

Inhalt

Vorwort	21
1 Einführung	23
1.1 Kleines Einstiegsprojekt zum Kennenlernen	23
1.1.1 Ein erstes XML-Dokument	23
1.1.2 Standardausgabe im Webbrowser	24
1.1.3 Wohlgeformtheit ist ein Muss	25
1.1.4 Gültige Dokumente per DTD oder Schema	26
1.1.5 Formatierte Datenausgabe	28
1.2 XML – universale Metasprache und Datenaustauschformat	29
1.2.1 Unabhängigkeit von Anwendungen und Plattformen	30
1.2.2 SGML → HTML → XML	30
1.2.3 Lob des Einfachen	31
1.2.4 Inhaltsbeschreibungssprache	31
1.2.5 Trennung von Inhalt und Form	32
1.2.6 Vom Dokumentformat zum allgemeinen Datenformat	32
1.2.7 Globale Sprache für den Datenaustausch	33
1.2.8 Interoperabilität	34
1.3 Übersicht über die Sprachfamilie XML	34
1.3.1 Kernspezifikationen	35
1.3.2 Ergänzende Spezifikationen	36
1.3.3 Programmierschnittstellen	36
1.3.4 XML-Anwendungen	37
1.4 XML-Editoren und Entwicklungsumgebungen	37
1.4.1 Editoren für XML	37
1.4.2 Schema- und Stylesheet-Designer	39
1.4.3 Entwicklungsumgebungen mit XML-Unterstützung	41
1.4.4 XML-Dokumente über Standardanwendungen	41
1.4.5 Parser und andere Prozessoren	42
1.5 Anwendungsbereiche	43
1.5.1 XML-Vokabulare	43
1.5.2 Datenaustausch zwischen Anwendungen	45
1.5.3 Verteilte Anwendungen und Webdienste	46

2.1	Aufbau eines XML-Dokuments	47
2.1.1	Entitäten und Informationseinheiten	47
2.1.2	Parsed und unparsed	48
2.1.3	Die logische Sicht auf die Daten	49
2.1.4	Der Prolog	51
2.1.5	Zeichenkodierung	51
2.1.6	Standalone or not	53
2.1.7	XML-Daten – der Baum der Elemente	53
2.1.8	Start-Tags und End-Tags	54
2.1.9	Elementtypen und ihre Namen	55
2.1.10	Regeln für die Namensgebung	56
2.1.11	Elementinhalt	57
2.1.12	Korrekte Schachtelung	58
2.1.13	Attribute	59
2.2	Die Regeln der Wohlgeformtheit	60
2.3	Elemente oder Attribute?	60
2.4	Reservierte Attribute	61
2.4.1	Sprachidentifikation	61
2.4.2	Leerraumbehandlung	62
2.5	Entitäten und Verweise darauf	62
2.5.1	Eingebaute und eigene Entitäten	62
2.5.2	Zeichenentitäten	63
2.6	CDATA-Sections	64
2.7	Kommentare	64
2.8	Verarbeitungsanweisungen	65
2.9	Namensräume	66
2.9.1	Das Problem der Mehrdeutigkeit	66
2.9.2	Eindeutigkeit durch URIs	66
2.9.3	Namensraumname und Präfix	67
2.9.4	Namensraumdeklaration und QNamen	67
2.9.5	Einsatz mehrerer Namensräume	68
2.10	XML Version 1.1	70

3.1	Metasprache und Markup-Vokabulare	71
3.1.1	Datenmodelle	71
3.1.2	Selbstbeschreibende Daten und Lesbarkeit	72
3.1.3	Dokumenttyp-Definition – DTD	72
3.1.4	XML-Schema	73
3.1.5	Vokabulare	73
3.2	Regeln der Gültigkeit	73
3.3	DTD oder Schema?	74
3.4	Definition eines Dokumentmodells	75
3.4.1	Interne DTD	75
3.4.2	Externe DTD	76
3.5	Deklarationen für gültige Komponenten	77
3.5.1	Vokabular und Grammatik der Informationseinheiten	78
3.5.2	Syntax der Dokumenttyp-Deklaration	78
3.5.3	Syntax der Elementtyp-Deklaration	78
3.5.4	Beispiel einer DTD für ein Kursprogramm	79
3.5.5	Inhaltsalternativen	81
3.5.6	Uneingeschränkte Inhaltsmodelle	82
3.5.7	Gemischter Inhalt	83
3.5.8	Inhaltsmodell und Reihenfolge	84
3.5.9	Kommentare	84
3.5.10	Die Hierarchie der Elemente	85
3.6	Dokumentinstanz	85
3.7	Attributlisten-Deklaration	86
3.7.1	Aufbau einer Attributliste	87
3.7.2	Attributtypen und Vorgaberegelungen	88
3.7.3	Verwendung der Attributlisten	89
3.8	Verweis auf andere Elemente	90
3.9	Verwendung von Entitäten	91
3.9.1	Interne Entitäten	92
3.9.2	Externe Entitäten	92
3.9.3	Notationen und ungeparste Entitäten	94
3.9.4	Verwendung von Parameterentitäten	95
3.9.5	Interne Parameterentitäten	95
3.9.6	Externe Parameterentitäten	95

3.10 Formen der DTD-Deklaration	96
3.10.1 Öffentliche und private DTDs	96
3.10.2 Kombination von externen und internen DTDs	97
3.10.3 Bedingte Abschnitte in externen DTDs	97
3.11 Zwei DTDs in der Praxis	98
3.11.1 Die Auszeichnungssprache DocBook	99
3.11.2 Das grafische Format SVG	101

4 Inhaltsmodelle mit XML-Schema 107

4.1 XML-Schema – der XML-basierte Standard	107
4.1.1 Defizite von DTDs	107
4.1.2 Anforderungen an XML-Schema	108
4.1.3 Die Spezifikation des W3C für XML-Schema	108
4.2 Erster Entwurf eines Schemas	109
4.2.1 Verknüpfung von Schema und Dokument	112
4.2.2 Der Baum der Schema-Elemente	113
4.2.3 Elemente und Datentypen	114
4.2.4 Komplexe Typen mit und ohne Namen	114
4.2.5 Sequenzen	115
4.2.6 Vorgegebene und abgeleitete Datentypen	116
4.2.7 Wie viel wovon?	116
4.3 Genereller Aufbau eines XML-Schemas	116
4.3.1 Das Vokabular	116
4.3.2 Die Komponenten eines XML-Schemas	117
4.4 Datentypen	117
4.4.1 Komplexe Datentypen	118
4.4.2 Inhaltsmodelle und Partikel	118
4.4.3 Erweiterbarkeit durch Wildcards	120
4.4.4 Einfache Typen	120
4.4.5 Benannte oder anonyme Typen	121
4.4.6 Vorgegebene und benutzerdefinierte Datentypen	122
4.4.7 XML-Schema 1.0 – Datentypen – Kurzreferenz	122
4.4.8 Werteraum, lexikalischer Raum und Facetten	126
4.4.9 Ableitung durch Einschränkung	127
4.4.10 Muster und reguläre Ausdrücke	128
4.4.11 Grenzwerte	130

4.4.12	Listen und Vereinigungen	130
4.4.13	Facetten der verschiedenen Datentypen	131
4.5	Definition der Struktur des Dokuments	133
4.5.1	Deklaration von Elementen	133
4.5.2	Attribute	135
4.5.3	Elementvarianten	135
4.5.4	Namensräume in XML-Schema	136
4.5.5	Zielnamensraum	137
4.5.6	Umgang mit lokalen Elementen und Attributen	139
4.6	Häufigkeitsbestimmungen	142
4.7	Default-Werte für Elemente und Attribute	143
4.8	Kompositoren	143
4.8.1	<xsd:sequence>	144
4.8.2	<xsd:all>	144
4.8.3	<xsd:choice>	145
4.8.4	Verschachtelte Gruppen	145
4.9	Arbeit mit benannten Modellgruppen	146
4.10	Definition von Attributgruppen	147
4.11	Schlüsselemente und Bezüge darauf	148
4.11.1	Eindeutigkeit	148
4.11.2	Bezüge auf Schlüsselemente	150
4.12	Kommentare	152
4.13	Ableitung komplexer Datentypen	152
4.13.1	Erweiterungen komplexer Elemente	152
4.13.2	Einschränkung komplexer Elemente	153
4.13.3	Steuerung der Ableitung von Datentypen	154
4.13.4	Abstraktionen	155
4.13.5	Gemischtwaren	156
4.13.6	Leere oder Nichts	158
4.13.7	Wiederverwendbarkeit	158
4.14	Designvarianten	160
4.14.1	Babuschka-Modelle	160
4.14.2	Stufenmodelle	161
4.15	Übernahme von Schema-Definitionen	162
4.15.1	Schemas inkludieren	163
4.15.2	Schemas importieren	165
4.15.3	Zuordnung von Schemas in XML-Dokumenten	169

4.16 XML-Schema 1.0 – Kurzreferenz	170
4.17 Exkurs zu XML-Schema 1.1	180
4.17.1 Versicherungen	180
4.17.2 Lockerungen der Regeln für Inhaltsmodelle	180
4.17.3 Offene Modelle	182
4.17.4 Schemaweite Attribute	182
4.17.5 Anpassen von Schemas	183
4.17.6 Neue Datentypen	183
4.17.7 Einsatz bedingter Datentypen	184

5 Navigation und Verknüpfung 187

5.1 Datenauswahl mit XPath	187
5.1.1 Baummodell und XPath-Ausdrücke	188
5.1.2 Vom Dokument zum Knotenbaum	188
5.1.3 Dokumentreihenfolge	190
5.1.4 Knotentypen	191
5.1.5 Lokalisierungspfade	192
5.1.6 Ausführliche Schreibweise	194
5.1.7 Lokalisierungsstufen und Achsen	194
5.1.8 Knotentest	198
5.1.9 Filtern mit Prädikaten	199
5.1.10 Test von XPath-Ausdrücken	199
5.1.11 XPath 1.0-Funktionen	201
5.2 XPath 2.0	205
5.2.1 Erweitertes Datenmodell	206
5.2.2 Neue Konstrukte für Ausdrücke	206
5.2.3 Neue Datentypen	207
5.2.4 Neue Operatoren	208
5.2.5 Die erweiterte Funktionenbibliothek	208
5.3 XPath 3.0	220
5.4 Verknüpfungen mit XLink	225
5.4.1 Mehr als Anker in HTML	225
5.4.2 Beziehungen zwischen Ressourcen	226
5.4.3 Link-Typen und andere Attribute	227
5.4.4 Beispiel für einen einfachen Link	230
5.4.5 Beispiel für einen Link vom Typ »extended«	230
5.4.6 XLink-Anwendungen	232

5.5	XBase	232
5.6	Über XPath hinaus: XPointer	233
5.6.1	URIs und Fragmentbezeichner	233
5.6.2	XPointer-Syntax	234
5.6.3	Das Schema element()	235
5.6.4	Das Schema xmlns()	235

6 Datenausgabe mit CSS 239

6.1	Cascading Stylesheets für XML	241
6.2	Arbeitsweise eines Stylesheets	241
6.3	Anlegen von Stylesheets	243
6.4	Vererben und Überschreiben	244
6.5	Selektortypen	246
6.6	Attributselektoren	247
6.7	Kontext- und Pseudoselektoren	247
6.8	Schriftauswahl und Textformatierung	248
6.8.1	Absolute Maßeinheiten	248
6.8.2	Relative Maßeinheiten	248
6.8.3	Prozentangaben	249
6.8.4	Maßangaben über Schlüsselwörter	249
6.9	Farbauswahl	249
6.10	Blöcke, Ränder, Rahmen, Füllung und Inhalt	249
6.11	Stylesheet-Kaskaden	251
6.12	Auflösung von Regelkonflikten	252
6.13	Zuordnung zu XML-Dokumenten	252
6.14	Schwächen von CSS	254

7 Umwandlungen mit XSLT 255

7.1	Sprache für Transformationen	255
7.1.1	Bedarf an Transformationen	255
7.1.2	Grundlegende Merkmale von XSLT	257

7.1.3	XSLT-Prozessoren	258
7.1.4	Die Elemente und Attribute von XSLT	259
7.1.5	Verknüpfung zwischen Stylesheet und Dokument	261
7.1.6	Das Element <stylesheet>	262
7.1.7	Top-Level-Elemente	262
7.1.8	Template-Regeln	263
7.1.9	Attributwert-Templates	265
7.1.10	Zugriff auf die Quelldaten	266
7.2	Ablauf der Transformation	267
7.2.1	Startpunkt Wurzelknoten	268
7.2.2	Anwendung von Templates	268
7.2.3	Rückgriff auf versteckte Templates	268
7.2.4	Auflösung von Template-Konflikten	269
7.3	Stylesheet mit nur einer Template-Regel	269
7.4	Eingebaute Template-Regeln	270
7.5	Designalternativen	271
7.6	Kontrolle der Knotenverarbeitung	273
7.6.1	Benannte Templates	274
7.6.2	Template-Auswahl mit XPath-Mustern	275
7.6.3	Kontext-Templates	277
7.6.4	Template-Modi	278
7.7	Datenübernahme aus der Quelldatei	280
7.8	Nummerierungen	281
7.8.1	Einfach	281
7.8.2	Mehrstufig	282
7.8.3	Zusammengesetzt	283
7.9	Verzweigungen und Wiederholungen	284
7.9.1	Bedingte Ausführung von Templates	284
7.9.2	Wahlmöglichkeiten	285
7.9.3	Schleifen	287
7.10	Sortieren und Gruppieren von Quelldaten	289
7.10.1	Sortierschlüssel	289
7.10.2	Sortierreihenfolge	291
7.11	Parameter und Variablen	292
7.11.1	Parameterübergabe	292
7.11.2	Globale Parameter	293
7.11.3	Lokale und globale Variablen	293
7.11.4	Eindeutige Namen	294

7.11.5	Typische Anwendungen von Variablen in XSLT	294
7.11.6	Rekursive Templates	299
7.12	Hinzufügen von Elementen und Attributen	301
7.12.1	Elemente und Attribute aus vorhandenen Informationen erzeugen	302
7.12.2	Attributlisten	303
7.12.3	Texte und Leerräume	304
7.12.4	Kontrolle der Ausgabe	304
7.13	Zusätzliche XSLT-Funktionen	305
7.13.1	Zugriff auf mehrere Quelldokumente	305
7.13.2	Zahlenformatierung	307
7.13.3	Liste der zusätzlichen Funktionen in XSLT	308
7.14	Mehrfache Verwendung von Stylesheets	309
7.14.1	Stylesheets einfügen	309
7.14.2	Stylesheets importieren	310
7.15	Übersetzungen zwischen XML-Vokabularen	310
7.15.1	Diverse Schemas für gleiche Informationen	311
7.15.2	Angleichung durch Transformation	312
7.16	Umwandlung von XML in HTML und XHTML	314
7.16.1	Datenübernahme und Ergänzungen	314
7.16.2	Generieren von CSS-Stylesheets	316
7.16.3	Aufbau einer Tabelle	316
7.16.4	Transformation in XHTML	317
7.16.5	XHTML-Module	318
7.16.6	Allgemeine Merkmale von XHTML	319
7.16.7	Aufbau eines XHTML-Dokuments	320
7.16.8	Automatische Übersetzung	321
7.17	XSLT-Editoren	322
7.18	Kurzreferenz zu XSLT 1.0	323
7.19	XSLT 2.0	331
7.19.1	Die wichtigsten Neuerungen	332
7.19.2	Neue Funktionen in XSLT 2.0	338
7.19.3	Neue Elemente	339

8 **Formatierung mit XSL** 343

8.1	Transformation und Formatierung	343
8.2	Formatierungsobjekte	344

8.3	Baum aus Bereichen – Areas	345
8.4	XSL-Bereichsmodell	345
8.4.1	Block-Bereiche und Inline-Bereiche	346
8.4.2	XSL und CSS	346
8.5	Testumgebung für XSL	347
8.6	Aufbau eines XSL-Stylesheets	349
8.6.1	Baum der Formatierungsobjekte	350
8.6.2	Seitenaufbau	350
8.6.3	Seitenfolgen	351
8.6.4	Einfügen von Fließtext	351
8.6.5	Blockobjekte	352
8.7	Verknüpfung mit dem Dokument und Ausgabe	354
8.8	Inline-Formatierungsobjekte	355
8.9	Ausgabe von Tabellen	356
8.9.1	Tabellenstruktur	356
8.9.2	Zellinhalte	357
8.10	Listen	359
8.11	Gesucht: visuelle Editoren	360
8.12	Übersicht über die Formatierungsobjekte von XSL	361
8.12.1	Übergeordnete Objekte	361
8.12.2	Blockformatierung	363
8.12.3	Inline-Formatierung	363
8.12.4	Tabellenformatierung	364
8.12.5	Listenformatierung	365
8.12.6	Formatierung für Verknüpfungen	365
8.12.7	Out-of-Line-Formatierung	366
8.12.8	Andere Objekte	366
9	Abfragen mit XQuery	367
9.1	Datenmodell und Verfahren	367
9.1.1	Zur Syntax	369
9.1.2	Instanzen des Datenmodells	370
9.1.3	W3C-Empfehlungen zu XQuery	372
9.2	Abfragepraxis	373
9.2.1	XQuery-Modul	374
9.2.2	Zugriff über das Web	375

9.3	FLWOR-Ausdrücke	375
9.3.1	Variablen in XQuery	379
9.3.2	Steuerung der Ausgabe	379
9.4	Fortgeschrittene Optionen	380
9.4.1	Auswertung zweier verbundener Dokumente	381
9.4.2	Kollektionen auswerten	382
9.4.3	Benutzerdefinierte Funktionen	382
9.5	Implementierungen	384

10 Programmierschnittstellen für XML 387

10.1	Abstrakte Schnittstellen: DOM und SAX	387
10.2	Document Object Model (DOM)	389
10.2.1	DOM Level	390
10.2.2	Objekte, Schnittstellen, Knoten und Knotentypen	391
10.2.3	Die allgemeine Node-Schnittstelle	391
10.2.4	Knotentypen und ihre Besonderheiten	393
10.2.5	Zusätzliche Schnittstellen	394
10.2.6	Zugriff über Namen	395
10.2.7	Verwandtschaften	395
10.2.8	Das Dokument als DOM-Baum	396
10.2.9	Document – die Mutter aller Knoten	398
10.2.10	Elementknoten	399
10.2.11	Textknoten	399
10.2.12	Attributknoten sind anders	400
10.2.13	Dokumentfragmente	400
10.2.14	Fehlerbehandlung	401
10.3	DOM-Implementierungen	401
10.4	Die MSXML-Implementierung von DOM	402
10.4.1	Schnittstellen in MSXML	403
10.4.2	Erweiterungen für Laden und Speichern	406
10.4.3	Erweiterungen der Node-Schnittstelle	406
10.5	Fingerübungen mit DOM	408
10.5.1	Daten eines XML-Dokuments abfragen	409
10.5.2	Zugriff über Elementnamen	414
10.5.3	Zugriff auf Attribute	415
10.5.4	Abfrage über einen Attributwert	417
10.5.5	Fehlerbehandlung	418

10.5.6	Neue Knoten einfügen	419
10.5.7	Neue Elementknoten	422
10.5.8	Neue Attributknoten	423
10.5.9	Unterelementknoten und Textknoten	423
10.5.10	Request und Response	424
10.6	Alternative zu DOM: Simple API for XML (SAX)	425
10.6.1	Vergesslicher Beobachter am Datenstrom	426
10.6.2	SAX2 unter Java	426
10.6.3	Der Kern der SAX-Schnittstellen	428
10.6.4	»ContentHandler«	429
10.6.5	Attribute	430
10.6.6	SAX2-Erweiterungen	431
10.6.7	Hilfsklassen	433
10.6.8	SAXParser und XMLReader	434
10.6.9	Konfigurieren des Parsers	435
10.6.10	Kleine Lagerauswertung mit SAX	438
10.6.11	Aufruf des Parsers	440
10.6.12	Fehlerbehandlung	441
10.6.13	SAX-Beispiel 1	443
10.6.14	SAX-Beispiel 2	446
10.6.15	SAX und DOM	448
10.7	Arbeit mit XML-Klassen in Visual Basic	449
10.7.1	XML-Architektur im .NET Framework	449
10.7.2	Lesen von XML-Daten	451
10.7.3	XMLReader im Vergleich zum SAX-Reader	452
10.7.4	Arbeitsweise von XmlReader	452
10.7.5	XML-Dokument mit XMLTextReader auswerten	453
10.7.6	Lesen von XML-Fragmenten	456
10.7.7	Validierung anhand von XML-Schemas oder DTDs	458
10.7.8	Schreiben von XML-Daten	460
10.7.9	XmlTextWriter	464
10.7.10	XML-Serialisierung und -Deserialisierung	468
10.8	Zugriff auf XML-Daten mit LINQ to XML	473
10.8.1	LINQ to XML	474
10.8.2	X-Klassen	474
10.8.3	Functional Construction	475
10.8.4	XML-Literale	476
10.8.5	Schreiben und Laden von XML-Dateien	479

11 Kommunikation zwischen Anwendungen 481

11.1 XML-Webdienste	482
11.1.1 Gemeinsame Nutzung von Komponenten	482
11.1.2 Offen gelegte Schnittstellen	482
11.1.3 Endpunkte	482
11.2 Beispiel für einen Webdienst	483
11.2.1 Webdienst mit ASP.NET	483
11.2.2 Einrichten eines Webdienstes	484
11.2.3 Webmethoden	486
11.2.4 Test des Webdienstes	486
11.2.5 Aufruf einer Methode	488
11.2.6 Nutzen des Webdienstes über eine Anwendung	488
11.2.7 Einfügen des Verweises auf den Webdienst	489
11.2.8 Proxyklasse	490
11.3 Nachrichten mit SOAP	491
11.3.1 Ein Rahmen für Nachrichten	491
11.3.2 Grundform einer SOAP-Nachricht	492
11.4 Dienstbeschreibung	495
11.4.1 Das WSDL-Vokabular	495
11.4.2 WSDL unter ASP.NET	496
11.5 Webdienste registrieren und finden	498
11.5.1 UDDI	498
11.5.2 Disco	499
11.5.3 Safety first!	500

12 XML in Office-Anwendungen 501

12.1 XML in Microsoft Office	502
12.1.1 Der Standard Office Open XML	502
12.1.2 Open XML für Excel	504
12.1.3 Open XML für Word	507
12.2 Die Alternative OpenDocument	508
12.3 Einsatz benutzerdefinierter Schemas in Office 2013	512
12.3.1 Zuordnen eines Schemas	513
12.3.2 Optionen beim Öffnen von XML-Dokumenten	514
12.3.3 Daten als XML-Tabelle übernehmen	515
12.3.4 XML-Tabellenbereiche	517

12.3.5	XML-Zuordnungen	518
12.3.6	Datenaktualisierung	519
12.3.7	Öffnen als schreibgeschützte Arbeitsmappe	520
12.3.8	Verwenden von XSLT-Stylesheets	520
12.3.9	Datenquelle und Tabelle manuell verknüpfen	523
12.3.10	»XmlMap«-Objekte	526
12.3.11	Tabelle auf Basis eines eigenen Schemas	526
12.3.12	Fehlererkennung	527
12.3.13	XML-Dokumente erzeugen	527
12.3.14	Schema-Einschränkungen	528
12.4	XML-basierte Formulare mit InfoPath 2013	528
12.4.1	Werkzeug für dynamische Formulare	529
12.4.2	Fingerübung mit InfoPath	529
12.4.3	Formular mit eigener Datenstruktur	530
12.4.4	Formularentwurf »from Scratch«	533
12.4.5	XPath-Ausdrücke für Berechnungen	534
12.4.6	Schema-Limits	535
12.4.7	Validierung per Schema	535
12.4.8	Zusatzprüfungen	535
12.4.9	Formularansichten	536
12.4.10	Veröffentlichung von Formularen	536
12.4.11	Vorlagenarchiv	536
12.4.12	Formulare ausfüllen	539
12.4.13	Speichern der eingegebenen Daten	540
12.4.14	Austausch mit anderen Anwendungen	541
13	Mapping – von XML oder nach XML	543
13.1	Codegenerierung für Transformationen	543
13.1.1	Oberfläche und Dateiformate	544
13.1.2	Funktionsbibliotheken	546
13.1.3	Von Schema zu Schema	547
13.2	Datenausgabe	549
13.3	Stylesheet-Generierung	550
13.4	Eigene Funktionen	551
13.5	Mapping von Datenbankdaten	552
13.6	Mapping für Excel-Tabellen	554
13.7	EDIFACT und ANSI X12	556

14 Publizieren mit EPUB 559

14.1 Electronic Publication	559
14.1.1 Content Documents	560
14.1.2 Paket-Format	563
14.1.3 Open Container Format	567
14.2 Tools für EPUB	569
14.3 Autorentools	572

15 HTML5 und XHTML 575

15.1 Unerfüllte Erwartungen	575
15.2 Die Wiederbelebung von HTML	578
15.3 Fixer oder lebendiger Standard?	579
15.4 Was ist neu?	579
15.5 DOCTYPE und Ausführungsmodus	580
15.6 HTML vs XHTML	581
15.7 Aussichten	582
15.8 XML-Inhalte im Browser	583
15.8.1 Die Wiederbelebung von SVG	583
15.8.2 Dynamische Illustrationen	584
15.9 Freiwillige gesucht	590

Anhang 591

A Glossar	591
B Webressourcen	599
B.1 Webseiten für Entwickler	599
B.2 Liste von Empfehlungen des W3C	602
B.3 Liste von wichtigen Namensräumen des W3C	605

Index	607
-------------	-----