

Einführung in XML

Seminarunterlage

Version: 3.05



Dieses Dokument wird durch die ORDIX AG veröffentlicht.

Copyright ORDIX AG. Alle Rechte vorbehalten.

Alle Produkt- und Dienstleistungs-Bezeichnungen sind Warenzeichen oder eingetragene Warenzeichen der jeweiligen Firmen und beziehen sich auf Eintragungen in den USA oder USA-Warenzeichen.

Weitere Logos und Produkt- oder Handelsnamen sind eingetragene Warenzeichen oder Warenzeichen der jeweiligen Unternehmen.

Kein Teil dieser Dokumentation darf ohne vorherige schriftliche Genehmigung der ORDIX AG weitergegeben oder benutzt werden.

Adressen der ORDIX AG

Die ORDIX AG besitzt folgende Geschäftsstellen

ORDIX AG
Karl-Schurz-Str. 19a
D-33100 Paderborn
Tel.: (+49) 0 52 51 / 10 63 - 0
Fax.: (+49) 01 80 / 1 67 34 90

ORDIX AG
An der alten Ziegelei 5
D-48157 Münster
Tel.: (+49) 02 51 / 9 24 35 – 00
Fax.: (+49) 01 80 / 1 67 34 90

ORDIX AG
Welser Str. 9
D-86368 Gersthofen
Tel.: (+49) 0821 / 507 492 –00
Fax.: (+49) 01 80 / 1 67 34 90

ORDIX AG
Kreuzberger Ring 13
D-65205 Wiesbaden
Tel.: (+49) 06 11 / 7 78 40 – 00
Fax.: (+49) 01 80 / 1 67 34 90

ORDIX AG
Wikingerstraße 18-20
D-51107 Köln
Tel.: (+49) 02 21 / 8 70 61 – 0
Fax.: (+49) 01 80 / 1 67 34 90

Internet: <http://www.ordix.de>

Email: seminare@ordix.de

Inhaltsverzeichnis

| | | |
|----------|---|-----------|
| 1 | Was ist XML? | 7 |
| 1.1 | Historie | 8 |
| 1.2 | Standard Generalized Markup Language (SGML) | 10 |
| 1.3 | Hypertext Markup Language (HTML) | 11 |
| 1.4 | eXtensible Markup Language (XML) | 12 |
| 1.5 | XHTML (eXtensible HTML) | 13 |
| 1.6 | Unterschiede SGML und XML | 14 |
| 1.7 | Unterschiede HTML und XML | 16 |
| 1.8 | URI & URL | 17 |
| 1.9 | Übergeordnete Zusammenhänge | 18 |
| 1.10 | XML-Editoren | 20 |
| 1.11 | Parser | 21 |
| 1.12 | Browser | 22 |
| 1.13 | Verwendung von XML | 23 |
| 1.14 | XML Anwendungen | 24 |
| 1.14.1 | WML (Wireless Markup Language) | 25 |
| 1.14.2 | SVG (Scalable Vector Graphics) | 26 |
| 1.14.3 | MathML (Mathematical Markup Language) | 27 |
| 2 | XML-Dokumente | 29 |
| 2.1 | XML-Daten / XML-Dokumente | 30 |
| 2.2 | Grundlegender Aufbau | 31 |
| 2.2.1 | Prolog | 32 |
| 2.2.2 | Prolog: XML-Deklaration | 33 |
| 2.2.3 | Prolog: XML-Deklaration – Zeichensätze | 34 |
| 2.2.4 | Prolog: Processing Instruction (PI) | 36 |
| 2.2.5 | Kommentare | 37 |
| 2.2.6 | Rumpf und Epilog | 38 |
| 2.2.7 | Tags | 39 |
| 2.2.8 | Namensgebung von Tags | 40 |
| 2.2.9 | Elemente | 41 |
| 2.2.10 | Content und leere Elemente | 42 |
| 2.2.11 | Schachtelung | 43 |
| 2.3 | Fachliche Struktur | 44 |
| 2.4 | Struktur Baum | 45 |
| 2.5 | XML Struktur | 46 |
| 2.6 | Reservierte Zeichen | 48 |
| 2.7 | Entities (Ersetzungen) | 49 |
| 2.8 | Eigene Entities definieren und verwenden | 51 |
| 2.9 | CDATA-Blöcke | 52 |
| 2.10 | Attribute | 53 |
| 2.11 | Vordefinierte Attribute | 55 |
| 2.12 | Verwendung von Elementen/Attributen | 57 |
| 2.13 | „Wohlgeformtheit“ | 58 |
| 2.14 | Gültigkeit | 59 |
| 2.15 | Namensräume | 60 |
| 2.15.1 | Verwendung von Namensräumen | 62 |
| 2.15.2 | Default Namensraum | 63 |
| 2.15.3 | Spezielle URIs (festgelegte Namensräume) | 64 |
| 3 | DTD | 65 |
| 3.1 | DTD (Dokument-Typ-Definition) | 66 |
| 3.2 | DTD Integration in das XML-Dokument | 67 |
| 3.3 | Interne DTD | 68 |
| 3.4 | Externe DTD | 69 |
| 3.5 | Private Externe DTDs (Systembezeichner) | 70 |
| 3.6 | Öffentliche Externe DTDs (Öffentliche Bezeichner) | 71 |

| | | |
|----------|--|------------|
| 3.7 | Genereller Aufbau einer DTD..... | 72 |
| 3.8 | Elemente | 73 |
| 3.8.1 | Inhaltsmodell (content model) | 74 |
| 3.8.2 | Einfache Inhaltstypen | 75 |
| 3.8.3 | Komplexe Inhaltstypen | 76 |
| 3.8.4 | Verknüpfungen | 77 |
| 3.8.5 | Operatoren / Indikatoren / Kardinalitäten | 78 |
| 3.9 | XML und korrespondierende DTD | 79 |
| 3.10 | Attribute | 81 |
| 3.10.1 | Vorgaben | 83 |
| 3.11 | Attributtyp | 85 |
| 3.11.1 | CDATA | 85 |
| 3.11.2 | Aufzählung | 86 |
| 3.11.3 | NMTOKEN(S)..... | 87 |
| 3.11.4 | ID | 89 |
| 3.11.5 | IDREF | 90 |
| 3.11.6 | IDREFS | 91 |
| 3.12 | Vordefinierte Attribute: xml:space und xml:lang..... | 92 |
| 3.13 | Typische Kombinationen..... | 93 |
| 3.14 | Entities..... | 94 |
| 3.14.1 | Allgemeine Entities | 95 |
| 3.14.2 | Parameter-Entity | 97 |
| 3.14.3 | Beispiel zu Parameter-Entity | 98 |
| 3.15 | Diverses | 99 |
| 4 | XSLT | 100 |
| 4.1 | 1 Warum XSL? | 101 |
| 4.2 | XSLT-Prozessor | 104 |
| 4.3 | Einbindung von XSL in XML | 105 |
| 4.4 | Kommandozeilenorientierter XSLT-Prozessor..... | 106 |
| 4.5 | Aufbau einer XSL Datei..... | 107 |
| 4.6 | Inhalte von <xsl:stylesheet> | 108 |
| 4.7 | Definition der Ausgabe-Art | 109 |
| 4.8 | Templates..... | 111 |
| 4.8.1 | Template Definition | 112 |
| 4.8.2 | Bewegen innerhalb der XML-Datei | 114 |
| 4.8.3 | Beispiel | 115 |
| 4.8.4 | XML Datei..... | 115 |
| 4.8.5 | Baumstruktur | 116 |
| 4.8.6 | Aufruf von weiteren Templates | 117 |
| 4.8.7 | Built-In Templates | 118 |
| 4.9 | Ausgabe der Inhalte | 119 |
| 4.9.1 | Ausgabe Internet Explorer..... | 120 |
| 4.9.2 | XSL-Datei | 121 |
| 4.10 | Kommentare..... | 122 |
| 4.11 | Texte ausgeben | 123 |
| 4.12 | Schleifen..... | 124 |
| 4.13 | Sortieren | 126 |
| 4.14 | Bedingungen | 128 |
| 4.15 | Integration von CSS-Anweisungen | 131 |
| 5 | XPath | 133 |
| 5.1 | XPath..... | 134 |
| 5.2 | Lokalisierungspfad | 136 |
| 5.3 | Lokalisierungsschritte..... | 137 |
| 5.4 | Knotentypen | 138 |
| 5.5 | Wurzelknoten | 140 |
| 5.6 | Achsen | 141 |
| 5.6.1 | Achse: self | 142 |
| 5.6.2 | Achse: attribute | 143 |

| | | |
|----------|---|------------|
| 5.6.3 | Achse: namespace | 143 |
| 5.6.4 | Achse: child | 144 |
| 5.6.5 | Achse: parent | 145 |
| 5.6.6 | Achse: descendant | 146 |
| 5.6.7 | Achse: descendant-or-self | 147 |
| 5.6.8 | Achse: ancestor | 148 |
| 5.6.9 | Achse: ancestor-or-self | 149 |
| 5.6.10 | Achse: following-sibling | 150 |
| 5.6.11 | Achse: following | 151 |
| 5.6.12 | Achse: preceding-sibling | 152 |
| 5.6.13 | Achse: preceding | 153 |
| 5.7 | Knotentest | 154 |
| 5.8 | Prädikate | 156 |
| 5.9 | Abgekürzte XPath-Syntax | 157 |
| 5.10 | Operatoren und Funktionen | 159 |
| 5.11 | Positionsfunktionen | 160 |
| 5.12 | Operatoren | 161 |
| 5.13 | Aggregatsfunktionen | 163 |
| 5.14 | Zeichenkettenfunktionen | 165 |
| 5.15 | Konvertierungsfunktionen | 168 |
| 6 | XSLT für Fortgeschrittene | 170 |
| 6.1 | Attribute Value Template | 171 |
| 6.2 | Grafiken | 172 |
| 6.3 | Links | 173 |
| 6.4 | <xsl:attribute> | 174 |
| 6.4.1 | Beispiel: <xsl:attribute> | 176 |
| 6.5 | <xsl:element> | 178 |
| 6.6 | Nummerierung | 180 |
| 6.6.1 | Beispiel: Nummerierung | 182 |
| 6.7 | Variablen | 184 |
| 6.8 | Parameter | 186 |
| 6.9 | Stylesheet-Parameter | 187 |
| 6.10 | Template-Parameter | 189 |
| 6.11 | Beispiel: Template-Parameter | 191 |
| 6.12 | Templates mit Modus | 193 |
| 6.13 | Kopieren von Elementen | 194 |
| 6.13.1 | Beispiel Kopieren von Elementen – XSL | 195 |
| 6.13.2 | Beispiel: Ergebnis – XML | 197 |
| 6.14 | Modularisierung | 198 |
| 7 | XML Schema | 199 |
| 7.1 | Was ist ein XML Schema? | 200 |
| 7.2 | Dokumenttypdefinition – DTD | 202 |
| 7.3 | Vor- und Nachteile | 203 |
| 7.4 | Verknüpfung mit einem XML-Dokument | 204 |
| 7.5 | Aufbau einer XML-Schema Datei | 205 |
| 7.6 | Datentypen | 206 |
| 7.7 | SimpleType | 207 |
| 7.7.1 | Beispiele | 208 |
| 7.8 | Einfache Elemente | 209 |
| 7.9 | Attribute | 211 |
| 7.10 | Komplexe Elemente | 213 |
| 7.11 | Content-Modell | 214 |
| 7.12 | Content-Modell: simple + Attribute | 215 |
| 7.13 | Content-Modell: sequence + Attribute | 217 |
| 7.14 | Eigener SimpleType | 221 |
| 7.15 | Facette | 222 |
| 7.15.1 | String-Typen | 222 |
| 7.15.2 | Enumeration | 223 |

| | | |
|----------|---|------------|
| 7.15.3 | Reguläre Ausdrücke (pattern) | 224 |
| 7.15.4 | WhiteSpace | 225 |
| 7.15.5 | Zahl-Typen | 226 |
| 7.15.6 | Datums/Zeit-Typen | 228 |
| 7.16 | Content-Modell: choice + Attribute | 229 |
| 7.17 | Content-Modell: all + Attribute | 231 |
| 7.18 | Content-Modell: sequence - choice – all | 233 |
| 7.19 | Content-Modell: EMPTY + Attribute | 234 |
| 7.20 | Globale und lokale Definitionen | 235 |
| 7.21 | Globale und lokale Elemente | 236 |
| 7.22 | Globale und lokale Attribute | 237 |
| 7.23 | Globale und lokale Typen | 238 |
| 7.24 | Content-Modell: Eigener ComplexType | 239 |
| 7.25 | Content-Modell: NULL Werte darstellen | 242 |
| 7.26 | Eindeutige Schlüssel | 244 |
| 7.27 | Referenz-Schlüssel | 247 |
| 8 | Einführung XSL-FO | 249 |
| 8.1 | Was ist XSL-FO | 250 |
| 8.2 | XSL-FO Prozessoren | 251 |
| 8.3 | 8.3 Aufbau einer XSL-FO Datei | 254 |
| 8.4 | XSL-FO Bereiche | 255 |
| 8.5 | XSL-FO Pages | 256 |
| 8.6 | XSL-FO Blöcke | 260 |
| 8.7 | Statischer Inhalt | 263 |
| 8.8 | Tabellen | 266 |
| 8.9 | Listen | 270 |
| 8.10 | XSL-FO und XSLT | 274 |