

Heather Williamson

Dynamic HTML browserübergreifend

Galileo Computing

Inhalt

Danksagungen 13

Vorwort 14

Aufbau dieses Buches 15

Was Sie in diesem Buch nicht finden werden 16

Zukünftige Entwicklungen 17

4

Rückmeldung gefällig? 17

Die Autorin 17

- 1 Dynamic HTML: Ein Überblick 19
 - 1.1 Was ist DHTML? 19
 - 1.1.1 Das Document Object Model 20
 - 1.1.2 Cascading Style Sheets 20
 - 1.2 Gründe für Dynamic HTML 21
 - 1.2.1 DHTML-Lösungen 21
 - 1.2.2 Plug-In-Lösungen 22
 - 1.2.3 Serverbasierte Lösungen 22
 - 1.3 Implementierung von Dynamic HTML durch Microsoft 22
 - 1.3.1 Verwenden der Microsoft-Programme 23
 - 1.3.2 Befehle von Microsoft 23
 - 1.3.3 Microsoft und das DOM 24
 - 1.3.4 Microsoft und Cascading Style Sheets 25
 - 1.4 Implementierung von Dynamic HTML durch Netscape 25
 - 1.4.1 Verwenden der Netscape-Programme 25
 - 1.4.2 Befehle von Netscape 25
 - 1.4.3 Netscape und das DOM 26
 - 1.4.4 Netscape und Cascading Style Sheets 26
 - 1.5 Gemeinsamkeiten der beiden Browser 26

2	Überblick über das Document Object Model	29
2.1	Was genau ist das DOM?	29
2.2	DOM-basierte Dokumentstruktur	31
2.3	Die Implementierung des DOM durch Microsoft	35
2.3.1	Das Ereignismodell von Microsoft	35
2.3.2	Echtzeitaktualisierungen von Text bei Microsoft	36
2.4	Die Implementierung des DOM durch Netscape	37
2.4.1	Wachstum bei Netscape	37
2.4.2	Unterstützte Netscape-Objekte	37
2.5	Gemeinsamkeiten der beiden Browser	38
2.5.1	Ein kompatibles Ereignismodell	38
2.5.2	Kompatible Objekte und Eigenschaften	39
3	Implementieren von Cascading Style Sheets	41
3.1	Implementierung der Cascading Style Sheets durch Microsoft	42
3.1.1	OpenType-Schriftartanpassung	43
3.1.2	Textdekorationen und andere Effekte	44
3.1.3	Seitenränder, Umrandungen und andere Rahmenmodelleigenschaften	46
3.1.4	Layer und Positionierung	50
3.2	Filter	53
3.3	Implementierung der Cascading Style Sheets durch Netscape	55
3.3.1	TrueDoc-Schriftartanpassung	55
3.3.2	Textdekorationen	58
3.3.3	Seitenränder, Umrandungen und andere Container-Eigenschaften	59
3.4	Layer und Positionierung	59
3.5	Gemeinsamkeiten der beiden Browser	60
	 Skriptsprachen	63
4.1	Ein schneller Blick auf Java	63
4.1.1	Die JavaScript-Familie	64
4.1.2	Der ECMAScript-Standard	65
4.1.3	Die Microsoft-Implementierung: JScript	68
4.1.4	Die Netscape-Implementierung: JavaScript	70
4.2	Gemeinsamkeiten der beiden Browser	70

5	Einrichten der Benutzerumgebung	75
5.1	Erkennen der Umgebung	75
5.2	Erkennen der Auflösung	80
6	Vorbereiten auf den Browserkrieg	89
6.1	Erstellen einer API mit Hilfe des DOM	90
6.2	Einrichten interner Seitenabzweigungen	91
6.3	Erstellen von Stylesheets	92
6.4	Erstellen von Grafiken	95
6.5	Erkennen von Layern	97
4		
7	Vorbereiten auf die Interaktion	103
7.1	Einrichten von Variablen	103
7.2	Vorausladen von Bildern	105
7.3	Erfassen von ImageMap-Ereignissen	107
7.4	Verwenden von Formularen	112
8	Implementieren von Interaktionen	121
8.1	Verwenden von Mausereignissen	121
8.1.1	Mausbewegungen	121
8.1.2	Mausclicks	130
8.1.3	Ändern der Seitengröße	146
8.2	Verwenden von Tastaturreignissen	147
8.2.1	Verwenden von Ereignissen innerhalb von Formularen	154
8.3	Verwenden von Ladezeiten	158
8.3.1	Laden eines Objekts	159
8.3.2	Laden einer Seite	160
8.4	Verwenden von Entladezeiten	161
8.5	Fehler	163
9	Arbeiten mit mehreren Layern	167
9.1	Anordnen und Indizieren von Layern	167
9.1.1	Relative Positionierung	167
9.1.2	Absolute Positionierung	168
9.1.3	Feste Positionierung	169

9.2	CSS (Cascading Style Sheet)-Eigenschaften	171
9.2.1	position	172
9.2.2	left	173
9.2.3	top	173
9.2.4	right	174
9.2.5	bottom	174
9.2.6	width	174
9.2.7	height	175
9.2.8	z-index	176
9.3	Layer-Code	176
9.4	Steuern der Sichtbarkeit von Layern	184
9.5	Erstellen Dynamischer Layer	186
9.5.1	Animieren von Layern	187
9.5.2	Zuweisen von Verhalten zu Layern	197
9.5.3	Ändern von Layer-Inhalten	208
9.6	Und so sieht die Site jetzt aus	212
10	Die WebRavin-Website	221
10.1	Die Startseite: index.htm	221
10.2	Das Hauptdokument: webravinmain.htm	223
10.3	Die wichtigsten Skripte: webravin.js	229
10.4	Scrollen von Layer-Inhalten: wrscrolling.js	236
10.5	Verwenden weiterer Dokumente: refs.htm	237
11	Entwicklung Ihrer Website	239
11.1	Aufgabenverteilung in einem Entwicklerteam	239
11.1.1	Projektmanager	240
11.1.2	Entwickler	241
11.1.3	Grafiker	241
11.1.4	Webserver-Administrator	242
11.1.5	Tester	242
11.2	Grundregeln für ein erfolgreiches Projektmanagement	242
11.3	Zusammenstellen der Projektbeschreibung	249
11.4	Fragen zum Schema	250
11.5	Grundlagen des Designs: Entwickeln des Schemas	251

12	Planen einer Site aus Sicht des Entwicklers	259
12.1	Entwickeln des Storyboards: ein fortschreitender Prozess	259
12.2	Entscheiden über die Navigationsmöglichkeiten	262
12.3	Erstellen von Grafiken	266
12.4	Planen der Interaktivität	267
12.5	Festlegen von Layern	269
12.6	Planen der Testphase	270
13	Erweitern der Gemeinschaft durch neue Zugangsmöglichkeiten	273
13.1	Richtlinien zur Erstellung von Websites mit optimierten Zugangsmöglichkeiten	274
13.2	Verwenden der Tags alt, title und longdesc	276
13.3	Akustische Stylesheets und Sprachsynthesizer	278
13.4	UNICODE und Internationalisierung	281
14	Abschließende Tipps, Tricks und Anmerkungen	289
14.1	So bekommen Sie Lob von Ihren Besuchern	289
14.1.1	Denken Sie bitte an Etiketete	290
14.1.2	Einrichten von eCommerce	293
14.1.3	Aufzeichnen und Lesen von Cookies	294
14.1.4	Drucken, aber richtig	295
14.2	Wie Sie die Aufmerksamkeit der Besucher auf sich ziehen	296
14.2.1	Schreiben Sie für den Markt	296
14.2.2	Halten Sie sich aus Frames raus	298
14.2.3	Umgehen der Schaltfläche »Übertragen«	298
14.2.4	Verwenden mehrerer Textspalten	299
14.3	Ein Blick in die Zukunft	300
14.3.1	Verbesserungen bei HTML	301
14.3.2	Cascading Style Sheets	301
14.3.3	Das Document Object Model	301
14.3.4	Entwicklungen im Bereich Extensible Markup Language	301

Anhang 303

- A Unterstützung einzelner Elemente durch die aktuellen Browser 303
 - B Kompatibilität mit Cascading Style Sheets 319
 - C Kompatibilität mit dem Document Object Model 343
 - D Kompatibilität mit JavaScript und JScript 369
- Index 385