564, 633
 - <html> 59, 88, 192, 351, 362, 370, 400, 425, 634
 - <i> 119, 123, 388, 389, 634
 - <iframe> 178, 224, 226, 228, 230, 232, 233, 234, 368, 635
 - <ilayer> (proprietär) 371
 - 111, 182, 186, 188, 190, 209, 223, 224, 226, 240, 271, 321, 329, 330, 336, 354, 372, 373, 390, 498, 636
 - <input> 253, 254, 261, 263, 270, 275, 278, 286, 287, 292, 300, 301, 364, 370, 390, 513, 527, 532, 555, 558, 564, 637
 - <ins> 128, 637
 - <isindex> (obsolet) 364
 - <kbd> 113, 119, 123, 638
 - <keygen> 273, 639
 - <label> 251, 252, 639
 - <layer> (proprietär) 371
 - <legend> 251, 640
 - 104, 177, 300, 322, 323, 324, 333, 338, 387, 468, 474, 557, 641
 - <line> 194, 206
 - <linearGradient> 196
 - <link> 75, 77, 82, 172, 367, 376, 519, 520, 570, 641
 - <listing> (obsolet) 365
 - <main> 100, 642
 - <map> 186, 226, 642
 - <mark> 119, 123, 643
 - <marquee> 372, 418
 - <marquee> (proprietär) 371

- <math> 130
- <menu> 298, 300, 301, 338, 644
- <menuitem> 644
- <meta> 67, 322, 328, 333, 370, 524, 645
- <meter> 297, 645
- <mfenced> 131
- <mfrac> 132
- <mi> 131
- <mn> 131
- <mo> 131
- <mrow> 132
- <msqrt> 132
- <msup> 131
- <multicol> 373
- <multicol> (proprietär) 371
- <nav> 176, 177, 302, 465, 492, 513, 646
- <nobr> (proprietär) 371
- <noframes> (obsolet) 365
- <noscript> 134, 538, 646
- <object> 224, 225, 226, 363, 367, 368, 370, 647
- 104, 468, 470, 648
- <optgroup> 258, 260, 648
- <option> 258, 259, 260, 261, 262, 564, 648
- <output> 291, 649
- <p> 90, 101, 160, 168, 327, 328, 380, 387, 389, 444, 460, 528, 555, 649
- <param> 226, 370, 650
- <plaintext> (obsolet) 365
- <polygon> 194, 206
- <pre> 49, 111, 113, 444, 650
- <progress> 293, 295, 651
- <q> 119, 124, 513, 652
- <radialGradient> 196
- <rect> 193, 205
- <rp> 119, 124, 652
- <rt> 119, 124, 653
- <ruby> 119, 124, 653
- <s> 125, 654
- <s> (obsolet) 365
- <samp> 113, 120, 125, 655
- <script> 49, 83, 200, 204, 317, 355, 368, 537, 539, 540, 562, 571, 572, 656
- <section> 91, 92, 97, 656
- <select> 258, 259, 260, 438, 564, 657
- <small> 120, 125, 658
- <small> (obsolet) 365
- <source> 210, 211, 212, 213, 215, 216, 658
- <spacer> (proprietär) 371
- 120, 125, 323, 328, 356, 460, 530, 533, 558, 659
- <stop> 197
- <strike> (obsolet) 365
- 120, 126, 660
- <style> 83, 355, 376, 378, 401, 428, 436, 448, 451, 460, 477, 488, 491, 492, 503, 506, 508, 510, 511, 519, 560, 571, 660
- <sub> 120, 126, 661
- <summary> 303
- <sup> 120, 126, 661
- <svg> 191
- <table> 142, 366, 367, 368, 369, 452, 477, 479, 480, 481, 662
- <tbody> 145, 663
- <td> 142, 366, 390, 477, 481, 663
- <text> 198, 340
- <textarea> 255, 256, 340, 374, 530, 664
- <tfoot> 144, 664
- <th> 142, 144, 366, 477, 665
- <thead> 144, 665
- <time> 120, 126, 321, 334, 346, 666
- <title> 59, 67, 70, 666
- <tr> 142, 477, 666
- <track> 667
- <tt> (obsolet) 365
- <u> 120, 127, 667
- <u> (obsolet) 365
- 104, 160, 175, 177, 324, 470, 474, 668
- <var> 127, 669
- <video> 209, 213, 214, 217, 223, 226, 336, 669
- <wbr> 103
- <xmp> (obsolet) 365
- aside 94
- base 84
- doctype 59
- floatende 489
- iframe 234
- leere 352
- menuitem 299, 301
- nav 94
- positionierte 485
- svg 192
- track 218
- ul 94
- unbekannte 137
- Elementhöhe 483
- Elementknoten 554
- Ellipse 194
- eMail
 - bcc 166
 - body 166
 - cc 166
 - subject 166
- E-Mail-Links 164
- Embedded 136
- Embedded Content 136
- Empfehlungskandidat 40
- Empfehlungsvorschlag 40
- enctype
 - application/x-www-form-urlencoded 248
 - multipart/ form-data 248
 - text/plain 247, 248
- Entities 763
- Entity
 - & 57, 167, 234
 - > 57
 - ⁢ 132
 - < 57
 - 48, 143, 234, 481, 527
 - ± 132
 - " 57
- Ereignisbehandler 525
- Erläuterung 106

- Event-Handler 525, 569
- onabort 703
- onafterprint 703
- onbeforeunload 704
- onbeforeprint 704
- onblur 286, 532, 704
- oncanplay 704
- oncanplaythrough 704
- onchange 300, 532, 704
- onclick 270, 301, 317, 528, 560, 571, 572, 705
- oncontextmenu 705
- oncuechange 705
- ondblclick 705
- ondrag 705
- ondragend 338, 706
- ondragenter 706
- ondragleave 706
- ondragover 517, 706
- ondragstart 338, 706
- ondrop 707
- ondurationchange 707
- onemptied 707
- onended 707
- onerror 536, 707
- onfocus 286, 708
- onformchange 708
- onforminput 292, 708
- onhashchange 708
- oninput 708
- onkeydown 709
- onkeypress 531, 709
- onkeyup 530, 709
- onload 536, 540, 562, 572, 709
- onloadeddata 709
- onloadedmetadata 710
- onloadstart 710
- onmessage 710
- onmousedown 527, 710
- onmousemove 710
- onmouseout 527, 528, 711
- onmouseover 527, 528, 711
- onmouseup 527, 711
- onmousewheel 711
- onoffline 712
- ononline 712
- onpause 712
- onplay 712
- onplaying 712
- onprogress 712
- onratechange 712
- onreadystatechange 713
- onreset 713
- onresize 713
- onscroll 713
- onselect 713
- onstalled 713
- onsubmit 293, 318, 533, 714
- onsuspend 714
- ontimeupdate 714
- onunload 536, 714
- onvolumechange 714
- onwaiting 714
- Extensible Hypertext Markup Language 28
- Extensible Markup Language 27
- Extensible Stylesheet Language 29, 361
- external 173
- Externe Stylesheets kodieren 377
- F**
- Fallunterscheidung 546
- Farbdefinition
 - Farbnamen 409
 - hexadezimal 406
 - rgba-Format 407
 - rgb-Format 407
 - Systemfarben 408
- Farbnamen 405, 409
- Farbtiefe 240
- Farbverläufe 195
- Farbwerte 405, 433
- Favicon 80
- Feldeingaben 281
- Feldinterpretation 281
- Fensterereignisse 534
- Fettschrift 198
- Fieldset 250
- File Transfer Potokoll 34
- Firefox 17, 33
- fixe Positionierung 485
- fixes Layout 492
- FLAC 242
- Flash 27
- Flashmovies 223
- Flexible Erweiterbarkeit 30
- float 94
- Floatende Elemente 489
- Flow-Content 133, 134, 142
- Font-Management-Programm 426
- Footer 89, 91
- Formatdefinitionen 394
- Formate 394
- Formatvorlagen 393
- Formeln 129
- Form-Mailer 306
- Formular
 - HTML 306
- Formularbereich 246
- Formulardaten 246, 310
- Formulardatenübertragung 248
- Formulare 245
 - in Datenbank speichern 310
- Formularfelder
 - versteckte 267
- Formularfeldinhalte 246
- Formular-Feldtypen 252
- Formularkontrolle 288
- Formularverarbeitung 306
- Formularversand 246, 534
- Frames
 - eingebettete 178, 224, 228
- Frameset 229
- Free Lossless Audio Codec 242
- Frontend 26
- FTP 34
- FTP-Programme
 - Commander 35
 - FileZilla 34
 - PuTTY 34
 - SpeedCommander 35
- Füllung 195
- function circle 205
- function ellipse 205
- function line 206

- function polygon 206
- function rect 204
- function setFill 206
- function setLinearFill 207
- function setRadialFill 208
- function setStroke 207
- function text 208
- function unset 207
- Function-Objekt 565
- Funktionen
 - ABC() 557, 558
 - anonyme 549
 - bpKreis() 562
 - BruttoBetrag() 549
 - calcTime() 536
 - check_radio() 531
 - CoolLink() 560
 - errorInfo() 536
 - GEO() 557, 558
 - hasContent() 533
 - hideMessage() 533
 - Init() 562
 - mouse_pos() 527
 - prepareInput() 533
 - rpKreis() 562
 - saveTime() 536
 - setTimeout() 536
 - show_charts_entered() 530
 - show_mouse_pos() 527
 - showTime() 536
 - toggle_bgcolor() 528
- Funktionsbibliothek 201
- Funktionsrahmen 449, 459
- Fußbereich 91
- G**
- Gammakorrektur 241
- Gatterzeichen 392, 406
- Gecko 33
- Gecko Rendering Engine 65
- Generierte Vektorgrafik 198, 567
- generische Schriftfamilien 426
- Geolocation-Schnittstelle 567
- geordnete Listen 103, 470
- Gesamtbreite
 - eines Elements 402
- Gesamthöhe
 - eines Elements 402
- GET 280, 311
- GET-Methode 246
- GIF 181
- GIF-Format 238
- Gitternetzlinien 142
- Gleichung 131
- Gliederungsansicht 420
- globale Attribute 184, 319
- Globale Attribute 337
 - accesskey 337
 - class 337
 - contenteditable 338
 - contextmenu 338
 - dir 338
 - draggable 338
 - hidden 339
 - id 339
 - itemid 340
 - itemprop 340
 - itemref 340
 - itemscope 340
 - itemtype 340
 - lang 340
 - spellcheck 340
 - Sprachsteuerung 342
 - style 341
 - tabindex 341
 - title 341
- GMT 74
- Google Chrome 17, 33
- Grafik
 - Breite 182
 - Höhe 182
 - PNG 321
- Grafiken 181
 - Web-gerechte 181
- Grafikformate
 - gif 238
 - GIF 35
 - JPG 35
 - PNG 35
- Grafikprogramme 35
 - Adobe Illustrator 35
 - Adobe Photoshop 35
 - CorelDRAW 35
 - Gimp 35
 - Google Picasa 35
 - IrfanView 35
 - Ulead PhotoImpact 35
- Grafikreferenz 160
- grafische Rahmen 456
- grafische Submit-Buttons 271
- Graphics Interchange Format 238
- greater than 51
- greater than equal 51
- Greenwich Mean Time 74
- Greenwich-Zeit 332
- Grid 416
- Grid-Layouts 158
- Grouping 100
- Gruppierende Elemente 100
- gt 51
- gte 51
- H**
- H.264 243
- Haftungsausschluss 180
- hCalendar 345, 346
- hCard 328, 346
- header 116
- Header 89, 91
- Heading 135
- hexadezimale Farbangaben 405
- Hilfe 77
- Hintergrundbild 461
- Hintergrundfarbe 380, 461, 477
- Hintergrundgrafik 463
- History-Objekt 566
- Hochkommata 353
- Höhe
 - der Grafik 182
 - von Elementen 483
- Homepage 18
- HTML 21
 - Dokumente 72
 - Entity 48
 - Syntax 376
 - Version 1.0 21
 - Version 2.0 21
 - Version 3.2 22

Version 4.0 22
 Version 5 22
 HTML 4.01
 frameset 61
 strict 61
 transitional 61
 HTML-Attribut 558
 HTML-Dokument 43
 Titel 69
 HTML-Elementobjekte 558
 HTML-Entitäten 234
 HTML-Parser 46, 349
 HTML-Text
 drehen 421
 HTML-Wurzelement 351
 HTT
 Request 74
 http
 Header 72
 Kommunikation 33
 HTTP 53
 HTTP-POST-Methode 247
 HTTP-Protokoll 254
 HTTP-Request 174, 246
 HTTPS-Protokoll 254
 HTTP-Übertragungsmethode
 246
 Hyperlinks 159, 504

I

IA5STRING 274
 IANA-Verzeichnis 789
 iCalendar 331
 id-Attribut 391
 if-Bedingung 545
 Image-Maps 185
 Inhalte
 automatische 508
 Inhaltsbereich 100
 Inline-Element 498
 Inline-Elemente 133, 399, 494
 (HTML 4) 116
 Innenabstand 446
 Innenabstand zum Rand 380
 Integration 30
 in andere Software-
 Umgebungen 30

Interactive Content 136
 interaktive Elemente 531
 Interessensgruppen 39
 Interest Groups 39
 Interlaced 238
 Interlaced-Funktion 240
 Internet Explorer 16, 33
 Internet-Explorer-Problem 137
 interpretierte Bereiche 47, 49
 iso-8859-1 377
 Item 320

J

Java-Applet 223, 225, 366, 367
 Java-Archivdatei 224
 JavaScript 25, 245, 547
 alert 549
 Anweisungsblock 542
 Array 557, 560
 Ausdrücke 542
 Bit-Operatoren 552
 break 547
 case 547
 Date 550
 Eigenschaften 550
 firstChild 557
 for 548, 557, 558, 570
 Frameworks 26
 function 549
 Funktion 542
 Funktionen 548
 getElementById() 562
 getElementsByName() 555
 getElementsByName() 555, 557
 history 551
 if 542, 546, 548, 550, 552
 if-else-Konstrukt 546
 Instanzen 550
 Kommentare 543
 length 557
 logische Operatoren 552
 Math 205, 551
 Methoden 550
 new 550, 557
 Notation von Zahlen 553
 Objekte 550

Operatoren 552
 PI 551
 Rechenoperatoren 552
 reservierte Wörter 543
 return 557
 Schleifen 547, 548
 Selbstvergebene Namen 542
 Steuerzeichen 553
 switch 547
 test() 569
 Variablen 543
 globale 544
 initialisieren 545
 lokale 544
 Vergleichsoperatoren 552
 Werte 543
 while 547
 XMLHttpRequest 566
 Zahlen 553
 Zeichenketten 553
 Zeichenverknüpfungs-
 Operator 552
 Zuweisungsoperator 552
 Javascript Audio Interface 212
 JavaScript Object Notation 551
 Joint Photographic Expert
 Group 239
 jpeg 239
 JPEG 181
 jpg 239
 JSON 551

K

Kalenderwochenangaben 280
 Kaskadierung 394
 Kindknoten 360, 554
 Klartextformat 22
 Kleinschreibung 351, 352
 Knoten 46, 360, 554
 Pfad 360
 Knoten-Set 360
 Kodierung
 ISO 8859-1 63
 Latin-1 63
 UTF-8 64
 Kombinationsfelder 260
 Kommentar-Cue 220

- Kompatibilitätsmodus 66, 67
 - Kompressionsfaktor 239
 - Kontaktdaten 325
 - Kontaktinformationen 98
 - Kontext-Menü 301
 - Kontextmenüs 298
 - Kontrollstrukturen 545
 - Konturen 449
 - Kopfbereich 58, 91
 - Kopfdaten 69
 - Kopfzellen 141, 142
 - Kopfzellenüberschriften 154
 - Kreis 194
 - Kreuztabelle 154
 - Kursivschrift 123
- L**
- Label 250
 - Lauftext 418
 - Layer-Technik 500
 - Layout
 - mit fester Breite 492
 - leere Elemente 352
 - Leerraum 47, 112, 353
 - Leerzeichen 48
 - Lesezeichen 173
 - less than 51
 - less than equal 51
 - Linie 194
 - Linkhaftung 179
 - Links
 - mit Ping-Funktion 174
 - projektinterne 159
 - rechtliche Aspekte 179
 - sprechende 161
 - typisierte 171
 - zu E-Mail-Adressen 164
 - zu fremden Webseiten 159
 - Linktypen 76
 - für Hyperlinks 172
 - Listen
 - ungeordnete 104, 175
 - Listen verschachteln 105
 - Listenelemente 133
 - Listenformatierung 468
 - Living Standard 41
 - Lizensierung 336
- Lizenzangaben
 - maschinenlesbare 336
 - Location-Objekt 566
 - logische Verlinkung
 - Linktypen 76
 - logisches Und 552
 - Loslassziel 338
 - lt 51
 - lte 51
 - Lynx 33
- M**
- mailto 165
 - Markupstruktur 554
 - Maskierung 166
 - Maßeinheit
 - em 194
 - ex 194
 - für numerische CSS-
Angaben 404
 - gr 416
 - Inch 194
 - Millimeter 194
 - Pica 194
 - Pixel 194
 - Punkt 194
 - Zentimeter 194
 - Zoll 194
 - Mathematical Markup
 - Language 29
 - MathML 29, 129
 - Math-Objekt 565
 - Mausereignisse 525
 - media query 382
 - Media-Queries 518
 - max-device-width 522, 523
 - max-width 522
 - min-device-width 524
 - only screen 522, 523
 - orientation 524
 - Media-Type
 - handheld 519
 - screen 519
 - Medienabfrage 211
 - Medienausgabetypen 382
 - Medientyp
 - application 790
 - audio 790
 - example 790
 - image 790
 - message 790
 - model 790
 - multipart 790
 - text 790
 - video 790
 - mehrere Submit-Buttons 289
 - Mehrzeilige Eingabefelder 255
 - Menüs 298
 - Meta-Angaben 69
 - im HTML5-Standard 72
 - mit http-equiv 72
 - mit name 72
 - zur Zeichenkodierung 71
 - Metadaten 134
 - Meta-Tag 71
 - Methoden
 - addColorStop() 208
 - alert() 546
 - appendChild() 565
 - arc() 205
 - back() 551
 - bezierCurveTo() 206
 - circle() 205
 - createAttribute() 567
 - createElement() 567
 - createLinearGradient() 208
 - createTextNode() 564
 - Date() 550
 - delete() 552
 - ellipse() 206
 - execCommand() 318
 - fillRect() 205
 - GET 247
 - getDate() 550
 - getElementById 201, 339
 - getElementById() 555, 565, 567
 - getElementsByName() 567
 - getElementsByName() 567
 - getHours() 550
 - HTTP-Request 246
 - line() 206
 - pkt_minus() 270

pkt_plus() 270
 polygon() 206
 prompt() 546, 548
 reset() 558
 sort() 558
 submit() 558
 typeof() 552
 url() 381
 void() 552
 Metrische Werte 296
 microformats.org 344
 Mikrodaten 319, 370
 Anwendung 325
 Aufbau 320
 Referenzen 322
 Standardisierung 347
 Typisierung 323
 Mikroformate 344
 Mime-Type 162
 MIME-Type 80, 82, 170, 188,
 226, 248, 266
 application/xhtml+xml 349
 application/x-javascript 540
 application/xml 349
 application/x-shockwave-
 flash 223
 application/x-www-form-
 urlencoded 291
 audio/* 266
 audio/ogg 211
 audio/x-aiff 211
 audio/x-midi 211
 audio/x-wav 211
 der Zieldatei 163
 image/* 266
 image/gif 238
 image/jpeg 239
 image/png 240
 multipart/form-data 291
 text/css 378
 text/html 349
 text/javascript 538, 540
 text/plain 291
 text/xml 349
 Übersicht 790
 video/* 266
 video/mpeg 215

 video/quicktime 215
 video/x-msvideo 215
 MIME-Typenreferenz 789
 Monatsangaben 280
 -moz-background-clip 467
 MP3 242
 MP4 243
 MPEG-1 Audio Layer 3 242
 MPEG-4 243
 MS Internet Explorer 65
 MySQL 35, 310

N

Nachrichtentext 166
 Nähe aller Inhalte 179
 named entities 57
 Namensraum 351
 Namensraumangabe 351
 Namensraumdeklaration 190
 Name-Wert-Paar 259, 320
 Navigation 93, 175, 302
 Navigationsbereich 89
 Navigationsleiste 93
 Navigator-Objekt 566
 Netscape 25
 Netscape-Farbnamen 410
 Netscape-Farbpalette 410
 Newsletter-Abonnenten 290
 node 46
 nofollow 173
 norereferrer 174
 Noscript-Bereich 538
 NOTE 220
 NOTE-Cue 220
 Number-Objekt 565
 Nummerierung
 automatische 508
 Numerische Entities 57

O

Objekte
 all 567
 Array 565
 document 564, 567
 elements 567
 forms 567
 images 567

 JavaScript 565
 location 546
 Math 562
 window 562, 567, 572
 Objekteigenschaft 527
 Objekteigenschaften 551
 Obsolete Attribute 365
 Öffnen eines neuen
 Browserfensters 160
 OGG 242
 OGG / Vorbis 242
 OpenType 435
 Opera 17, 33, 65
 Oracle 310
 Outline-Struktur 135
 Outlining 420

P

Paged Media 424
 PCDATA 49
 PDF-Format 162
 Performance 30
 Permalinks 173
 Permanent-URL 87
 Pfadangaben
 absolute 54
 relative 55
 PHP 35, 306
 Phrasing Content 135, 289
 Phrasing-Content 101, 104,
 109, 297
 Phrasing-Context 295, 538
 Phrasing-Elemente 494
 Pingback 81
 Pingback-Nachrichten 82
 Pingback-Server 81
 Pixelgrafik 181
 Pixelkoordinaten 187
 Platzhaltertexte 284
 Plugin 164
 png 240
 PNG 181
 PNG-Format 240
 Polygon 194
 Popup-Menüs 298
 Portable Network Graphic 240
 Portnummer 54

- positionierte Elemente 485
 - Positionierung
 - absolute 485
 - fixe 485
 - relative 485
 - Positionierung absolute 560
 - POST 308, 311
 - PostgreSQL 310
 - POST-Requests 174
 - PostScript 64
 - präformatierte Bereiche 112
 - mit Leerraum 112
 - Pragma-Direktive 72, 74
 - Präzisionsfunktionen 30
 - Prefetching 171
 - Presto 33
 - Property 320
 - Proposed Recommendation 40
 - Proprietäre Attribute 372
 - Proprietäre Elemente 371
 - Proprietäres HTML 370
 - Prototyp-Objekt 565
 - Prozentkodierung 248
 - Pseudoelement
 - :after 510
 - :before 510
 - Pseudo-Elemente 392
 - Pseudoklassen
 - :active 506
 - :focus 506
 - :hover 506
 - :link 506
 - :visited 506
 - Pseudo-Klassen 392
 - für Hyperlinks 504
 - Punktlichte 239
 - PURL 87
 - PUT 247
- Q**
- qualified names 362
 - qualifizierte Namen 362
 - Quirks-Modus 65
 - Quotation-Mark 57
- R**
- Radiobuttons 262, 531
 - Rahmen 142, 380, 449, 477
 - Rahmenart 449
 - Rahmendicke 449, 452
 - Rahmenfarbe 449, 452
 - Rahmenoptionen 477
 - Rahmentyp 452
 - RCDATA 49
 - RDF 29, 85, 343, 347, 370
 - RDF/XML 86
 - RDFa 87
 - Really Simple Syndication 29
 - Rechteck 193
 - Recommendations 39
 - Referenzieren 52
 - Referenzierung 52
 - RegExp-Objekt 566
 - regulärer Ausdruck 278
 - relative Pfadangaben 55
 - relative Positionierung 485
 - Reset-Buttons 269
 - Resource Description Format 85
 - Resource Description Framework 29
 - Review-Prozesse 40
 - RFC
 - 1034 (Domain Names) 280
 - 1630 (URIs) 54
 - 2045 (MIME/Part One) 790
 - 2046 (MIME/Part Two) 790
 - 2313 (RSA) 274
 - 2426 (vCard) 326
 - 2445 (iCalendar) 332
 - 3279 (X.509) 274
 - 3986 (Syntax von URIs) 54
 - 4281 (Codecs) 211, 215
 - 4288 (Media Specs & Registration) 790
 - 4289 (MIME/Part Four) 790
 - 4646 (Language Tags) 795
 - 5322 (Internet Message Format) 280
 - RGB 276, 280
 - RGBa-Format 407
 - RGB-Format 407
 - Richtext 256
 - Richtext-Eingabefeld 315
 - Richtext-Eingaben 315
 - RSA Signature Algorithm 274
 - RSS 29, 75
 - Ruby-Text 420
 - Rückwärtskompatibilität 45
- S**
- Safari 17, 33
 - Same-Origin-Policy 232
 - Sandboxing 232
 - Sans Serif 425
 - Scalable Vector Graphics 28, 189
 - Schaltflächen 268
 - Schatteneffekt 432
 - Schichtposition 500
 - Schleifen 547
 - Schlüsselgenerierung 273
 - Schlüsselwörter 70
 - Schmuckrahmen 449
 - Schrägstrich 43
 - Schreibrichtung 65, 288
 - Schriftart 64, 198, 380, 425
 - Schriftartendatei 435
 - Schriftattribute 434
 - Schriftfarbe 380, 432
 - Schriftformatierung 425
 - Schriftgewicht 427, 434
 - Schriftgröße 198, 380, 404, 427
 - Schriftlaufweite 430
 - Schriftneigung 434
 - Schriftschatten 432
 - Schriftstil 427
 - Schriftvariante 427
 - Screen-Objekt 566
 - Script-Bereich 83, 527, 537
 - Script-Einbindung 82
 - Scriptsprache
 - Perl 36
 - PHP 36
 - Python 36
 - Scroll-Verhalten 231
 - Sectioning 89, 135
 - Secure File Transfer Potokoll 34
 - Seitenleiste 93
 - Selektor 386

- Gewichtung 396
- Nachfahren 388
- Semantische Auszeichnung 356
- semantisches Web 85
- SEO 51, 70
- Server-seitiges Scripting 26
- SFTP 34
- SGML 21, 23, 60, 356, 358
- sichere Datenübertragung 273
- Sichtbarkeit von Inhalten 494
- Signed Public Key and Challenge 273
- Slideshow 420
- SMIL 27
- soft hyphen 102
- Spalten 146
 - vordefinieren 147
- Spaltengruppen 146
- Spam 165
- Spam-Filter 165
- Spezifität 396
- Spinner-Controls 276
- SPKAC 273
- Sprachausgabesystem 152
- Sprachkürzel 795
- Sprachsteuerung 342
- Sprachsynthesizer 420
- SSH-Client 34
- Standalone-Tag 44
- Standardeingabekanal 247
- Standards-Modus 65
- Steuerzeichen in Zeichenketten 553
- strike-through 125
- String-Objekt 566
- Style-Bereich 83
 - im HTML-Dokumentkopf 377
- Style-Definitionen
 - in separate CSS-Dateien auslagern 375
- Style-Einbindung 82
- Stylesheet 82
- Submit-Buttons 268
 - mehrere 289
- Suchformulare 246
- Suchmaschinen 69
 - Suchmaschinenoptimierung 70
 - Suchmaschinen-Robot 173
 - SVG 27, 28, 182, 188
 - in HTML5 191
 - in XHTML 192
 - SVG-Grafik 225
 - Syntax-Highlighting 30
 - Szenenwechsel 218
 - szenische Dialoge 109
- T**
- Tabbed Interface 339
- Tabelle
 - Titelbereich 148
 - Zusammenfassung 152
- Tabellen 141
 - Alternative Überschrift 154
 - Backgrounds 157
 - expliziter Kopfzellenbezug 155
 - Kopfzellen 154
- Tabellenbeschriftung 148, 481
 - ausrichten 481
- Tabellenformatierung 474
- Tabellenfuß 141
- Tabellenkopf 141, 144
- Tabellenkörper 141
- Tabellenlayouts 157
- Tabellenzeile 142
- Tabellenzellen 133
- Tabellenzellen verbinden 149
- Tabulator 48, 286
- Tag 43
 - abschließendes 43
 - einleitendes 43
- Tastaturereignisse 529
- Tastaturkürzel 284
- Telefonnummern 281
- Text drehen 198
- Textabsatz 90, 101
- textarea 49
- Textauszeichnungen 116
- Text-Browser 33
- Textdateien 164
- Texteinrückung 440
- Textknoten 554
- Textkontrolle 421
 - textlevel semantics 116
 - Textlevel-Semantics 101, 104, 116
 - Textrichtungsänderung 118
 - Textrichtungswechsel 423
 - Textumbruchkontrolle 421, 442
 - Theora 243
 - Titel 69
 - Toolbar-Menüs 298
 - Tools
 - HTMLValidator 36
 - HTML Tidy 36
 - XenuLink 36
 - Tooltip 49, 184, 342
 - Tracks 217
 - Transformationskomponente 29
 - Transparenz 238, 240, 500, 502
 - Trennlinien 114
 - Trident 33
 - TrueType 64, 435
 - TV-Geräte 424
 - type
 - application/atom+xml 79
 - application/rss+xml 79
 - checkbox 263, 300
 - color 276
 - command 300
 - date 276, 278
 - datetime 276
 - datetime-local 276
 - email 276
 - hidden 317
 - image 271
 - image/vnd.microsoft.icon 80
 - image/x-icon 80
 - month 276
 - number 276
 - password 275, 390
 - pattern 276
 - radio 300
 - range 276, 278
 - reset 270, 271
 - search 276
 - submit 270, 271, 272
 - tel 276

text 278
 text/css 82, 83
 text/javascript 83, 538
 time 276
 url 276
 week 276

U

Überschrift 90, 95
 Überschriftenebene 95
 Überschriftenstruktur 135
 Uhrzeit 129
 Uhrzeitangaben 279
 Umriss 195
 unbekannte Attribute 344
 underlined 127
 ungeordnete Listen 104, 468
 Unicode 56
 Uniform Resource Locator 53
 Uniform Resource Names 53
 Universal Resource Identifier 53
 Universaleditoren
 Linux
 Bluefish 32
 gEdit 32
 Kate 32
 Quanta 32
 SciTE 32
 Screem 32
 Vi 32
 WebMaker 32
 Mac OS X
 BBEdit 32
 Bluefish 32
 Coda 32
 Espresso 32
 Pagespinner 32
 skEdit 32
 Smultron 32
 SubEthaEdit 32
 TextMate 32
 TextWrangler 32
 Plattformunabhängig
 Eclipse PDT 33
 Emacs 33

jEdit 33
 Vim 33
 Windows
 1stPage 31
 40tudeHTML 31
 Boxer 31
 CoffeeCup 31
 EditPlus 31
 EmEditor 31
 Homesite 31
 HTML-Kit 31
 Notepad++ 31
 NoteTab 31
 SciTE 31
 Scribe! 31
 SuperHTML 31
 Text Hawk 31
 TextPad 31
 TSW WebCoder 31
 UltraEdit 31
 Weaverslave 31
 Webocton Scriptly 31
 Zeus 32
 Universalselektor 387, 388, 492
 Unobtrusive Programming 528, 569
 unterstrichen 127, 434
 Untertitel 218
 URI 53
 URL 53
 URL-Adressbasis 84
 URN 53
 Usability 275
 User-Tracking 174
 UTC-Zeit 332
 utf-8 377
 UTF-8 381
V
 valide 60
 Validierung 357
 Variablen 121
 vCal 325
 vCalendar 331
 vCard 319, 325
 Vektorgrafik 182, 188

Veraltetes HTML 363
 Vererbung 394
 in CSS 394
 Verlinkung
 logische 75
 Verlustfreie Kompression 240
 verschachtelte HTML-Elemente 387
 formatieren 387
 Verschachtelung 44, 91
 von Elementen 44
 von Listen 105
 Verschachtelungsregeln 355
 Versteckte Formularfelder 267
 Verweise zu Ankern 354
 verweissensitive Grafiken 185, 224
 Verweisziel 159
 Verzeichnisstruktur 51
 vEvent 319, 331, 333
 VGA-Farben 409
 Video-Codecs 241
 Video-Untertitel 217
 Vieleck 194
 Viewport 400, 488, 493
 VML (Vektorgrafikformat von Microsoft) 188
 Vollbild 234
 Vorbis 242
 Vordefinieren
 E-Mail-Betreff 165
 E-Mail-Inhalt 165
 VPN 34

W

W3C-HTML 41
 W3-Konsortium 39, 41
 Wallpaper 465
 Web Hypertext Application Technology Working Group 41
 Web Open Font Format 437
 Web-Auftritt 18
 Web-Browser 33
 Web-Fonts 435
 Webkit 33
 Webkit Rendering Engine 65

- Weblog 19
 - Web-Plattform 20
 - Web-Präsenz 18
 - Webseiten-Bereiche 89
 - Webseitenlayout 421
 - Webserver 35
 - Apache 36
 - inklusive PHP und MySQL 36
 - LightHTTP 36
 - Xitami 36
 - Web-Shop 19
 - Website-Typologie 18
 - Web-Sockets-Schnittstelle 567
 - Web-Storage-Schnittstelle 567
 - WebVTT-Format 219
 - Weiterleitung
 - automatische 75
 - Wertzuzuweisung 540
 - WHATWG 41, 87, 136, 319, 336
 - WHATWG-HTML 41
 - Whitespace 47, 112
 - Whitespace-Zeichen 48
 - Wiki
 - DokuWiki 38
 - MediaWiki 38
 - Wikidot 38
 - Window-Objekt 566
 - Wochentagnamen 74
 - WOFF 437
 - word break 103
 - Working Drafts 40
 - Working Groups 39
 - Works-Anwendung 336
 - Wortabstand 430
 - Wrapper-Bereich 492
 - Wurzelknoten 360
 - WYSIWYG 37, 318
 - WYSIWYG-Werkzeuge 17
- X**
- XBL 414
 - XHTML 21, 22, 28, 59, 349
 - Dateninsel 362
 - Namensraum 351
 - Version 1.0 23
 - Version 1.1 23
 - Version 2.0 23
 - Version 5 23
 - XHTML 1.0
 - frameset 62
 - strict 62
 - transitional 62
 - XHTML 1.1 62
 - XHTML-Dokumenttyp 62
 - XMDP 328
 - XMDP-Profil 328
 - XML 23, 27, 59, 192, 356, 419, 498, 553
 - Deklaration 59, 350
 - Namensraum 362
 - Schema 358
 - Verarbeitungsanweisung 376
 - Version 59
 - XML Binding Language 414
 - XML Path 29
 - XML-Deklaration 350
 - XMLHttpRequest 25
 - XMLHTTP-Schnittstelle 245
 - XML-Namensräume 361
 - XML-Namensraumkonzept 190
 - XML-Parser 349
 - XPath 29
 - XSD 358
 - XSL 360, 361
 - XSLT 23, 29, 361
 - Ergebnisbaum 361
 - Quellbaum 361
- Z**
- Zeichenabstand 380, 430
 - Zeichenkodierung 30, 56, 246, 249, 256, 350, 377, 539
 - UTF-8 248
 - Zeichenkodierung für externe Stylesheets 381
 - Zeichenkodierungen 63
 - Zeichensätze
 - ISO-8859-1- 757
 - ISO-8859-2 763
 - ISO-8859-3 763
 - ISO-8859-4 763
 - ISO-8859-9 763
 - Latin Extended A 763
 - Latin Extended B 763
 - Latin-1 757
 - Zeichenvorrat 63
 - Zeigerfunktion 180
 - Zeilenhöhe 404, 430, 438, 441
 - Zeilenkontrolle 30
 - Zeilenumbruch 102
 - Zeitzoneangabe 129
 - zentrierte Anordnung 440
 - Zielfenster 249
 - ZIP-Datei 162
 - Zueigenmachen fremder Inhalte 180
 - Zugriff
 - auf CSS-Eigenschaften eines HTML-Elements 560
 - Zusatzbemerkung 120

Stefan Münz / Clemens Gull

HTML5 Handbuch



HTML5 ist die Sprache des Web. Und das HTML5-Handbuch von Stefan Münz und Clemens Gull ist die Grammatik dazu. Wie einst die von Stefan Münz mitbegründete Onlinereferenz SELFHTML beantwortet dieses Buch alle Fragen zum neuen Internetstandard. Immer wichtiger wird dabei die Ausgabe auf Smartphones und Tablets, deshalb ist diesem Thema hier ein eigenes Kapitel gewidmet. Auch das nicht mehr wegzudenkende jQuery wird behandelt. Das unverzichtbare Standardwerk für jeden Webprofil!

Die Systematik des neuen HTML

Mit HTML5 ist so viel möglich wie nie zuvor, aber es dient weiterhin im Wesentlichen dazu, Webseiteninhalte zu strukturieren. Anhand konkreter Beispiele führen die Autoren Stefan Münz und Clemens Gull die Leser detailliert an die Systematik der neuen Websprache heran – nichts bleibt außen vor.

Aktive Inhalte einbinden

Dieses Buch zeigt, wie Sie vektorgrafische und Multimedia-Elemente ganz einfach in HTML5 umsetzen und so Ihrer Kreativität freien Lauf lassen. Darüber hinaus erfahren Sie, wie Sie Imagemaps umsetzen und aktive Inhalte in Ihre Seiten integrieren.

Das optimale Design für jedes Gerät mit CSS3

Breiten Raum nimmt auch die Gestaltung mithilfe von Cascading Style Sheets (CSS) ein. Anhand vieler praktischer Beispiele erfahren Sie, wie Sie CSS3 optimal einsetzen und Ihre Inhalte durch Media Queries dem verwendeten Endgerät anpassen.

Alle wichtigen Infos stets zur Hand

Fester Bestandteil dieses Buchs ist ein umfangreicher Referenzteil, in dem Sie alle HTML5-Elemente, -Attribute und -Events sowie CSS-Eigenschaften finden. Darüber hinaus enthält der Referenzteil eine Zeichenreferenz sowie eine Übersicht über die im Internet verwendeten Sprachenkürzel.

Aus dem Inhalt:

- Webseitenerstellung heute
- HTML und XHTML
- Dateikonventionen und Referenzierung
- Zeichenkodierung in HTML5
- Quirks- und Kompatibilitätsmodi der Browser
- Textstrukturierung
- Kopf- und Fußleisten, Navigation, Seitenleisten
- Textauszeichnungen
- Mathematische Formeln in HTML5
- Tabellen in HTML5
- Hyperlinks: Aufbau, Anker, Ping-Funktion, Deeplinks
- Grafik und Multimedia
- Pixel- und Vektorgrafik
- Das neue canvas-Element
- Eingebettete Audio- und Videoressourcen
- JavaScript-Framework jQuery

Über die Autoren:

Stefan Münz ist der geistige Vater und ursprüngliche Autor der bekannten Onlinedokumentation SELFHTML.



Clemens Gull ist als Dozent für die Fachhochschule Salzburg aktiv. Sein Weblog „Guru 2.0“ (www.guru-20.info) gehört zu den meistgelesenen deutschsprachigen Blogs zum Thema Internetprogrammierung.



Alle Codebeispiele dieses Buchs finden Sie unter:

<http://webkompetenz.wikidot.com/docs:html-handbuch>



9 783645 603454

Besuchen Sie unsere Website
www.franzis.de

FRANZIS