

# Index

COPYRIGHTED MATERIAL



# Index

## SYMBOLS

- < > (angle brackets), 431
- \$ (dollar sign), 75
- / (forward slash), 13, 266
- [ ] (optional) character, 431
- | (pipe) character, 278, 431
- ' (single quote), 93
- \* (wildcard) character, 158, 277

## A

**absolute paths, XPath scripting language, 266**

`absoluteChildNumber()` **method, 81**

**abstracted classes, 33, 420**

**abstraction**

- end of chapter exercise, 54
- object model, 232

**accessor functions, XPath scripting language, 282**

**Active Server Pages (ASP), 52–53**

**addition XPath numeric operator, 284**

**algorithms, index, 200**

**alias, in SELECT statement, 92**

**alphanumeric characters, element naming rules, 17**

`ALTER INDEX` **command, 122**

`ALTER TABLE` **command, 89, 122**

**ancestor node, XQuery language, 303**

`ancestorChildNumber()` **method, 81**

**AND logical operator, 93**

**angle brackets (< >), 431**

**annotations, 193–195**

**ANSI characters, 26**

**answers to exercise. See also end of chapter exercise**

- abstracted classes, 33, 420
- DTD (Document Type Definition), 426
- HTML script errors, 421
- NXD (native XML database), 424–425
- object model, 423–424
- Oracle database, 421–422
- script exercise, 420
- SOAP (Simple Object Access Protocol), 423
- SQL Server database, 422
- SQL (Structure Query Language), 421
- web page determination, 421
- XForms, 426
- XLink, 426
- XML script errors, 421
- XPath scripting language, 425
- XPointer, 426
- XQuery language, 425–426
- XSD (XML Schema Definition), 423, 426
- XSL elements, 420–421

`appendChild()` **method, 34**  
**application-specific vocabularies, 387**  
`arc` **attribute, XLink, 321, 330**  
`arcrole` **attribute, XLink, 321**  
**AS clause**  
    in `ORDER BY` clause, 101–102  
    in subqueries, 114–115  
**ASCII code character entities, 364–365**  
**ASP (Active Server Pages), 52–53**  
**atomic value, XQuery language, 302**  
**ATTLIST declaration, DTD attributes, 360**  
**Attr class**  
    attributes associated with, 44  
    code, 44–45  
    functions performed by, 32  
**attribute nodes**  
    XPath scripting language, 266  
    XQuery language, 302  
**attributes**  
    child elements, 20  
    DTD (Document Type Definition)  
        ATTLIST declaration, 360  
        defaults, 362–364  
        types, 361–362  
    how to use, 21  
    metadata representation, 22  
    with multiple values, 21  
    name-value pairs, 14  
    object model, 231  
    population example, 19–20  
    programming complexity, 21  
    syntax rules, 14–15  
    XLink (XML Linking Language), 320  
**AUTO mode, FOR XML clause, 169, 174–177**  
**axes, XPath scripting language, 279–280**

## B

**Backus–Naur Form syntax notation, 431–432**  
**binary text object, 151**  
**binding data, with XPath scripting language, 346–347**  
**bioinformatics, commercial uses of XML, 393**

**bitmaps, indexes, 258**  
**body section, SOAP, 211**  
**Boolean functions, XPath scripting language, 286**  
**branch section, index, 257**  
**browsers**  
    browser-based control tools, Oracle database, 161  
    Internet Explorer  
        defined, 1  
        executing XForms in, 341  
        XLink support, 320  
        XPointer and, 337  
    Netscape, 1  
    non-mainstream browser limitations, 12  
**B2B (Business-to-Business)**  
    commercial uses of XML, 393  
    document transfers, 220–222  
**built-in entities, 364–365**  
**bulk data transfers, 5**

## C

`calculate` **property, XForms, 344**  
**cardinality**  
    DTD (Document Type Definition) elements, 357–360  
    XSD (XML Schema Definition), 373–374  
**cartographical data, commercial uses of XML, 394**  
**cascading style sheets (CSS), 4**  
**case sensitivity**  
    basic syntax rules, 14  
    XQuery syntax rules, 303  
**CAST function, 147**  
**catalogs, commercial uses of XML, 393**  
**categories, DTD elements, 354–355**  
**CDATA SECTION**  
    classes, 52  
    overview, 24  
**Central Processing Unit (CPU), 143**  
**change commands, metadata, 89**  
**character data, DTD, 352**

**Character Large Binary Object (CLOB)**

DB2 database, 442–444

overview, 261

specification, 132

**child elements, 20****child node, XQuery language, 302**`childNumber()` method, 81**classes**

abstracted, 33, 420

Attr

attributes associated with, 44

code, 44–45

functions performed by, 32

CDATA SECTION, 52

Comment, 52

descriptions, 32

Document

attributes and methods of, 39

documentElement attribute, 41

end of chapter exercise, 55

events and, 42

functions performed by, 32

inheritance and, 38

load() method, 40

Element

attributes and methods of, 42

code, 43

functions performed by, 32

tagName attribute, 42

hierarchical structure, 30–31

NamedNodeMap, 35

Node

appendChild method, 34

attributes and methods of, 33

functions performed by, 32

Try It Out exercise, 35–37

NodeList

attributes and methods of, 34

functions performed by, 32

object model, 231

objects versus, 32

parseError

attributes and methods, 47

code, 48–49

ProcessingInstruction, 52

Text

attributes and methods, 45–46

code, 46

functions performed by, 32

**CLOB (Character Large Binary Object)**

DB2 database, 442–444

overview, 261

specification, 132

**closing tags**

how to use, 2

syntax rules, 14

**collection of nodes, 30****collections, object model, 231****command example, XQuery language, 296–297**

Comment class, 52

**comment nodes**

XPath scripting language, 266

XQuery language nodes, 302

**comments**

basic syntax rules, 17

XQuery syntax rules, 303

**commercial uses of XML**

bioinformatics, 393

B2B (Business to Business), 393

cartographical data, 394

catalogs, 393

data transfers, 393

document management, 393

genetics, 393

geographical data, 394

geological data, 394

geospatial data, 394

incomplete data, 393

manufacturing, 393

medicine, 393

news feeds, 393

personalization, 393

### **commercial uses of XML (continued)**

- reasons for using, 392
- web page personalization, 393
- Web Services, 393

### **COMMIT statement, 121**

### **complex data**

- management, 382
- types, 378–379

### **compression, XML advantages, 3**

### **concatenation**

- XMLCONCAT method, 135
- XPath expressions, 278

### **concurrency, NXD, 261**

### **conditional expressions, 70**

### **constraint property, XForms, 344**

### **constructor functions, XPath scripting language, 283–284**

### **content**

- classification, NXD, 260–261
- document, editing and removing, 159–161
- DTD elements, 355–357
- element, 18
- management system, document-centric documents, 260

### **context functions, XPath scripting language, 276, 289–290**

### **context indexes, 259**

### **Control Center, DB2 database, 443**

### **control structure elements**

- xsl:choose, 72–73
- xsl:for--each, 71
- xsl:if, 70–71
- xsl:otherwise, 72–73
- xsl:param, 75
- xsl:sort, 74
- xsl:variable, 75
- xsl:when, 72–73

### **correlated subqueries, 112**

### **CPU (Central Processing Unit), 143**

### **CREATE TABLE command, 89, 122**

### **cross joins, 108**

### **CSS (cascading style sheets), 4**

### **current function, 77**

### **custom data types, XSD, 375**

### **custom entities, DTD, 365–369**

### **custom-built functions, 303**

## **D**

### **DAD (Document Access Definition), 437**

### **data access, DOM, 11**

### **data islands**

- defined, 7
- Try It Out exercise, 9–10

### **data management, 382**

### **data order, XML advantages, 3**

### **data retrieval elements**

- xsl:output, 70
- xsl:text, 70
- xsl:value--of, 69–70

### **data structure, object model, 225–226**

### **data transfers, bulk, 5**

### **data types**

- complex, 378–379
- custom, 375
- date, 372–373
- double, 372
- miscellaneous, XSD, 373
- numeric, 372
- restriction, 376–377
- simple, 376
- string, 371–372
- table containing, 168
- time, 372–373
- typed, 166–167
- untyped, 166–167
- user-defined types, 146
- in XForms, 342–343
- XMLCLOB, 436
- XMLFILE, 436
- XMLType
  - CLOB object, 132
  - defined, 434
  - discussed, 131
  - methods, 132–133

- subprograms, 132
- uses for, 132
- XMLVARCHAR, 436
- database sample**
  - discussed, 427
  - logical relationship, 428
  - object model, 430
  - physical relationship, 429
- database tables. See tables**
- data-centric documents, 260–261**
- date**
  - formatDate() method, 81
  - XPath scripting language functions, 286–287
- date data types, 373**
- DBMS\_XMLGEN package, 151–152**
- DBPrism native XML database, 386**
- DB2 database**
  - CLOB objects, 442–444
  - Control Center, 443
  - document creation, 437
  - REC2XML function, 437–438
  - XML Extender software, 436
  - XML functions, 438–441
  - XMLCLOB data type, 436
  - XMLFILE data type, 436
  - XMLVARCHAR data type, 436
- debugging XPath tracing functions, 283**
- decimal data type, 372**
- defaults, DTD attributes, 362–364**
- DELETE command**
  - defined, 89
  - table records, 119
- deleting records from tables, 119**
- demographics, document creation, 153–154**
- density expression calculation, table queries, 102**
- depth() method, 81**
- descendent node, XQuery language, 303**
- DHTML (Dynamic HTML), 319**
- Directory Name Service (DNS), 285**
- division XPath numeric operator, 284**
- DNS (Directory Name Service), 285**
- Document Access Definition (DAD), 437**
- Document class**
  - attributes and methods of, 39
  - documentElement attribute, 41
  - end of chapter exercise, 55
  - events and, 42
  - functions performed by, 32
  - inheritance and, 38
  - load() method, 40
- document creation**
  - using DB2 database, 437
  - using Oracle database
    - content, editing and removing, 159–161
    - database storage, 153–154
    - demographics, 153–154
    - discussed, 131–132
    - document retrieval, 154–155
    - duplicated records, 142
    - file size considerations, 143
    - GROUP BY clause, 144
    - highlighted name example, 137
    - implementation, 134–135
    - methods, list of, 135
    - multiple fields, 141
    - multiple XML fragment example, 137–138
    - nested tables, 145
    - packages, 134
    - PL/SQL package, 151–153
    - relational structure, 140
    - results, reading, 155–159
    - root tag, 145
    - sorting order example, 139
    - SYS\_XMLGEN function, 146
    - tag creation, 136
    - TYPE objects, 147
    - white space removal, 138
    - XMLType data type, 131–133
  - using SQL
    - discussed, 122
    - multiple-layer document, 125–126
    - pseudocode, 127–128
    - root tag, 123
    - two-dimensional structure, 123–124

## document nodes

XPath scripting language, 266

XQuery language, 302

**Document Object Model. See DOM**

**Document Type Definition. See DTD**

documentElement **attribute, 41**

## documents

adding to SQL Server, 185–186

data-centric, 260–261

document-centric, 260

generation, ASP, 52–53

object model comparison, 238

splitting data between, XPointer, 333–334

transfers

basic, 205–207

B2B (Business–2–Business), 220–222

external data, 219–222

HTTP protocol, 208

metadata, 210

semantics in, 209–210

SOAP (Simple Object Access Protocol),  
210–214

transformation processing, 208

Web Services, 207–209

type declaration, DTD, 352–353

validation of, DTD, 12–13, 252

**dollar sign (\$), 75**

**DOM (Document Object Model)**

abstracted class, 33

Attr class

attributes associated with, 44

code, 44–45

functions performed by, 32

CDATA Section class, 24, 52

class descriptions, 32

class hierarchical structure, 30–31

Comment class, 52

data access, 11

Document class

attributes and methods of, 39

documentElement attribute, 41

end of chapter exercise, 55

events and, 42

functions performed by, 32

inheritance and, 38

load() method, 40

dynamic access, 11

Element class

attributes and methods of, 42

code, 43

functions performed by, 32

tagName attribute, 42

generic access, 11

hierarchical array structure, 12

HttpRequest object, 50–51

NamedNodeMap class, 35

Node class

appendChild() method, 34

attributes and methods of, 33

functions performed by, 32

Try It Out exercise, 35–37

NodeList class

attributes and methods of, 34

functions performed by, 32

objects

classes versus, 32

hierarchical structure, 30–31

parseError class

attributes and methods, 47

code, 48–49

ProcessingInstruction class, 52

programmatic manipulation, 29

Text class

attributes and methods, 45–46

code, 46

functions performed by, 32

XSL combined with, 82–84

double **data type, 372**

DROP INDEX **command, 122**

DROP TABLE **command, 89**

DROP TYPE **command, 148–149**

**DTD (Document Type Definition)**

attributes

ATTLIST declaration, 360

defaults, 362–364

types, 361–362

character data, 352  
 defined, 433  
 discussed, 246, 351  
 document type declaration, 352–353  
 document validation, 252  
 elements  
   cardinality, 357–360  
   categories, 354–355  
   content, 355–357  
 end of chapter exercise, 380  
 entities  
   ASCII code character, 364–365  
   built-in, 364–365  
   custom, 365–369  
   parameter, 365–369  
 validation of documents, 12–13

**deduplicated table records, 142****duration functions, XPath scripting language, 286–287****dynamic access, DOM, 11****dynamic data, 13****Dynamic HTML (DHTML), 319****E****Element class**

attributes and methods of, 42  
 code, 43  
 functions performed by, 32  
 tagName attribute, 42

**element nodes**

XPath scripting language, 266  
 XQuery language, 302

**element--available function, 77****elements**

attributes, 19–20  
 child, 20  
 content of, 18  
 DTD (Document Type Definition)  
   cardinality, 357–360  
   categories, 354–355  
   content, 355–357  
 extensible, 18

naming rules, 17  
 relationships between, 18  
 syntax rules, 14–15  
 xsl:apply--template, 64–65  
 xsl:attribute, 67  
 xsl:call--template, 66  
 xsl:choose, 72–73  
 xsl:comment, 61  
 xsl:copy, 67–68  
 xsl:copy--of, 68–69  
 xsl:element, 66–67  
 xsl:for--each, 71  
 xsl:if, 70–71  
 xsl:otherwise, 72–73  
 xsl:output, 70  
 xsl:param, 75  
 xsl:processing--instruction, 60  
 xsl:script, 61  
 xsl:sort, 74  
 xsl:stylesheet, 60  
 xsl:template, 61–63  
 xsl:text, 70  
 xsl:transform, 60  
 xsl:value--of, 69–70  
 xsl:variable, 75  
 xsl:when, 72–73

**ELEMENTS mode, FOR XML clause, 169****EMPTY element, 354****end of chapter exercise. See also answers to exercise**

abstracted classes, 54  
 DTD (Document Type Definition), 380  
 HTML script errors, 27  
 NXD (native XML database), 262–263  
 object model, 242–243  
 Oracle database, 162–163  
 script exercise, 55  
 SOAP (Simple Object Access Protocol), 222–223  
 SQL Server database, 202–203  
 SQL (Structured Query Language), 129–130  
 web page determination, 28

### end of chapter exercise (continued)

XForms, 349  
XLink, 349  
XML script errors, 28  
XPath scripting language, 293–294  
XPointer, 349  
XQuery language, 314  
XSD (XML Schema Definition), 223, 380  
XSL elements, 85–86

### ending root tag, syntax rules, 13

### entities, DTD

ASCII code character, 364–365  
built-in, 364–365  
custom, 365–369  
parameter, 365–369

### envelope section, SOAP, 211

### equality XPath numeric operator, 284

### ERDs (Entity Relationship Diagrams)

classification, 104, 322  
object model, 232

### error functions, XPath scripting language, 283

### escape sequence, 24

### events, 42

### execution paths, optional selection of, 70

### exercise. See answers to exercise; end of chapter exercise; Try It Out exercise

### exist() method, 169, 189

### eXist native XML database, 385

### EXISTSNODE method

description of, 133  
string verification, 156

### EXPLICIT mode, FOR XML clause, 169, 179–181

### exponential notation, 99

### expressions

precedence in, 98–99  
XPath scripting language  
concatenation, 278  
list of, 273–274  
multiple path, 278  
numeric, 284  
path, 272

syntax, 268–271

Try It Out exercise, 290–292

wildcard (\*) character, 277

### extended links, XLink

basic interpretation, 328  
extended attribute, 321  
metadata structure, splitting databases by, 322–325  
uses for, 322

### Extensible HTML (XHTML), 433

### Extensible Markup Language. See XML

### Extensible Style Sheet Language Transformations (XSLT), 58, 265, 434

### Extensible Style Sheets. See XSL

### external data, document transfers, 219–222

### EXTRACT method

description of, 133  
SELECT statement, 140

## F

### fields, table queries

header, 91  
name changes, 101–102

### file size considerations, document creation, 143

### flat data, object model, 227–229, 237

### flexibility

in XLink, 332  
XML advantages, 3

### float data type, 372

### FLWOR statement, XQuery language

embedded for loops and communication, 309–310  
for loop, 303–304  
let clause, 308  
order by clause, 307–308  
return clause, 303–305  
where clause, 306–307

### for loop, 277, 303–304

### FOR XML clause

AUTO mode, 169, 174–177  
BINARY BASE64 mode, 170  
ELEMENTS mode, 169

EXPLICIT mode, 169, 179–181  
 PATH mode, 170, 181–183  
 RAW mode, 169–173  
 SELECT statement, 169  
 XMLDATA mode, 170  
 XMLSCHEMA mode, 170  
 formatDate() **method, 81**  
 formatIndex() **method, 81**  
 format--number **function, 77**  
 formatNumber() **method, 81**  
 formatTime() **method, 81**  
**forward slash (/), 13, 266**  
 from **attribute, XLink, 321**  
 FROM **clause, 89**  
**full outer joins, 109–110**  
**full-text indexes, 259**  
 function--available **function, 77**  
**functions. See also methods**  
   CAST, 147  
   custom-built, 303  
   element--available, 77  
   format--number, 77  
   function--available, 77  
   generate--id, 77  
   instance(), 345  
   interpretation, SOAP messages, 214  
   last(), 276, 290  
   methods versus, 76–77  
   MULTISET, 147  
   node(), 277  
   node--set, 77  
   OPENROWSET, 185  
   OPENXML, 183–184  
   position(), 276, 290  
   POWER, 99  
   property(), 345  
   REC2XML, 437–438  
   ROUND, 96  
   system--property, 77  
   TO\_Char, 93  
   TRIM, 138  
   XMLQuery, 311–312

XMLTable, 311–312  
 XPath scripting language  
   accessor, 282  
   Boolean, 286  
   constructor, 283–284  
   context, 289–290  
   date, 286–287  
   duration, 286–287  
   error, 283  
   list of, 281–282  
   node, 288  
   QName, 287  
   sequence, 288–289  
   string, 285  
   time, 286–287  
   tracing, 283  
   URI, 285  
 xpointer(), 337–338  
 XQuery language, 303  
 XSL, 76–80  
 ZD, 97

## G

generate--id **function, 77**  
**generic access, DOM, 11**  
**genetics, commercial uses of XML, 393**  
**geographical data, commercial uses of XML, 394**  
**geological data, commercial uses of XML, 394**  
**geospatial data, commercial uses of XML, 394**  
 GETNAMESPACE **method, 133**  
 GETROOTELEMENT **method, 133**  
 GETSTRINGVAL **method, 133**  
**global type, XSD, 370**  
**GPS (Global Positioning System), 394**  
**Graphical User Interface (GUI), 161**  
**greater-than XPath numeric operator, 284**  
 GROUP BY **clause**  
   defined, 88  
   HAVING clause, 103  
   SELECT statement, 102–104  
   SQL/XML standard, 144  
**GUI (Graphical User Interface), 161**

## H

### hashing indexes, 257

HAVING clause, GROUP BY clause, 103

header fields, table queries, 91

header section, SOAP, 211

hierarchical array structure, DOM, 12

HTML (Hypertext Markup Language)

discussed, 1

embedding XML in, 7–9

embedding XQuery code into, 299–301

script errors, end of chapter exercise, 27

XML comparison, 2

HTTP protocol, document transfers, 207

HttpRequest object, 50–51

hyperlinks. See XLink

Hypertext Markup Language. See HTML

## I

<IMG> tag, 8

INCITS (International Committee for Information Technology Standards), 134

incomplete data, commercial uses of XML, 393

indexes

algorithms, 200

ALTER INDEX command, 122

binary tree structure, 257

bitmaps, 258

branch section, 257

changes in, 121

context, 259

DROP INDEX command, 122

formatIndex() method, 81

full-text, 259

hashing, 257

leaf section, 257

NXD (native XML database), 258

path, 200

physical start of, 257

pointers, 258

primary XML, 200–201

property, 200

secondary XML, 200–201

side tables, 259

size, 200

sorting order, 200, 258

structural, 259

uses for, 199, 256

value, 200, 259

industry-specific vocabularies, 387, 390

inheritance

defined, 30

Document class and, 38

multiple, 232

object model, 231

inner joins, 108

input types, XForms, 342

Input/Output (I/O)

discussed, 143

storage considerations, 382–383

INSERT command

adding records to tables, 119

defined, 89

subqueries, 120

instance() function, 345

int data type, 372

integer division XPath numeric operator, 284

International Committee for Information Technology Standards (INCITS), 134

Internet Explorer browser

defined, 1

executing XForms in, 341

XLink support, 320

XPointer and, 337

inter-relationships, object model, 226

I/O (Input/Output)

overview, 143

storage considerations, 382–383

IP addresses, URI functions, 285

ISFRAGMENT method, 133

item value, XQuery language, 302

**J****Java virtual machine (JVM), 297****joins**

- cross, 108
- full outer, 109–110
- inner, 108
- JOIN clause
  - ON clause, 106
  - discussed, 88, 107
  - USING clause, 104–106
- left outer, 109
- natural, 107
- outer, 109–110
- right outer, 109–110
- self, 108

**JVM (Java virtual machine), 297****L**label **attribute, XLink, 321, 329****LAN (local area network), 207**language **attribute, 62**last() **function, 276, 290****layered structure, object model, 235–236****leaf**

- content of elements, 18
- leaf section, index, 257

**left outer joins, 109****less-than XPath numeric operator, 284**let **clause, FLWOR statement, 308****links. See XLink****list elements, XSD, 377–378**LOAD() **method, 40, 443****local area network (LAN), 207****local type, XSD, 370**locator **attribute, XLink, 321****locking, node level, 261****logical operators**

- discussed, 93
- OR, 93–94

**logical relationship, sample database, 428**long **data type, 372****lowercase characters, element naming rules, 17****M****manufacturing, commercial uses of XML, 393****many-to-many relationship, object model, 226, 232, 234**match **attribute, 62****MathML (Mathematics XML) vocabulary, 387–389**maxOccurs **value, 374****medicine, commercial uses of XML, 393****memory, 143****merged table records, joins, 104–107****message creation, SOAP, 215–218****metadata**

- change commands, 89
- defined, 19
- document transfers, 206, 210
- NXD (native XML database), 249
- structure, splitting databases by, 322–325

**methods. See also functions**

- absoluteChildNumber(), 81
- ancestorChildNumber(), 81
- appendChild(), 34
- childNumber(), 81
- depth(), 81
- exist(), 169, 189
- EXISTSNODE, 133, 156
- EXTRACT, 133
- formatDate(), 81
- formatIndex(), 81
- formatNumber(), 81
- formatTime(), 81
- functions versus, 76–77
- GETNAMESPACE, 133
- GETROOTELEMENT, 133
- GETSTRINGVAL, 133
- ISFRAGMENT, 133
- LOAD(), 40, 443
- modify(), 169, 190
- nodes(), 169
- object model, 231, 235
- query(), 169, 186
- text(), 157

## methods (continued)

TRANSFORM, 133  
uniqueID(), 81  
value(), 169, 189  
XMLAGG  
  description of, 135, 439  
  disadvantages, 140  
  lines of output, 139  
  sorting order example, 139  
XMLATTRIBUTES  
  description of, 135, 438  
  highlighted name example, 137  
XMLCOLATTVAL  
  description of, 135  
  relational structure form, 140  
XMLCONCAT  
  description of, 135, 439  
  multiple XML fragment example,  
    137–138  
XMLELEMENT  
  description of, 135, 438  
  multiple layers, embedding, 136  
  tag creation, 136  
XMLFOREST  
  description of, 135, 438  
  multiple fields, 141  
XMLNAMESPACE, 439  
XMLTRANSFORM, 135  
XMLType data type, 132–133  
XSL method syntax, 81  
**Microsoft website, 165**  
minOccurs **value, 374**  
**miscellaneous data types, XSD, 373**  
**model description, XForms, 339–340**  
modify() **method, 169, 190**  
**modulus XPath numeric operator, 284**  
**multiple inheritance, 232**  
**multiple path expressions, XPath scripting  
  language, 278**  
**multiple values, attributes with, 21**  
**multiplication XPath numeric operator, 284**  
MULTISET **function, 147**

## N

**name change fields, table queries, 101–102**

NamedNodeMap **class, 35**

**namespace nodes**

  XPath scripting language, 266

  XQuery language, 302

**namespaces**

  defined, 211

  discussed, 24

  prefix, adding to tags, 25–26

  semantic conflict, 25

  SOAP message, 211–213

  XForm, 340–341

**name–value pairs, 14**

**naming rules, elements, 17**

**native XML database. See NXD**

**natural joins, 107**

negativeInteger **data type, 372**

**nested subqueries, 110**

**nested tables, 145**

**Netscape browser, 1**

**news feeds, 389, 393**

Node **class**

  appendChild() method, 34

  attributes and methods of, 33

  functions performed by, 32

  Try It Out exercise, 35–37

node() **function, 277**

NodeList **class**

  attributes and methods of, 34

  functions performed by, 32

**nodes**

  collection, 30

  node level locking, NXD, 261

  pattern matching, 69–70

  relationships between elements, 18

  relationships, XPath scripting language,  
    266–268, 288

  xsl:attribute element, 67

  xsl:copy element, 67–68

  xsl:copy--of element, 68–69

  xsl:element element, 66–67

- `nodes()` **method, 169**
- `node--set` **function, 77**
- `nonNegativeInteger` **data type, 372**
- `nonPositiveInteger()` **data type, 372**
- normalized structure, object model, 226, 231**
- NOT logical operator, 93**
- numbers, formatNumber() method, 81**
- numeric data types, XSD, 372**
- numeric functions, XPath scripting language, 284**
- NXD (native XML database)**
  - concurrency, 261
  - content classification, 260–261
  - data-centric documents, 260–261
  - DBPrism database, 386
  - defined, 248, 433
  - discussed, 245, 384
  - document-centric document, 260
  - end of chapter exercise, 262–263
  - eXist database, 385
  - index creation, 258
  - metadata values, 249
  - multiple-layered document example, 251
  - node level locking, 261
  - reading, 261
  - repository information, 246
  - schema independent, 252–255
  - single-layered document example, 248–249
  - storage in, 261
  - Tamino database, 385
  - TigerLogic database, 385
  - transactional control, 261
  - uses for, 247
  - when to use, 394–395
  - XHive database, 385
  - XIndice database, 386
- O**
- object actions, XForms, 344–345**
- object model**
  - abstraction, 232
  - attributes, 231
  - classes, 231
  - collections, 231
  - data structure, 225–226
  - document comparison, 238
  - end of chapter exercise, 242–243
  - ERDs, 232
  - flat data, 227–229, 237
  - inheritance, 231
  - inter-relationships, 226
  - layered structure, 235–236
  - many-to-many relationship, 226, 232, 234
  - methods, 231, 235
  - normalized structure, 226, 231
  - one-to-many relationship, 226
  - relational structure, 226–230
  - sample database, 430
  - specializations, 232
- objects, DOM**
  - classes versus, 32
  - hierarchical structure, 30–31
  - HTTPRequest, 50–51
- OLAP (Online Application Processing), 104**
- ON clause, JOIN clause, 106**
- one-to-many relationship, object model, 226**
- open standards, XML advantages, 3**
- opening tags**
  - how to use, 2
  - syntax rules, 14
- OPENROWSET function, 185**
- OPENXML function, 183–184**
- operators, XPath scripting language, 275–276**
- optional ([ ]) character, 431**
- OR logical operator, 93–94**
- Oracle database**
  - browser-based control tools, 161
  - document creation
    - content, editing and removing, 159–161
    - database storage, 153–154
    - demographics, 153–154
    - document retrieval, 154–155
    - duplicated records, 142
    - file size considerations, 143

## Oracle database, document creation (continued)

- GROUP BY clause, 144
- highlighted name example, 137
- methods, list of, 135
- multiple fields, 141
- nested tables, 145
- relational structure form, 140
- results, reading, 155–159
- root tag, 145
- sorting order example, 139
- tag creation, 136
- white space removal, 138

end of chapter exercise, 162–163

GUI (Graphical User Interface), 161

packages, 134

PL/SQL package, 151–153

SYS\_XMLGEN function, 146

TYPE objects, 147

user-defined types, 146

XML document creation, 134–135

XMLType data type

- CLOB object, 132
- discussed, 131
- methods, 132–133
- subprograms, 132
- uses for, 132

XQuery language support, 311–312

**ORDER BY clause**

- AS clause in, 101–102
- defined, 88
- FLWOR statement, 307–308
- functionality, 100
- WHERE clause, 102

**order of data, XML advantages, 3**

order--by **attribute**

- xsl:apply--template element, 64
- xsl:for--each element, 71

**outer joins, 109–110**

**P**

**packages, Oracle database, 134**

**parameter entities, DTD, 365–369**

**parameters, xsl:param element, 75**

**parent node, XQuery language, 302**

**parseError class**

- attributes and methods, 47
- code, 48–49

**parser, 5**

**path expressions, XPath scripting language, 272**

**path indexes, 200**

**PATH mode, FOR XML clause, 170, 181–183**

**pattern matching**

- nodes, 69–70
- standard document, 187
- syntax, 81

**performance issues, storage considerations, 382–383**

**personalization, commercial uses of XML, 393**

**physical relationship, sample database, 429**

**pipe (|) character, 278, 431**

**PL/SQL package, 151–153**

**pointers, index, 258**

**position() function, 276, 290**

**positiveInteger data type, 372**

**POWER function, 99**

**precedence**

- in expressions, 98–99
- in SQL, 95–98

**predicate syntax, XPath scripting language, 274–275**

**prefix, adding to tags, 25–26**

**primary XML indexes, 200–201**

**processing instruction**

- ProcessingInstruction class, 52
- XPath scripting language, 266
- XQuery language, 302
- xsl:comment element, 61
- xsl:processing--instruction element, 60
- xsl:script element, 61
- xsl:stylesheet element, 60
- xsl:transform element, 60

**programmatic manipulation, DOM, 29**

**programming complexity, attribute use, 21**

`property()` **function, 345**

**property indexes, 200**

**pseudocode, document creation, 127–128**

## Q

**QName functions, XPath scripting language, 287**

**queries, relational database tables. See also**

### XQuery language

density expression calculation, 102

discussed, 88

field headers, 91

field name changes, 101–102

joins

merged records, 104–107

natural, 107

precedence

in expressions, 98–99

in SQL, 95–98

record sorting order, 100

records, returning from multiple, 115

simple writing example, 89–92

subqueries

AS clause in, 114–115

correlated, 112

defined, 88, 110

INSERT command, 120

nested, 110

record changes, 114–115

sorting order, 114

UPDATE command, 120

uses for, 111

values returned by, 112

summarized records, 102–104

unwanted records, filtering out, 92–95

`query()` **method, 169, 186**

## R

**RAM (Random Access Memory), 143**

**RAW mode, FOR XML clause, 169–173**

**reading NXD, 261**

**read-only property**

accessor functions, 282

readonly property, XForms, 344

**Really Simple Syndication (RSS), 390**

**records, table**

adding, 119

deleting, 119

duplicated, 142

merged, 104–107

sorting order, 100

summarized, 102–104

updating, 119

**REC2XML function, 437–438**

**relational database tables. See tables**

**relational structure, object model, 226–230**

**relationships between elements, 18**

**relative paths, XPath scripting language, 266**

**relevant property, XForms, 344**

**repository information, NXD, 246**

**required property, XForms, 344**

**reserved characters, 24**

**resource attribute, XLink, 321**

**restriction data types, XSD, 376–377**

**return clause, FLWOR statement, 303–305**

**right outer joins, 109–110**

**role attribute, XLink, 321**

**ROLLBACK statement, 121**

**root nodes**

syntax rules, 13

tags, 2

XPath scripting language, 266

**root tag, SQL/XML standard, 145**

**ROUND function, 96**

**RSS (Really Simple Syndication), 390**

## S

**sample database**

discussed, 427

logical relationship, 428

object model, 430

physical relationship, 429

## **Saxon program, 297–299**

## **scalability, XML advantages, 3**

## **scalar expression, 146**

## **schema, XSD**

- annotations, 193–195
- cardinality, 373–374
- complex data types, 378–379
- custom data types, 375
- date data types, 372–373
- defined, 434
- discussed, 167, 351
- end of chapter exercise, 223, 380
- global type, 370
- hierarchical relational database table structure
  - example, 191–192
- list elements, 377–378
- local type, 370
- mapping to tables, 192–193
- miscellaneous data types, 373
- numeric data types, 372
- restriction data types, 376–377
- sequencing, 374–375
- simple data types, 376
- simple type facets, 376
- storing as schema collection, 195–199
- string data types, 371–372
- structures, 371
- substitution, 379–380
- time data types, 372–373
- union list declarations, 378

## **scripts, end of chapter exercise, 28, 55**

## **secondary XML indexes, 200–201**

### **select attribute**

- `xsl:apply-template` element, 64
- `xsl:for-each` element, 71

### **SELECT statement**

- alias in, 92
- defined, 88
- EXTRACT method, 140
- FOR XML clause, 169–170
- FROM clause, 89
- GROUP BY clause, 102–104

### ORDER BY clause

- AS clause in, 101–102
- functionality, 100
- WHERE clause and, 102
- Try It Out exercise, 116–118
- UNION clause, 115
- WHERE clause, 92–95

## **self joins, 108**

## **self-describing data, 246**

## **semantic conflict, namespaces, 25**

## **sequencing**

- functions, XPath scripting language, 288–289
- XSD, 374–375

## **shared components, XQuery language, 296**

## **short data type, 372**

## **sibling node, XQuery language, 303**

## **side tables, index values, 259**

## **simple attribute, XLink, 321**

## **simple data types, XSD, 376**

## **Simple Object Access Protocol. See SOAP**

## **simple type facets, 376**

## **single quote (‘), 93**

## **single root node, syntax rules, 13**

## **size, index, 200**

## **SOAP (Simple Object Access Protocol)**

- body section, 211
- defined, 433
- discussed, 209
- document transfers, 210–214
- end of chapter exercise, 222–223
- envelope section, 211
- function interpretation, 214
- header section, 211
- message creation, 215–218
- namespaces, 211–213
- uses for, 210, 219

## **sorting order**

- index, 200, 258
- subqueries, 114
- table records, 100
- XMLAGG method example, 139
- `xsl:sort` element, 74

**space characters**

- element naming rules, 17
- removal, 138

**specializations, object model, 232****SQL Server database**

- data type storage limitation, 186
- documents, adding, 185–186
- downloading, 165
- end of chapter exercise, 202–203
- OPENXML function, 183–184
- XML data types
  - documents, adding to server, 185–186
  - exist() method, 169, 189
  - indexes, creating, 199–201
  - modify() method, 169, 190
  - modifying, 186–190
  - nodes() method, 169
  - pattern matching expressions, 187
  - query() method, 168, 186
  - retrieving, 186–190
  - string patterns, searching, 186
  - table containing, 168
  - typed, 166–167
  - untyped, 166–167
  - uses for, 167
  - value() method, 169, 189

**SQL (Structured Query Language)**

- defined, 87
- end of chapter exercise, 129–130
- functionality, 88
- precedence in, 95–98
- SELECT statement
  - alias in, 92
  - defined, 88
  - EXTRACT method, 140
  - FOR XML clause, 169–170
  - FROM clause, 89
  - GROUP BY clause, 102–104
  - ORDER BY clause, 100–102
  - Try It Out exercise, 116–118
  - UNION clause, 115
  - WHERE clause, 92–95

## SQL/XML standard

- duplicated records, 142
- file size considerations, 143
- GROUP BY clause, 144
- highlighted name example, 137
- methods, list of, 135
- multiple fields, 141
- nested tables, 145
- relational structure form, 140
- root tag, 145
- sorting order example, 139
- tag creation, 136
- white space removal, 138
- XML document creation
  - discussed, 122
  - multiple-layer document, 125–126
  - program control, 127
  - pseudocode, 127–128
  - root tag, 123
  - two-dimensional structure, 123–124

**sql2000server web server, 82****src attribute**

- <IMG> tag, 8
- <XML> tag, 7

**static data, 13****storage**

- document creation, 153–154
- in NXD, 261
- performance issues, 382–383

**string data types, XSD, 371–372****strings**

- functions, XPath scripting language, 285
- patterns, searching, 186
- verification, EXISTSNOE method, 156

**structural indexes, 259****structural validation, DTD, 12–13****Structured Query Language. See SQL; SQL Server database****structures, XSD, 371****style sheets**

- CSS (cascading style sheets), 4
- references, syntax rules, 13

## style sheets (continued)

XSL (Extensible Style Sheets)

B2B (Business–2–Business) transfers, 220

control structure elements, 70–74

data retrieval elements, 69–70

defined, 3, 58, 434

DOM combined with, 82–84

element functionality, 59

end of chapter exercise, 85–87

function syntax, 77

functions versus method, 76–77

method syntax, 81

node creation elements, 66–69

pattern matching syntax, 81

processing instruction elements, 60–61

transformation elements, 61–65

Try It Out exercise, 78–80

XSLT (Extensible Style Sheet Language

Transformations), 58, 265, 434

**subprograms, XMLType data type, 132**

## subqueries

AS clause in, 114–115

correlated, 112

defined, 88, 110

embedding multiple layers, 111–113

INSERT command, 120

nested, 110

record changes, 114–115

sorting order, 114

UPDATE command, 120

uses for, 111

values returned by, 112

**substitution, 379–380**

**subtraction XPath numeric operator, 284**

**summarized records, table queries, 102–104**

## syntax

Backus–Naur Form notation, 431–432

case sensitivity rules, 14

comment rules, 17

element attributes, 14–15

ending root tag rules, 13

opening and closing tag rules, 14

pattern matching, 81

root node rules, 13

rules, 13–17

style sheet rules, 13

Try It Out exercise, 22–23

XQuery language, 303

XSL method, 81

`system--property` **function, 77**

`SYS_XMLGEN` **function, 146**

## T

### tables

ALTER TABLE command, 89, 122

annotations, 193–195

CREATE TABLE command, 89, 122

DROP TABLE command, 89

ERDs, 104

mapping XSD schema to, 192–193

nested, 145

queries

density expression calculation, 102

discussed, 88

field headers, 91

field name changes, 101–102

joins, 104–108

precedence in expressions, 98–99

precedence in SQL, 95–98

simple writing example, 89–92

subqueries, 110–114

records

adding, 119

deleting, 119

duplicated, 142

merged, 104–107

sorting order, 100

summarized, 102–104

unwanted, filtering out, 92–95

updating, 119

structure of, 88

transactions, 121

tagName **attribute, 42**

**tags**

- closing
  - how to use, 2
  - syntax rules, 14
- creation, XMLELEMENT method, 136
- <IMG>, 8
- opening
  - how to use, 2
  - syntax rules, 14
- prefix, adding, 25–26
- root node
  - how to use, 2
  - syntax rules, 13
- <XML>
  - src attribute, 7
  - syntax rules, 13

**Tamino native XML database, 385****templates**

- xsl:apply--template element, 64–65
- xsl:call--template element, 66
- xsl:template element, 61–63

**Text class**

- attributes and methods, 45–46
- code, 46
- functions performed by, 32

**text() method, 157****text nodes, XQuery language, 302****TigerLogic native XML database, 385****time**

- data types, XSD, 372–373
- formatTime() method, 81
- functions, XPath scripting language, 286–287

**to attribute, XLink, 321****TO\_CHAR function, 93****tracing function, XPath scripting language, 283****transactions**

- control, NXD, 261
- multiple database table changes, 121
- transactional data, 13

**transfers, document**

- basic transfers, 205–207
- B2B (Business–2–Business), 220–222

- external data, 219–222
- HTTP protocol, 208
- metadata, 210
- semantics in, 209–210
- SOAP (Simple Object Access protocol), 210–214
- transformation processing, 208
- Web Services, 207–209

**TRANSFORM method, 133****transformation elements**

- document transfers, 221
  - xsl:apply--templates, 64–65
  - xsl:call--template, 66
  - xsl:template, 61–63

**transformation processing, document transfers, 208****TRIM function, 138****Try It Out exercise**

- data islands, 9–10
- flat data, object model, 239–241
- Node class, 35–37
- SELECT statement, 116–118
- simple document creation, 6–7
- syntax, 22–23
- XPath expressions, 290–292
- XSL functions, 78–80

**TYPE objects, Oracle database, 147****type property, XForms, 344****type-caste, 147****typed XML data type, 166–167****types, DTD attribute, 361–362****U****unary minus or negation XPath numeric operator, 284****unary plus XPath numeric operator, 284****Unicode characters, 26****UNION clause**

- defined, 89
- SELECT statement, 115

**union list declarations, XSD, 378****uniqueID() method, 81****Universal Resource Locator (URL), 285**

`unsignedInt` **data type, 372**  
`unsignedLong` **data type, 372**  
`unsignedOut` **data type, 372**  
`unsignedShort` **data type, 372**  
**untyped XML data type, 166–167**  
**UPDATE command**  
discussed, 89  
subqueries, 120  
table records, 119  
**uppercase characters, element naming rules, 17**  
**URI function, XPath scripting language, 285**  
**URL (Universal Resource Locator), 285**  
**user-defined types, 146**  
**USING clause, JOIN clause, 104–106**

## V

**validation, document, 12–13, 252**  
**value indexes, 200, 259**  
`value()` **method, 169, 189**  
**VLDB (very large databases), 382**  
**vocabularies**  
application-specific, 387  
document example, 391–392  
industry-specific, 387, 390  
MathML (Mathematics XML), 387–389  
news feeds, 389  
overview, 386  
RSS (Really Simple Syndication), 390

## W

**WAN (wide area network), 207**  
**web page**  
determination, end of chapter exercise, 28  
integration, XML advantages, 3  
personalization, commercial uses of XML, 393  
**Web Service Description Language (WSDL), 219–220, 433**  
**Web Services**  
commercial uses of XML, 393  
document transfers, 207–209  
Web Services protocol, 208–209

**websites**  
Microsoft, 165  
W3C (World Wide Web Consortium), 58, 211  
**WHERE clause**  
defined, 88  
FLWOR statement, 306–307  
ORDER BY clause, 102  
SELECT statement, 92–95  
**white space, 138**  
**wide area network (WAN), 207**  
**wildcard (\*) character, 158, 277**  
**WSDL (Web Service Description Language), 219–220, 433**  
**W3C (World Wide Web Consortium)**  
website, 58, 211

## X

**XForms (XML Forms)**  
calculate property, 344  
constraint property, 344  
data types in, 342–343  
defined, 433  
discussed, 315  
end of chapter exercise, 349  
executing in Internet Explorer, 341  
input types, 342  
`instance()` function, 345  
model description, 339–340  
namespaces, 340–341  
object actions, 344–345  
properties, restricting values with, 343–344  
`property()` function, 345  
readonly property, 344  
relevant property, 344  
required property, 344  
type property, 344  
**XHive native XML database, 385**  
**XHTML (Extensible HTML), 433**  
**XInclude tool, 348, 433**  
**XIndex native XML database, 386**

**XLink (XML Linking Language)**

arc attribute, 321, 330  
 arcrole attribute, 321  
 attributes, list of, 320  
 defined, 316, 433  
 discussed, 315  
 end of chapter exercise, 349  
 extended attribute, 321  
 extended links  
   basic XML interpretation, 328  
   metadata structure, splitting databases by, 322–325  
   uses for, 322  
 flexibility in, 332  
 from attribute, 321  
 HTML links versus, 316  
 label attribute, 321, 329  
 links  
   Internet Explorer support, 320  
   refining between multiple documents, 331  
 locator attribute, 321  
 resource attribute, 321  
 role attribute, 321  
 simple attribute, 321  
 simple example of, 317–318  
 standards, 319  
 to attribute, 321

**XML data types, SQL Server database**

documents, adding to server, 185–186  
 exist() method, 169, 189  
 indexes, creating, 199–201  
 modify() method, 169, 190  
 modifying, 186–190  
 nodes() method, 169  
 pattern matching expressions, 187  
 query() method, 168, 186  
 retrieving, 186–190  
 string patterns, searching, 186  
 table containing, 168  
 typed, 166–167  
 untyped, 166–167  
 uses for, 167  
 value() method, 169, 189

**XML DOM (XML Document Object Model), 266, 433****XML Editor, 433****XML Extender software, 436****XML (Extensible Markup Language)**

advantages of, 3  
 commercial uses of, 392–394  
 defined, 1, 434  
 embedding in HTML, 7–9  
 HTML comparison, 2  
 simple document creation, 4–7  
 uses for, 381

**XML Format (XML-F0), 348****XML Forms. See XForms****XML Infoset (XML Information Set), 266****XML Linking Language. See XLink****XML Pointing Language. See XPointer****XML Schema Definition. See XSD****<XML> tag**

src attribute, 7  
 syntax rules, 13

**XML Topic Maps (XTM), 394****XMLAGG method**

description of, 135, 439  
 disadvantages, 140  
 lines of output, 139  
 sorting order example, 139

**XMLATTRIBUTES method**

description of, 135, 438  
 highlighted name example, 137

**XMLCLOB data type, 436****XMLCOLATTVAL method**

description of, 135  
 relational structure form, 140

**XMLCONCAT method**

description of, 135, 439  
 multiple XML fragment example, 137–138

**XMLDATA mode, FOR XML clause, 170****XMLELEMENT method**

description of, 135, 438  
 multiple layers, embedding, 136  
 tag creation, 136

**XMLFILE data type, 436**

## **XML-FO (XML Format), 348**

### **XMLFOREST method**

- description of, 135, 438
- multiple fields, 141

### **XMLNAMESPACE method, 439**

### **XMLQuery function, 311–312**

### **XMLSCHEMA mode, FOR XML clause, 170**

### **XMLSERIALIZE function, 438**

### **XMLTable function, 311–312**

### **XMLTRANSFORM method, 135**

### **XML2CLOB function, 438**

### **XMLType data type**

- CLOB object, 132
- defined, 434
- discussed, 131
- methods, 132–133
- subprograms, 132
- uses for, 132

### **XMLVARCHAR data type, 436**

### **XPath scripting language**

- absolute and relative paths, 266
- axes, 279–280
- binding data, 346–347
- comment nodes, 266
- context functions, 276
- defined, 434
- discussed, 58, 133
- document nodes, 266
- end of chapter exercise, 293–294
- expressions
  - concatenation, 278
  - list of, 273–274
  - multiple path, 278
  - numeric, 284
  - path, 272
  - syntax, 268–271
  - Try It Out exercise, 290–292
  - wildcard (\*) character, 277
- functionality, 266
- functions
  - accessor, 282
  - Boolean, 286

constructor, 283–284

context, 289–290

date, 286–287

duration, 286–287

error, 283

list of, 281–282

node, 288

QName, 287

sequence, 288–289

string, 285

time, 286–287

tracing, 283

URI, 285

namespace nodes, 266

node relationships, 266–268

operators, 275–276

predicate syntax, 274–275

processing instruction nodes, 266

root nodes, 266

### **XPointer (XML Pointing Language)**

defined, 434

discussed, 315

documents, linking to, 335–336

end of chapter exercise, 349

Internet Explorer and, 337

splitting data between documents, 333–334

XPath functionality with, 338

xpointer() function, 337–338

### **XQuery language**

ancestor node, 303

atomic value, 302

attribute nodes, 302

child node, 302

code, embedding into HTML, 299–301

command example, 296–297

comment nodes, 302

defined, 434

descendent node, 303

discussed, 295

document nodes, 302

element nodes, 302

end of chapter exercise, 314

- FLWOR statement
  - embedded for loops and communication, 309–310
  - for loop, 303–305
  - let clause, 308
  - order by clause, 307–308
  - return clause, 303–305
  - where clause, 306–307
- functions in, 303
- item value, 302
- namespace nodes, 302
- Oracle database support, 311–312
- parent node, 302
- processing-instruction nodes, 302
- Saxon program and, 297–299
- shared components, 296
- sibling node, 303
- syntax rules, 303
- text nodes, 302
- uses for, 296
- XQueryX variation, 312–313
- XSD (XML Schema Definition)**
  - annotations, 193–195
  - cardinality, 373–374
  - complex data types, 378–379
  - custom data types, 375
  - date data types, 372–373
  - defined, 434
  - discussed, 167, 351
  - end of chapter exercise, 223, 380
  - global type, 370
  - hierarchical relational database table structure
    - example, 191–192
  - list elements, 377–378
  - local type, 370
  - mapping to tables, 192–193
  - miscellaneous data types, 373
  - numeric data types, 372
  - restriction data types, 376–377
  - sequencing, 374–375
  - simple data types, 376
  - simple type facets, 376
  - storing as schema collection, 195–199
  - string data types, 371–372
  - structures, 371
  - substitution, 379–380
  - time data types, 372–373
  - union list declarations, 378
- XSL (Extensible Style Sheets)**
  - B2B (Business–2–Business) transfers, 220
  - control structure elements, 70–74
  - data retrieval elements, 69–70
  - defined, 3, 58, 434
  - DOM combined with, 82–84
  - element functionality, 59
  - end of chapter exercise, 85–87
  - function syntax, 77
  - functions versus method, 76–77
  - method syntax, 81
  - node creation elements, 66–69
  - pattern matching syntax, 81
  - processing instruction elements, 60–61
  - transformation elements
    - document transfers, 221
    - xsl:apply--templates, 64–65
    - xsl:call--template, 66
    - xsl:template, 61–63
  - xsl:apply--templates **element, 64–65**
  - xsl:attribute **element, 67**
  - xsl:call--template **element, 66**
  - xsl:choose **element, 72–73**
  - xsl:comment **element, 61**
  - xsl:copy **element, 67–68**
  - xsl:copy--of **element, 68–69**
  - xsl:element **element, 66–67**
  - XSL-FO format, 58, 434**
  - xsl:for--each **element, 71**
  - xsl:if **element, 70–71**
  - xsl:otherwise **element, 72–73**
  - xsl:output **element, 70**
  - xsl:param **element, 75**
  - xsl:processing--instruction **element, 60**
  - xsl:script **element, 61**

## xsl:sort element

---

xsl:sort **element, 74**

xsl:stylesheet **element, 60**

**XSLT (Extensible Style Sheet Language Transformations), 58, 265, 434**

xsl:template **element, 61–63**

xsl:text **element, 70**

xsl:transform **element, 60**

xsl:value-of **element, 69–70**

xsl:variable **element, 75**

xsl:when **element, 72–73**

**XTM (XML Topic Maps), 394**

## Y

**YYYY–MM–DD date format, 372**

## Z

**ZD function, 97**















