

Index

SYMBOLS AND NUMERICS

& (ampersand)

- XML reserved character, 224
- XML symbol for, 224

<- ->, XML comments, 225

' (apostrophe), XML symbol for, 224

=== (equal signs), type comparison operator, 64–65

== (equal signs), type conversion operator, 64–65

> (greater than sign), XML, 224

(hash character), comment indicator, 436

< (less than sign), XML, 224

" (quotation marks), XML symbol for, 224

<? ?> (processing instructions), 122–124

1xx Informational responses, 207–208

2xx Success responses, 207–208

3xx Redirection responses, 207–208

4xx Client Error responses, 207–208

5xx Server Error responses, 207–208

302 Found status code, 427

404 Not Found status code, 431–433

A

<abbr> element, 272–273

Abdera Atom toolkit, 374–375

Accessibility...: Problem—Design—Solution, 95

accessibility issues

- aural CSS, 106
- automatic redirects, 107
- automatic refresh, 107
- BuzzWatch, 25–28
- color, 96–99
- dealing with images, 96–99
- deprecated tags, 99
- device independent scripts, 105
- distractions, 104
- frames, 106
- headers, 107–108
- inline styles, 99
- labels on form inputs, 105
- legislation, 95
- links, 104
- Olympic Games case study, 95
- pop-up windows, 107
- previewing pages, 108
- separating content from presentation, 99–103
- sidebars, 107–108
- tables, 99–103
- text emphasis, 99–103
- user-friendly URLs, 104
- W3C recommendations, 95

Adleman, Len, 469

Adobe SVG plug-in, 141

ADO.NET, converting relational data to XML, 343–348

adr & geo microformats, 271–272

Advanced Systems Format (ASF), 449

aggregation

- BuzzWatch, 2
- feeds, 412

AIFF (Audio Interchange File Format), 448

Ajax (Asynchronous JavaScript and XML). See also JavaScript.

- definition, 70
- displaying XML information on a page, 71–75
- extracting information from XML files, 71–75
- Google Web Toolkit, 78–79
- HTTP queries, 71–75
- loading XML files, 71–75
- memory leaks, 79–81
- technologies, 70–71
- user experience, 71
- XMLHttpRequest object
 - example, 72–75
 - history, 71–72
- YUI (Yahoo! UI) Library, 75–78
- YUIConnectionManager object, 77–78

aligning text, 128

alternate value, 260

Amazon.com, 294, 406

& (ampersand) entity, 224

ampersand (&), XML, 224

animation, 136–137

Apache

- Abdera Atom toolkit, 374–375
- authentication, 466–467
- authorization, 466–467
- SSL (Secure Sockets Layer)
 - certificate request, generating on Unix, 476–477
 - certificate request, submitting in IIS, 477
 - configuring the site, 478
 - issued certificates. installing, 477–479
 - setting up, 474–478
- Web server, 208

APIs

DOM

- online resources, 51
- overview, 49–50
- reference, 51
- Flickr SOAP, 295–299
- Google Maps, 417–419
- JAXP (Java API for XML Processing), 315–317
- serialization, 315–317
- XML, 235

' (apostrophe) entity, 224

apostrophe (‘), XML symbol for, 224

APP (Atom Publishing Protocol), 235, 291

application architecture, 374

application layer, 195

“Architectural Styles and the Design of Network-based Software Architectures,” 194

Architecture of the World Wide Web, 194

articles. See books and publications.

ASF (Advanced Systems Format), 449

ASP.NET

- authentication
 - configuring, 466
 - overview, 462–464
- authorization
 - configuring, 466
 - overview, 462–464
- dynamic content (XML), serving over HTTP, 317–320
- HTTP redirection, 433
- serving XML dynamic content over HTTP, 317–320

Asynchronous JavaScript and XML (Ajax). See Ajax (Asynchronous JavaScript and XML).

Atom format

- alternate value, 260
- description, 258–260
- enclosure value, 260
- enclosures, 262–263
- history of, 257
- href attribute, 260
- hreflang attribute, 260

length attribute, 260
 <link> element, 260
 podcasting, 262–263
 rel attribute, 260
 related value, 260
 self value, 260
 specifications, 261
 title attribute, 260
 type attribute, 260
 via value, 260

Atom Publishing Protocol (APP), 235, 291

Atom service for syndication channels

Apache Abdera toolkit, 374–375
 application architecture, 374
 code overview
 AtomServlet.java class, 375, 377–383
 clearFeed method, 384
 clearing the feed, 376
 compare method, 384–385
 core files, 375
 createEntry method, 387
 createFeed method, 385–386
 isEntryInFeed method, 387
 loadFeed method, 386
 MailClient.java class, 375
 MailEntryCollector.java class, 375
 MessageReader.java class, 375
 Settings.java class, 375, 388
 sortEntries method, 384
 Utils.java class, 375, 383–388
 writeTo method, 386–387
 e-mail support
 MailClient.java class, 393–396
 MailEntryCollector implementation,
 396–400
 MessageReader interface, 396
 readMessage method, 399–400
 feed reader, 373
 user interface, 372–373

Atom Syndication Format, 235

AtomServlet.java class, 375, 377–383

attributes, XML, 223–224

Audio Interchange File Format (AIFF), 448

Audio Video Interleave (AVI), 448–449

aural CSS, 106

authentication

in Apache, 466–467
 in ASP.NET
 configuring, 466
 overview, 462–464
 custom, web services, 487–489
 definition, 459–460
 HTTP type, 466–467
 in IIS
 configuring, 464–465
 overview, 462–464
 providers, 466–467

authority section, URIs, 197

authorization

in Apache, 466–467
 in ASP.NET
 configuring, 466
 overview, 462–464
 definition, 460
 in IIS
 configuring, 464–465
 overview, 462–464
 providers, 466–467

autodiscovery, 251

AVI (Audio Video Interleave), 448–449

B

badges, 416–417, 418

Berners-Lee, Tim

“Cool URIs don’t change,” 198
 web browsers, 194
 and the WWW (World Wide Web), 168

binary data, converting to XML

automatically, 363–370
 manually, 358–363
 from Microsoft Word, 358
 overview, 229–230
 viewing raw data, 361

binding data to external source, 188–190

blind users. See accessibility issues.

Blogs channel, 274

BMP (Basic Multilingual Plane), 227

bookmarks

design issues, 84–88

sharing, 293

books and publications

Accessibility...: Problem—Design—Solution, 95

“Architectural Styles and the Design of Network-based Software Architectures,” 194

Architecture of the World Wide Web, 194

“Cool URIs don’t change,” 198

HTTP (Hypertext Transfer Protocol) specifications, 194

URI (Uniform Resource Identifier) specification, 194

Web Content Accessibility Guidelines, 95

BPE4LAWS. See WS-BPEL.

broken links, 421. See also URIs (Uniform Resource Identifiers), maintenance.

browser-based applications. See XAML (eXtensible Application Markup Language).

browsers. See also rich clients.

behavior, modifying with XUL, 182–184

directing to content, XUL, 180–182

drawbacks, 168

security issues, 168

WHATWG (Web Hypertext Application Technology Working Group), 142–143

browsing source code, 174–175

business card example, 208–212

`<button>` element

OpenLaszlo, 173

XUL, 181

buttons

OpenLaszlo, 173

XForms, 145

Buy.com, 406

BuzzWatch

aggregation, 2

application tasks, 9–10

client side, 4–5

database query, 13–20

document presentation, 1–12

internals, examining

HTTP headers, 7–9

scripts, 7–9

web server log, 6–7

libraries, 4–5

maintainability, 28–33

menu bar

creating, 10–12

populating, 12

structure of, 2

microformats, 29–30

pages

illustration, 3

structure of, 2

proxies, 20

query strings, 18

schemas, 23–24

scripting languages, 5–6, 33–35

server side, 5

syndication formats, 5

technical architecture, 4–6

technologies, 5

transforming XML into (X)HTML, 30–33

URL space, defining, 34–35

watches, 13–20

web accessibility, 25–28

web server interaction, 12–24

XHTML, semantic information, 29–33

C

C#, serving XML dynamic content over HTTP, 317–320

c (Cubic Bézier curve) command, 135

Caesar shift encryption, 467–468

callbacks, limiting, 92–93

calls, REST services, 291

`<canvas>` element, 172–173

Cascading Style Sheets (CSS). See CSS (Cascading Style Sheets).

case management example, 288–292

categories of mashups, 402, 405**CDATA sections, 230****change management, URIs, 425–428****channels**

Blogs, 274

microformats, 274

channels, syndication

Atom service

Apache Abdera toolkit, 374–375

application architecture, 374

feed reader, 373

user interface, 372–373

Atom service, code overview

AtomServlet.java class, 375, 377–383

clearFeed method, 384

clearing the feed, 376

compare method, 384–385

core files, 375

createEntry method, 387

createFeed method, 385–386

isEntryInFeed method, 387

loadFeed method, 386

MailClient.java class, 375

MailEntryCollector.java class, 375

MessageReader.java class, 375

Settings.java class, 375, 388

sortEntries method, 384

Utils.java class, 375, 383–388

writeTo method, 386–387

Atom service, e-mail support

MailClient.java class, 393–396

 MailEntryCollector implementation,
 396–400

MessageReader interface, 396

readMessage method, 399–400

compiling, 388–389

deploying, 388–389

e-mail for

configuring accounts, 392–393

local mail server, 391–393

mail support architecture, 391

MailClient.java class, 393–396

 MailEntryCollector implementation,
 396–400

MessageReader interface, 396

overview, 389–390

readMessage method, 399–400

going live, 389

running the application, 388–389

character data, XML, 224–225**character encoding**

serving XML over HTTP, 310

XHTML, 46–47

XML, 227–229

classified ads, microformats, 272**clearFeed method, 384****Client Error responses, 207–208****client side applications, 4–5****client-server pattern, 195****client-side transformation, 343–348****Close path (z) command, 135****code points, XML, 227****code security**

data overflow, 482

definition, 460

general principles, 479–482

SQL injection, 479–481

XSS (cross-site scripting), 481–482

collections, URI maintenance, 423–424**color**

accessibility issues, 96–99

SVG, 140

combining multiple requests, 91–92**community microformats, 273–274****compare method, 384–385****compile options, OpenLaszlo, 176****compiling syndication channels, 388–389****compound microformats, 271–272****concise communication, 90–91****confidentiality, 460****CONNECT method, 204****container multimedia formats, 447–454****content, separating from presentation, 99–103****content encoding, RSS 2.0, 250**

content microformats, 269
content negotiation, 214–215, 433
content tagging microformats, 271
content types, 205
contexts, URI maintenance, 428
converting non-XML databases
 binary data to XML
 automatically, 363–370
 manually, 358–363
 from Microsoft Word, 358
 viewing raw data, 361
 overview, 333–334
 relational data to XML
 with ADO.NET, 343–348
 client-side transformation, 343–348
 controlling XML format, 340–343
 with Oracle XSQL, 348–358
 overview, 334–339
 querystring parameters, 353–354
 reformatting date, 353
 renaming `<ROWSET>` and `<ROW>` elements, 352
 renaming XML elements, 351–352
 retrieving data as XML, 339–340
 server-side transformation, 339–343
 with SQL Server, 339–343
 XSLT transformation, 354–358
cookies, REST, 203
“Cool URIs don’t change,” 198
coverage microformats, 271–273
`createEntry` **method, 387**
`createFeed` **method, 385–386**
cross-site scripting (XSS), 481–482
CSS (Cascading Style Sheets)
 aural, 106
 creating pages, 40–41
 for the hearing impaired, 106
 and microformats, 267–268
 rounding corners, 52–53
 tabs, 54–55
 uses for, 51
 and XHTML, 44
CSS Zen Garden, 269

Cubic Bézier curve (c) command, 135
cURL command-line HTTP client, 208
`currentTarget` **property, 140**
CVs, microformats, 272

D

data formats, XML, 235
data microformats, 269
data templates, XAML, 190
databases
 querying, BuzzWatch example, 13–20
 XML, 328–331
databases, non-XML
 converting binary data to XML
 automatically, 363–370
 manually, 358–363
 from Microsoft Word, 358
 viewing raw data, 361
 converting relational data to XML
 with ADO.NET, 343–348
 client-side transformation, 343–348
 controlling XML format, 340–343
 with Oracle XSQL, 348–358
 overview, 334–339
 querystring parameters, 353–354
 reformatting date, 353
 renaming `<ROWSET>` and `<ROW>` elements, 352
 renaming XML elements, 351–352
 retrieving data as XML, 339–340
 server-side transformation, 339–343
 with SQL Server, 339–343
 XSLT transformation, 354–358
 overview, 333–334
`datapath` **attribute, 178**
`<dataset>` **element, 178**
`<DataTemplate>` **element, 190**
datatypes, XML, 231–232
date, reformatting, 353
deaf users. See accessibility issues.
declaration, XML, 225
declarative animation, 137–139

decryption, 468**DELETE method, 204****del.icio.us, 293****deploying**

feeds

- through external sources, 415
- on your server, 414

OpenLaszlo, 176

syndication channels, 388–389

descriptive multimedia formats, 436–447**design issues**

accessibility

- aural CSS, 106

- automatic redirects, 107

- automatic refresh, 107

- color, 96–99

- dealing with images, 96–99

- deprecated tags, 99

- device independent scripts, 105

- distractions, 104

- frames, 106

- headers, 107–108

- inline styles, 99

- labels on form inputs, 105

- legislation, 95

- links, 104

- pop-up windows, 107

- previewing pages, 108

- separating content from presentation, 99–103

- sidebars, 107–108

- tables, 99–103

- text emphasis, 99–103

- user-friendly URLs, 104

- W3C recommendations, 95

bookmarks, 84–88

down-level device support, 93–94

minimizing traffic

- combining multiple requests, 91–92

- concise communication, 90–91

- excessive network traffic, 92

- limiting callbacks, 92–93

- processing on the server, 92

- removing error correction, 92

- spying on the user, 93

- too much feedback, 92

- navigation, 88–89

design pattern microformats, 272–273**desktop-based applications. See XAML (eXtensible Application Markup Language).****DevBoi tool, 51****developer aids, OpenLaszlo, 175–179****device independent scripts, accessibility issues, 105****digital certificates, 469–470****directory inclusion microformats, 271****disambiguation microformats, 269–271****distractions, accessibility issues, 104****Document Type Definition (DTD), 230****documents**

- microformat, 273

- presentation, 1–12

Dodds, Leigh, 294**DOM (Document Object Model)**

API

- online resources, 51

- overview, 49–50

- reference, 51

definition, 47

levels, 49

minimizing updates, 70

structure of, 47–49

domain names, Web 1.0, 196**domain-specific mini languages, microformats, 271–272****down-level device support, design issues, 93–94****drawing a rectangle, 127****DTD (Document Type Definition), 230****dynamic content (XML), serving over HTTP**

- with ASP.NET, 317–320

- with C#, 317–320

- example, 323–324

- JAXP (Java API for XML Processing), 315–317

dynamic content (XML), serving over HTTP

(continued)

- JSP, 314–315
- life cycle, 312
- with Perl, 322
- with PHP, 322
- principles, 312
- with Python, 322
- with Ruby on Rails, 321–322
- serialization APIs, 315–317
- server-side XSLT, 322–324
- servlets, 313–314
- XML pipeline language, 325–328
- XML platform, 325–328
- XML with Java, 313–317

dynamic typing, 113

E

E (Elliptical curve) command, 135

early security incorporation, 461

ebXML (Electronic Business XML), 302

elements

- syndication, cross-format comparisons, 261–262
- XML, 222–223

e-mail channel, microformats, 274

e-mail for syndication channels

- configuring accounts, 392–393
- local mail server, 391–393
- mail support architecture, 391
- MailClient.java class, 393–396
- MailEntryCollector implementation, 396–400
- MessageReader interface, 396
- overview, 389–390
- readMessage method, 399–400

empty elements, 45–46

enclosure value, 260

enclosures, 262–263

encoding multimedia formats, 447–454

entities, XML, 224, 230

envelope opacity, SOAP, 303–304

equal signs (==), type conversion operator, 64–65

equal signs (===), type comparison operator, 64–65

error checking, JavaScript functions, 68

error correction, removing, 92

escape characters, XML, 224–225

event handling, OpenLaszlo, 173

events microformats, 272

events timeline example, 275–285

examples, BuzzWatch

- aggregation, 2
- application tasks, 9–10
- client side, 4–5
- database query, 13–20
- document presentation, 1–12
- internals, examining
 - HTTP headers, 7–9
 - scripts, 7–9
 - web server log, 6–7
- libraries, 4–5
- maintainability, 28–33
- menu bar
 - creating, 10–12
 - populating, 12
 - structure of, 2
- microformats, 29–30
- pages
 - illustration, 3
 - structure of, 2
- proxies, 20
- query strings, 18
- schemas, 23–24
- scripting languages, 5–6, 33–35
- server side, 5
- syndication formats, 5
- technical architecture, 4–6
- technologies, 5
- transforming XML into (X)HTML, 30–33
- URL space, defining, 34–35

watches, 13–20
 web accessibility, 25–28
 web server interaction, 12–24
 XHTML, semantic information, 29–33

examples, “Hello World”

SVG, 126–130
 XForms, 143–147
 XSLT, 112–114

excessive network traffic, 92

eXist, 329–330

extensibility

RSS 2.0, 250
 XHTML, 44–45

eXtensible Application Markup Language (XAML). See XAML (eXtensible Application Markup Language).

eXtensible Markup Language (XML). See XML.

F

feed reader, 373

feedback, excessive, 92

feeds

aggregation, 412
 creating, 411–414
 definition, 244–245
 deploying
 through external sources, 415
 on your server, 414

Fielding, Roy T., 194, 200

5xx Server Error responses, 207–208

FLAC codec, 452

Flickr SOAP API, 295–299

FLV (Flash video), 442

font attributes, SVG, 128

for/against/abstain link, microformats, 271

for-in iterator, 65–66

form labels, accessibility issues, 105. See also XForms.

4xx Client Error responses, 207–208

404 Not Found status code, 431–433

fragment section, URIs, 198

frames, accessibility issues, 106

frameworks, rich clients

OpenLaszlo, 170–179
 XAML, 186–190
 XUL, 179–186

functions, JavaScript

arguments, 67–68
 error checking, 68
 overloading, 67–68
 programming, 66–67

future-proofing URIs, 422–425

G

Garrett, James, 70

GET method, 204

Gonze, Lucas, 293, 294

Google Maps API, 417–419

Google Web Toolkit, 78–79

GRDDL (Gleaning Descriptions from Dialects of Languages), 277

greater than sign (>), XML, 224

> (greater than) entity, 224

H

H (Horizontal line) command, 135

handicapped accessibility

aural CSS, 106
 automatic redirects, 107
 automatic refresh, 107
 BuzzWatch, 25–28
 color, 96–99
 dealing with images, 96–99
 deprecated tags, 99
 device independent scripts, 105
 distractions, 104
 frames, 106
 headers, 107–108
 inline styles, 99
 labels on form inputs, 105
 legislation, 95

handicapped accessibility (continued)

- links, 104
- Olympic Games case study, 95
- pop-up windows, 107
- previewing pages, 108
- separating content from presentation, 99–103
- sidebars, 107–108
- tables, 99–103
- text emphasis, 99–103
- user-friendly URLs, 104
- W3C recommendations, 95

`<handler>` **element**, 173

hash character (#), comment indicator, 436

hAtom, 271–272

hCalendar, 272

hCard, 272

HEAD method, 204

headers, accessibility issues, 107–108

hearing impaired users. See accessibility issues.

“Hello World” examples. See also BuzzWatch.

- SVG, 126–130

- XForms, 143–147

- XSLT, 112–114

hFolk microformats, 272

hidden code, dangers of, 199

hierarchies, URI, 423–424

history of, 246–247

hListing microformats, 272

Horizontal line (H) command, 135

href attribute, 260

hreflang attribute, 260

hResume microformats, 272

hReview microformats, 272

HTML (HypertextMarkup Language)

- alternative to XML, 238–240

- empty elements, 45–46

- future of, 162–164

- history of, 161–162

- IDs, 46

- names, 46

- profiles, microformats, 270

- semantic, 269

- validity, 38–40

- W3C proposals, 162–163

- WHATWG proposals, 163–164

- versus* XHTML, 44–47

- XHTML 2.0 *versus* HTML 5, 164

HTTP

- CONNECT method**, 204

- DELETE method**, 204

- deleting resources, 204

- example, 209–212

- GET method**, 204

- HEAD method**, 204

- headers, BuzzWatch, 7–9

- methods, 204

- OPTIONS method**, 204

- POST method**, 204

- posting to the server, 204

- proxies, 204

- PUT method**, 204

- queries, 71–75

- redirection

 - ASP.NET, 433

 - basics, 425–426

 - content negotiation, 433

 - deprecated methods, 427–428

 - IIS, 433

 - intermediate information page, 427

 - path to query parameters, 431

 - permanent, 426, 430

 - printable shortcuts, 427

 - proxying, 431–433

 - redirect after **POST** pattern, 426

 - server-side, 428

 - temporary, 426–427, 430

 - trailing slashes, 430–431

 - user-friendly shortcuts, 427

- reference toolkit, 208

- requesting resources, 204

- returning resource headers, 204

services, building

- 1xx Informational responses, 207–208
- 2xx Success responses, 207–208
- 3xx Redirection responses, 207–208
- 4xx Client Error responses, 207–208
- 5xx Server Error responses, 207–208

content types, 205

media types, 205

MIME type, 205

representations required, 206–207

resources of interest, 205–206

server response, 207–208

specifications, 194

streaming protocol, 455–456

supported methods, listing, 204

TRACE method, 204

tracing requests, 204

uploading resource representations, 204

HTTP, serving XML

- character encoding, 310
- dynamic content
 - with ASP.NET, 317–320
 - with C#, 317–320
- example, 323–324
- JAXP (Java API for XML Processing), 315–317
- JSP, 314–315
- life cycle, 312
- with Perl, 322
- with PHP, 322
- principles, 312
- with Python, 322
- with Ruby on Rails, 321–322
- serialization APIs, 315–317
- server-side XSLT, 322–324
- servlets, 313–314
- XML pipeline language, 325–328
- XML platform, 325–328
- XML with Java, 313–317
- with eXist, 329–330
- versus HTML, 309–310
- JSON, 332
- media types, 309

- static content, 310–311
- Subversion, 331
- version control, 331
- WebDAV, 331
- well-formed XML, 310
- XML databases, 328–331
- XQuery, 328–331

HTTP 404 Not Found **message**, 194

HTTPTracer graphic HTTP monitor, 208, 215–219

Hypertext Transfer Protocol (HTTP). See **HTTP**.

HypertextMarkup Language (HTML). See **HTML**.

I

IDs, 46

IIS (Internet Information Server)

- authentication
 - configuring, 464–465
 - overview, 462–464
- authorization
 - configuring, 464–465
 - overview, 462–464
- HTTP redirection, 433
- SSL (Secure Sockets Layer)
 - certificate request, submitting, 477
 - certificate requests, generating, 472–473
 - certificate requests, submitting, 473
 - configuring the site, 474–475
 - issued certificates, installing, 474
 - setting up, 471–475

images, accessibility issues, 96–99

Informational responses, 207–208

inline styles, accessibility issues, 99

instructions versus literals, 113

integrity, 460

Internet, versus World Wide Web, 168

IRC channel microformats, 274

`isEntryInFeed` **method**, 387

issues of design. See **design issues**.

`ItemSource` **attribute**, 190

iterating with JavaScript, 65–66

J

Java servlets, URI maintenance, 428–429

JavaScript. See also Ajax.

`==` operator, 64–65

`===` operator, 64–65

alternative to XML, 238

`for-in` iterator, 65–66

functions

arguments, 67–68

error checking, 68

overloading, 67–68

programming, 66–67

iterating, 65–66

optimization

minimizing DOM updates, 70

reducing download time, 69–70

type conversion, 64–65

`typeof` operator, 63

typing, 62–64

variables

undefined, 62–64

uninitialized *versus* undeclared, 63

JAXP (Java API for XML Processing), 315–317

journal articles. See books and publications.

JSON (JavaScript Object Notation)

alternative to XML, 236–238

serving XML over HTTP, 332

JSP (Java Server Pages), 314–315

L

L (Line) command, 135

labels on forms, accessibility issues, 105

layered protocols, 195

layered security approach, 461–462

layout elements, OpenLaszlo, 172–175

legislation, accessibility issues, 95

`length` attribute, 260

`lengthAdjust` attribute, 129

less than sign (<), XML, 224

libraries, BuzzWatch, 4–5

license links, microformats, 271

life cycle, 312

limiting callbacks, 92–93

`<link>` element, 260

link tagging, microformats, 272

links, accessibility issues, 104

list boxes, XAML, 190

`<ListBox>` element, 190

lists, microformats, 272

`loadFeed` method, 386

localization, 184–186

location independence, 424–425

locations, microformats, 271–272

< (less than) entity, 224

.lzx file extension, 170

M

M (Moveto) command, 134

M3U format, 436–438

magazine articles. See books and publications.

`MailClient.java` class, 375, 393–396

`MailEntryCollector` implementation,
396–400

`MailEntryCollector.java` class, 375

maintainability, 28–33

manifest files, 180–182

mapping

badges, 416–417, 418

Google Maps API, 417–419

mashups, 402–404

overview, 415–416

tools for URI maintenance, 428–433

maps, on web pages, 417–419

mashups

business model, 406–407

categories, 402, 405

definition, 401

examples, 402–404. *See also* BuzzWatch.

mapping, 402–404

messaging, 402

mobile, 402

movies, 402

- search, 402
- shopping, 402
- sports, 402
- uses for, 404–405
- media types, 205, 309**
- memory leaks, 79–81**
- menu bar**
 - creating, 10–12
 - populating, 12
 - structure of, 2
- MessageReader **interface, 396**
- MessageReader.java **class, 375**
- messages**
 - digests, 469
 - encryption, 467–469
 - security, 460
 - watching, 215–219
- messaging mashups, 402**
- methods, HTTP, 204**
- microformat documents**
 - application architecture, 275–276
 - code overview, 277–281
 - conversion, 277
 - creating, 275–285
 - events timeline example, 275–285
 - GRDDL (Gleaning Descriptions from Dialects of Languages), 277
 - options, 275–276
 - query, 278–280
 - RDF Query Language, 278–280
 - RDF store, 277
 - rendering, 280–281
 - source code, 281–285
 - SPARQL Protocol, 278–280
 - tidying, 278
 - Timeline application, 280–281
 - Turtle syntax, 279
- microformats**
 - <abbr> element, 272–273
 - adr & geo, 271–272
 - basics, 266–275
 - Blogs channel, 274
 - BuzzWatch, 29–30
 - classified ads, 272
 - community, 273–274
 - composite documents, 273
 - compound, 271–272
 - content, 269
 - content tagging, 271
 - coverage, 271–273
 - and CSS, 267–268
 - CVs, 272
 - data, 269
 - design patterns, 272–273
 - directory inclusion, 271
 - disambiguation, 269–271
 - domain-specific mini languages, 271–272
 - e-mail channel, 274
 - events, 272
 - for/against/abstain link, 271
 - hAtom, 271–272
 - hCalendar, 272
 - hCard, 272
 - hFolk, 272
 - hListing, 272
 - hResume, 272
 - hReview, 272
 - HTML profiles, 270
 - IRC channel, 274
 - key features, 266
 - license links, 271
 - link tagging, 272
 - lists, 272
 - locations, 271–272
 - organizations, 272
 - outlines, 272
 - people, 272
 - presentation, 269
 - primary channels, 274
 - principles, 274
 - process, 274–275
 - relational, 271
 - rel-directory, 271
 - rel-license, 271

microformats (continued)

- rel-payment, 271
- rel-tag, 271
- resumes, 272
- reviews, 272
- semantic HTML, 269
- social networks, 271
- standards.org 2.0, 273–275
- syndication, 271–272
- tip jars, 271
- VotELinks, 271
- Wiki channel, 274
- XMDP profiles, 270–271
- XOXO, 272
- XPN, 271

MIME type, 205

MIME types, 250–251

mobile devices, XHTML, 44

mobile mashups, 402

`mod_alias`, 428–433

`mod_negotiation`, 433

`mod_rewrite`, 428–433

Moveto (M) command, 134

movie mashups, 402

movies. See multimedia.

MPEG formats, 449–451

MSDN, 51

Multicast protocol, 455

multimedia

- AIFF format, 448
- ASF format, 449
- AVI format, 448–449
- container formats, 447–454
- descriptive formats, 436–447
- encoding formats, 447–454
- FLAC, 452
- FLV format, 442
- HTTP streaming protocol, 455–456
- M3U format, 436–438
- MPEG formats, 449–451
- Multicast protocol, 455
- Ogg format, 452

- PLS format, 438
- podcasts, 440–443
- progressive download protocol, 455–456
- protocols, 454–456
- pseudo-streaming protocol, 455–456
- QuickTime, 442
- RDT, 455
- RTMP, 455
- RTSP, 455
- SMIL format, 443–446
- Speex, 452
- Tarkin, 452
- Theora format, 452
- Unicast protocol, 455–456
- Vorbis format, 452
- WAV format, 448
- Writ, 452
- Xiph.org Foundation, 452
- XSPF format, 438

multiple representations, 212–214

music. See multimedia.

N

names, 46

namespaced elements, 172

namespaces

- case sensitivity, 181
- WS-*, 298
- XML, 225–227
- XSLT, 113
- XUL, 181

navigation, design issues, 88–89

network layer, 195

network traffic, excessive, 92

non-XML databases, converting

- binary data to XML
 - automatically, 363–370
 - manually, 358–363
 - from Microsoft Word, 358
 - viewing raw data, 361
- overview, 333–334

relational data to XML
 with ADO.NET, 343–348
 client-side transformation, 343–348
 controlling XML format, 340–343
 with Oracle XSQL, 348–358
 overview, 334–339
 querystring parameters, 353–354
 reformatting date, 353
 renaming `<ROWSET>` and `<ROW>` elements, 352
 renaming XML elements, 351–352
 retrieving data as XML, 339–340
 server-side transformation, 339–343
 with SQL Server, 339–343
 XSLT transformation, 354–358

O

Ogg multimedia format, 452

Olympic Games case study, 95

1xx Informational responses, 207–208

one-click subscription, 250–251

online resources

DevBoi tool, 51
 DOM API, 51
 JSON, 332
 mashups, 402
 MSDN, 51
 PyXML package, 322
 quick access, 59
 QuirksMode, 51
 REST services, 294
 W3C DOM specifications, 51
 XML Reference Guide, 322
 XUL, 179
 ZVON.org, 51

`onLoad` **function, 183**

`onMenuItemCommand` **function, 183–184**

OpenLaszlo. See also XAML; XUL.

basics, 170–175
 browsing source code, 174–175
`<button>` element, 173
 buttons, 173

`<canvas>` element, 172–173
 compile options, 176
`datapath` attribute, 178
`<dataset>` element, 178
 deployment options, 176
 description, 170
 developer aids, 175–179
 drawbacks, 179
 event handling, 173
`<handler>` element, 173
 layout elements, 172–175
 .lzx file extension, 170
 namespaced elements, 172
 populating a page, example, 176–179
`<simplelayout>` element, 172–173
 spacing elements, 172
`<text>` element, 173–174
 text elements, 173–174
`<view>` element, 178
 viewing source code, 176

optimization, JavaScript, 69–70

`OPTIONS` **method, 204**

Oracle XSQL

converting relational data to XML, 348–358
 required downloads, 349
 setting up, 348–350
 XSQL page, examples, 350–358

organizations, microformats, 272

outlines, microformats, 272

overloading functions, JavaScript, 67–68

P

pages. See also CSS (Cascading Style Sheets).

BuzzWatch
 illustration, 3
 structure of, 2
 creating
 choosing elements, 41–42
 CSS (Cascading Style Sheets), 40–41
 HTML *versus* XHTML, 42–47
 valid HTML, 38–40

pages (continued)

development tools, 55–59
displaying XML information with Ajax, 71–75
previewing for accessibility issues, 108
rounding corners, 52–53
tabs, 54–55

papers. See books and publications.

path section, URIs, 198

path to query parameters, 431

`pathSegList` property, 140

people, microformats, 272

Perl, 322

PHP, 322

physical link layer, 195

PIs (Processing Instructions), 122–124, 230

PKI (public key infrastructure), 468–469

playlists, 292–294. See also multimedia.

PLS multimedia format, 438

podcasting, 262–263

podcasts

AIFF format, 448
ASF format, 449
Atom playlist, 440–443
AVI format, 448–449
container formats, 447–454
descriptive formats, 436–447
encoding formats, 447–454
FLAC, 452
FLV format, 442
HTTP streaming protocol, 455–456
M3U format, 436–438
MPEG formats, 449–451
Multicast protocol, 455
Ogg format, 452
PLS format, 438
podcasts, 440–443
progressive download protocol, 455–456
protocols, 454–456
pseudo-streaming protocol, 455–456
QuickTime, 442
RDT, 455
RTMP, 455

RTSP, 455

SMIL format, 443–446

Speex, 452

Tarkin, 452

Theora format, 452

Unicast protocol, 455–456

Vorbis format, 452

WAV format, 448

Writ, 452

Xiph.org Foundation, 452

XSPF format, 438

populating a page, OpenLaszlo, 176–179

pop-up windows, accessibility issues, 107

`POST` method, 204

posting data to the server, 204

Prescod, Paul, 294

presentation microformats, 269

previewing pages, accessibility issues, 108

principles of design. See design issues.

Processing Instructions (PIs), 122–124, 230

processing on the server, 92

progressive download protocol, 455–456

protocols

multimedia, 454–456

XML, 235

proxies

BuzzWatch, 20

HTTP, 204

proxying HTTP redirection, 431–433

pseudo-streaming protocol, 455–456

public key infrastructure (PKI), 468–469

publications. See books and publications.

`PUT` method, 204

Python, 322

Q

Q (Quadratic Bézier curve) command, 134–135

query languages, 234

query section, URIs, 198

query strings, BuzzWatch, 18

querying microformat documents, 278–280

querystring parameters, 353–354

QuickTime, 442
QuirksMode, 51
 " (quote marks) entity, 224
 quotation marks (“), XML symbol for, 224
 quote marks (") entity, 224

R

RDF (Resource Description Framework), 251–256
RDF Query Language, 278–280
RDF store, 277
RDT (Real Data Transport), 455
readMessage method, 399–400
Real Time Messaging Protocol (RTMP), 455
Real Time Streaming Protocol (RTSP), 455
Really Simple Syndication (RSS). *See* **RSS (Really Simple Syndication)**.
rectangle, drawing, 127
redirect after POST pattern, 426
Redirection responses, 207–208
redirects, automatic, 107
Reed, David, 454
reference toolkit, 208
reformatting date, 353
refresh, automatic, 107
rel attribute, 260
related value, 260
relational data, converting to XML
 with ADO.NET, 343–348
 client-side transformation, 343–348
 controlling XML format, 340–343
 with Oracle XSQL, 348–358
 overview, 334–339
 querystring parameters, 353–354
 reformatting date, 353
 renaming <ROWSET> and <ROW> elements, 352
 renaming XML elements, 351–352
 retrieving data as XML, 339–340
 server-side transformation, 339–343
 with SQL Server, 339–343
 XSLT transformation, 354–358

relational microformats, 271
relative URIs, 12
RELAX NG schema, 23–24
rel-directory microformats, 271
reliability, 460
rel-license microformats, 271
rel-payment microformats, 271
rel-tag microformats, 271
renaming
 <ROWSET> and <ROW> elements, 352
 XML elements, 351–352
rendering microformat documents, 280–281
Representational State Transfer (REST). *See* **REST (Representational State Transfer)**.
representations
 content negotiation, 214–215
 deleting resources, 204
 description, 201–202
 multiple, 212–214
 posting data to the server, 204
 requesting, 204
 returning headers, 204
 tracing requests, 204
 uploading, 204
 watching messages, 215–219
requesting representations, 204
requests, combining multiple, 91–92
reserved characters, XML, 224–225
Resource Description Framework (RDF), 251–256
resources
 description, 251–256
 requesting, 204
 REST, 201
 returning headers, 204
 security, 459–460
REST (Representational State Transfer)
 cookies, 203
 introduction, 200–201
 representations
 content negotiation, 214–215
 deleting resources, 204

REST (continued)

- description, 201–202
- multiple, 212–214
- posting data to the server, 204
- requesting, 204
- returning headers, 204
- tracing requests, 204
- uploading, 204
- watching messages, 215–219
- resources, 201
- state, 202–203
- transfer with HTTP methods, 203–204

REST services

- APP (Atom Publishing Protocol), 291
- calls, 291
- case management example, 288–292
- designing URI space, 288–289
- online resources, 294
- playlist sharing, example, 292–294
- in the real world, 292–294
- sample application, 288–292
- SOAP, 303–304
- tool support, 305–307
- versus WS-*, 303–307
- WSDL, 304–305

resumes, microformats, 272

returning representation headers, 204

reviews, microformats, 272

rich clients. See also browsers.

- current uses for, 169
- frameworks
 - OpenLaszlo, 170–179
 - XAML, 186–190
 - XUL, 179–186
- history of, 167–169

rich media. See multimedia.

Rivest, Ron, 469

root element, XML, 222–223

rounding corners, 52–53

<ROW> element, renaming, 352

<ROWSET> element, renaming, 352

RSS (Really Simple Syndication), tag map example, 132–142

RSS (Really Simple Syndication) 1.0

- definition, 246
- modules, 256–257
- RDF (Resource Description Framework), 251–256
- resource description, 251–256
- specifications, 256–257

RSS (Really Simple Syndication) 2.0

- autodiscovery, 251
- content encoding, 250
- definition, 246
- extensibility, 250
- item identification, 250
- MIME types, 250–251
- one-click subscription, 250–251
- simple content, 247–251
- specifications, 250

RSS Bandit, 373

RTMP (Real Time Messaging Protocol), 455

RTSP (Real Time Streaming Protocol), 455

Ruby on Rails, 321–322

S

SAML (Security Assertion Markup Language), 302

sample Web 2.0 applications. See BuzzWatch; “Hello World” examples.

Scalable Vector Graphics (SVG). See SVG (Scalable Vector Graphics).

schemas

- BuzzWatch, 23–24
- languages, XML, 232–233
- RELAX NG, 23–24
- W3C XML Schema language, 23–24

scheme section, URIs, 197

screen scraping

- analyzing pages, 409–410
- benefits, 407–408
- definition, 407

downloading pages, 408–409

drawbacks, 407–408

example, 410–411

fair practices, 408

scripting

accessibility issues, 105

BuzzWatch, 7–9

JSON (JavaScript Object Notation)

 alternative to XML, 236–238

 serving XML over HTTP, 332

languages, 5–6, 33–35

scripting, Ajax. *See also* JavaScript.

definition, 70

displaying XML information on a page, 71–75

extracting information from XML files, 71–75

Google Web Toolkit, 78–79

HTTP queries, 71–75

loading XML files, 71–75

memory leaks, 79–81

technologies, 70–71

user experience, 71

`XMLHttpRequest` object

 example, 72–75

 history, 71–72

YUI (Yahoo! UI) Library, 75–78

`YUIConnectionManager` object, 77–78

scripting, JavaScript. *See also* Ajax.

`==` operator, 64–65

`===` operator, 64–65

alternative to XML, 238

`for-in` iterator, 65–66

functions

 arguments, 67–68

 error checking, 68

 overloading, 67–68

 programming, 66–67

iterating, 65–66

optimization

 minimizing DOM updates, 70

 reducing download time, 69–70

type conversion, 64–65

`typeof` operator, 63

typing, 62–64

variables

 undefined, 62–64

 uninitialized *versus* undeclared, 63

search mashups, 402

Secure Sockets Layer (SSL). *See* SSL (Secure Sockets Layer).

security

browser issues, 168

Caesar shift encryption, 467–468

code

 data overflow, 482

 definition, 460

 general principles, 479–482

 SQL injection, 479–481

 XSS (cross-site scripting), 481–482

confidentiality, 460

decryption, 468

definition, 459–460

digital certificates, 469–470

early incorporation, 461

historical lessons, 461

integrity, 460

layered approach, 461–462

message, 460

message digests, 469

message encryption, 467–469

PKI (public key infrastructure), 468–469

reliability, 460

resource, 459–460

SAML, 302

standards-based techniques, 461

symmetric encryption, 467–468

user access control, SOAP, 303

web services

 adding policy, 486–487

 creating clients, 490–493

 custom authentication, 487–489

 definition, 482–483

 description, 485–486

 improvements, 493

 using, 483–484

WS-Security, 302

security, authentication

- in Apache, 466–467
- in ASPNET
 - configuring, 466
 - overview, 462–464
- definition, 459–460
- HTTP type, 466–467
- in IIS
 - configuring, 464–465
 - overview, 462–464
- providers, 466–467

security, authorization

- in Apache, 466–467
- in ASPNET
 - configuring, 466
 - overview, 462–464
- definition, 460
- in IIS
 - configuring, 464–465
 - overview, 462–464
- providers, 466–467

security, SSL (Secure Sockets Layer)

- in Apache
 - certificate request, generating on Unix, 476–477
 - certificate request, submitting in IIS, 477
 - configuring the site, 478
 - issued certificates, installing, 477–479
 - setting up, 474–478
- communication stages, 471
- in IIS
 - certificate requests, generating, 472–473
 - certificate requests, submitting, 473
 - configuring the site, 474–475
 - issued certificates, installing, 474
 - setting up, 471–475
- overview, 470
- TLS (Transport Socket Layer), 470

Security Assertion Markup Language (SAML), 302

`self` value, 260

serialization APIs, 315–317

Server Error responses, 207–208

server-side

- applications, 5
- HTTP redirection, 428
- transformation, 339–343
- XSLT, 322–324

servlets, 313–314

`Settings.java` class, 375, 388

Shamir, Avi, 469

shopping mashups, 402

shortcuts

- HTTP redirection, 427
- printable, 427
- user-friendly, 427

sidebars, accessibility issues, 107–108

Simple Object Access Protocol (SOAP). See SOAP (Simple Object Access Protocol).

`<simplelayout>` element, 172–173

SMIL (Synchronized Multimedia Integration Language), 136

SMIL multimedia format, 443–446

SOAP (Simple Object Access Protocol)

- caching, 303
- envelope opacity, 303–304
- logging system, 303
- REST services, 303–304
- user access control, 303
- WS-*, 295–299, 303–304

social networks, microformats, 271

`sortEntries` method, 384

source code, browsing, 174–175

spacing elements, OpenLaszlo, 172

SPARQL Protocol, 278–280

specifications, 261

Speex codec, 452

sports mashups, 402

spying on the user, 93

SQL injection, 479–481

SQL Server, converting relational data to XML, 339–343

SSL (Secure Sockets Layer)

- in Apache
 - certificate request, generating on Unix, 476–477
 - certificate request, submitting in IIS, 477
 - configuring the site, 478
 - issued certificates. installing, 477–479
 - setting up, 474–478
- communication stages, 471
- in IIS
 - certificate requests, generating, 472–473
 - certificate requests, submitting, 473
 - configuring the site, 474–475
 - issued certificates, installing, 474
 - setting up, 471–475
- overview, 470
- TLS (Transport Socket Layer), 470

standards org 2.0, 273–275

standards-based security techniques, 461

state, REST, 202–203

static content, 310–311

styles. See also CSS (Cascading Style Sheets).

- inline, accessibility issues, 99
- SVG, 130–132
- XForms, 147–151
- XSLT, 117–118

Subversion, 331

Success responses, 207–208

surrogate pairs, 229

SVG (Scalable Vector Graphics)

- Adobe plug-in, 141
- aligning text, 128
- animation, 136–137
- c (Cubic Bézier curve) command, 135
- color, 140
- currentTarget property, 140
- declarative animation, 137–139
- drawing a rectangle, 127
- E (Elliptical curve) command, 135
- examples
 - “Hello World” example, 126–130
 - RSS tag map, 132–142

- font attributes, 128
- H (Horizontal line) command, 135
- L (Line) command, 135
- M (Moveto) command, 134
- overview, 124–126
- pathSegList property, 140
- Q (Quadratic Bézier curve) command, 134–135
- SMIL, 136
- styles, 130–132
- swapping objects, 127
- text length calculation, 129
- V (Vertical) command, 135
- visibility attribute, 137
- Z (Close path) command, 135

swapping objects, 127

symmetric encryption, 467–468

Synchronized Multimedia Integration Language (SMIL), 136

syndication

- basics, 243–245
- feeds, 244–245
- microcontent, 244
- microformats, 271–272
- process description, 246

syndication channels

- compiling the application, 388–389
- deploying the application, 388–389
- e-mail support
 - configuring accounts, 392–393
 - local mail server, 391–393
 - mail support architecture, 391
 - overview, 389–390
- going live, 389
- running the application, 388–389

syndication channels, Atom service

- Apache Abdera toolkit, 374–375
- application architecture, 374
- code overview
 - AtomServlet.java class, 375, 377–383
 - clearFeed method, 384
 - clearing the feed, 376
 - compare method, 384–385

syndication channels, Atom service (continued)

- core files, 375
- createEntry method, 387
- createFeed method, 385–386
- isEntryInFeed method, 387
- loadFeed method, 386
- MailClient.java class, 375
- MailEntryCollector.java class, 375
- MessageReader.java class, 375
- Settings.java class, 375, 388
- sortEntries method, 384
- Utils.java class, 375, 383–388
- writeTo method, 386–387

e-mail support

- MailClient.java class, 393–396
- MailEntryCollector implementation, 396–400
- MessageReader interface, 396
- readMessage method, 399–400

feed reader, 373

user interface, 372–373

syndication formats

Atom

- alternate value, 260
- description, 258–260
- enclosure value, 260
- enclosures, 262–263
- history of, 257
- href attribute, 260
- hreflang attribute, 260
- length attribute, 260
- <link> element, 260
- podcasting, 262–263
- rel attribute, 260
- related value, 260
- self value, 260
- specifications, 261
- title attribute, 260
- type attribute, 260
- via value, 260

BuzzWatch, 5

elements, cross-format comparisons, 261–262

history of, 246–247

RSS 1.0

- definition, 246
- modules, 256–257
- RDF (Resource Description Framework), 251–256
- resource description, 251–256
- specifications, 256–257

RSS 2.0

- autodiscovery, 251
- content encoding, 250
- definition, 246
- extensibility, 250
- item identification, 250
- MIME types, 250–251
- one-click subscription, 250–251
- simple content, 247–251
- specifications, 250

T

tables, accessibility issues, 99–103

tabs, CSS for, 54–55

Tarkin codec, 452

technology agnosticism, 422–423

templates

XAML data, 190

XSLT, 113

temporary HTTP redirection, 426–427, 430

text

aligning, 128

elements, OpenLaszlo, 173–174

emphasis, accessibility issues, 99–103

length calculation, 129

text blocks, XAML, 190

<text> element, 173–174

<TextBlock> element, 190

Theora codec, 452

3xx Redirection responses, 207–208

302 Found status code, 427

Timeline application, 280–281

tip jars, microformats, 271

`title` attribute, 260

too much feedback, 92

tools

- Abdera Atom toolkit, 374–375
- Apache Web server, 208
- cURL command-line HTTP client, 208
- DevBoi for Firefox, 59
- DOM Inspector for Firefox, 55–56
- Firefox browser, 208
- Google Web Toolkit, 78–79
- HTML Tidy, 278
- HTTP reference toolkit, 208
- HTTPTracer graphic HTTP monitor, 208, 215–219
- Internet Explorer Developer Toolbar, 58–59
- page development, 55–59
- Twinkle, 279
- URI maintenance, 428–433
- Web Developer Toolbar for Firefox, 57

`TRACE` method, 204

tracing representation requests, 204

tracing requests, 204

traffic, design issues

- combining multiple requests, 91–92
- concise communication, 90–91
- excessive network traffic, 92
- limiting callbacks, 92–93
- processing on the server, 92
- removing error correction, 92
- spying on the user, 93
- too much feedback, 92

trailing slashes, 424–425, 430–431

transport layer, 195

Turtle syntax, 279

2xx Success responses, 207–208

`type` attribute, 260

type comparison operator (===), 64–65

type conversion operator (==), 64–65

`typeof` operator, 63

typing, JavaScript, 62–64

U

UBL (Universal Business Language), 235

UDDI, 302

undefined variables, 62–64

Unicast protocol, 455–456

Unicode, 227–229

uploading resource representations, 204

URIs (Uniform Resource Identifiers)

- authority section, 197
- description, 197–199
- fragment section, 198
- path section, 198
- query section, 198
- relative, 12
- scheme section, 197
- space, designing, 288–289
- specification, 194

URIs (Uniform Resource Identifiers),

maintenance

- change management, 425–428
- collections, 423–424
- contexts, 428
- future-proofing, 422–425
- hierarchies, 423–424
- HTTP redirection
 - ASP.NET, 433
 - basics, 425–426
 - content negotiation, 433
 - deprecated methods, 427–428
 - IIS, 433
 - intermediate information page, 427
 - path to query parameters, 431
 - permanent, 426, 430
 - printable shortcuts, 427
 - proxying, 431–433
 - redirect after `POST` pattern, 426
 - server-side, 428
 - temporary, 426–427, 430
 - trailing slashes, 430–431
 - user-friendly shortcuts, 427

URIs (Uniform Resource Identifiers), maintenance (continued)

Java servlets, 428–429
location independence, 424–425
mapping tools, 428–433
`mod_alias`, 428–433
`mod_negotiation`, 433
`mod_rewrite`, 428–433
technology agnosticism, 422–423
trailing slashes, 424–425, 430–431

URLs (Uniform Resource Locators)

accessibility issues, 104
description, 196–199
space, defining, 34–35
structure of, 194

user experience, 71

user interface, 372–373. See also Ajax; XForms; XUL.

syndication channels, 372–373

UTF-8 encoding, 228–229

UTF-16 encoding, 228–229

`Utils.java` class, 375, 383–388

V

v (Vertical) command, 135

variables, JavaScript

undefined, 62–64
uninitialized *versus* undeclared, 63

version control, 331

via value, 260

video. See multimedia.

`<view>` element, 178

viewing source code, 176

`visibility` attribute, 137

visually impaired users. See accessibility issues.

vocabulary, 265

Vorbis codec, 452

VoteLinks, 271

W

W3C (World Wide Web Consortium)

accessibility recommendations, 95
DOM specifications, 51
HTML proposals, 162–163
Semantic Web Initiative, 252
XML Schema, 23–24, 232–233

watches, 13–20

watching messages, 215–219

WAV format, 448

Web

versus Internet, 168
key reference documents, 194

Web 1.0

application layer, 195
client-server pattern, 195
components of, 194–199
domain names, 196
hidden code, dangers of, 199
HTTP 404 Not Found message, 194
layered protocols, 195
network layer, 195
physical link layer, 195
transport layer, 195
two-way communication, 200

URIs

authority section, 197
description, 197–199
fragment section, 198
path section, 198
query section, 198
scheme section, 197

URLs

description, 196–199
structure of, 194
web servers, purpose of, 199–200

web browsers. See browsers.

Web Content Accessibility Guidelines, 95

- Web Hypertext Application Technology Working Group (WHATWG).** *See* **WHATWG (Web Hypertext Application Technology Working Group).**
- web pages**
 extracting information from. *See* screen scraping.
 maps on, 417–419
- web servers**
 interaction, 12–24
 logs, 6–7
 purpose of, 199–200
- web services**
 definition, 287–288
 ebXML (Electronic Business XML), 302
 machine readable. *See* WSDL.
 REST services
 APP (Atom Publishing Protocol), 291
 calls, 291
 case management example, 288–292
 designing URI space, 288–289
 online resources, 294
 playlist sharing, example, 292–294
 in the real world, 292–294
 sample application, 288–292
 SOAP, 303–304
 tool support, 305–307
 versus WS-*, 303–307
 WSDL, 304–305
 SAML, 302
 security
 adding policy, 486–487
 creating clients, 490–493
 custom authentication, 487–489
 definition, 482–483
 description, 485–486
 improvements, 493
 using, 483–484
 specifications, 302
 UDDI, 302
- WS-
 Flickr SOAP API, 295–299
 namespaces, 298
versus REST, 303–307
 SOAP, 295–299, 303–304
 tool support, 305–307
 Web Services protocol stack, 301–302
 WSDL, 299–301, 304–305
 WS-BPEL, 302
 WS-Choreography, 302
 WS-Reliability, 302
 WS-Security, 302
- Web Services protocol stack, 301–302**
- web sites, extracting information from.** *See* **screen scraping.**
- WebDAV, 331**
- well-formed XML, 310**
- WHATWG (Web Hypertext Application Technology Working Group)**
 HTML proposals, 163–164
 XForms, 142–143
- Wiki channel, 274**
- Winer, Dave, 247**
- World Wide Web (WWW).** *See* **Web.**
- World Wide Web Consortium (W3C).** *See* **W3C (World Wide Web Consortium).**
- Writ codec, 452**
- `writeTo` **method, 386–387**
- WS-
 Flickr SOAP API, 295–299
 namespaces, 298
versus REST services, 303–307
 SOAP, 295–299, 303–304
 tool support, 305–307
 Web Services protocol stack, 301–302
 WSDL, 299–301, 304–305
- WS (web services).** *See* **web services.**
- WS-BPEL, 302**
- WS-Choreography, 302**

WSDL

- REST services, 304–305
- WS-*, 299–301, 304–305

WS-Reliability, 302

WS-Security, 302

WWW (World Wide Web). *See* **Web.**

WXS, 232–233

X

XAML (eXtensible Application Markup Language). *See also* **OpenLaszlo**; **XUL**.

- basics, 186–188
- binding data to external source, 188–190
- data templates, 190
- `<DataTemplate>` element, 190
- description, 186
- `ItemSource` attribute, 190
- list boxes, 190
- `<ListBox>` element, 190
- required libraries, 186
- text blocks, 190
- `<TextBlock>` element, 190

XForms

- buttons, 145
- examples
 - form to edit an Atom feed, 151–161
 - “Hello World,” 143–147
- illustrations, 146–147, 149, 151
- native implementations, 142–143
- styles, 147–151
- `xforms:input` element, 145
- `xforms:output` element, 145
- `xforms:trigger` element, 145
- and XML, 234
- `xforms:input` **element, 145**
- `xforms:output` **element, 145**
- `xforms:trigger` **element, 145**

XHTML

- BuzzWatch
 - semantic information, 29–33
 - transforming from XML, 30–33

- character encoding, 46–47
- and CSS, 44
- ease of use, 44
- empty elements, 45–46
- extensibility, 44–45
- versus* HTML, 44–47
- IDs, 46
- on mobile devices, 44
- names, 46
- reasons for using, 44–45
- Version 1.0, 43
- Version 1.1, 43
- and XML, 234
- as XML application, 45
- XML declaration, 46–47

XHTML 2.0 *versus* HTML 5, 164

XInclude, 234

Xiph.org Foundation, 452

XMDP profiles, 270–271

XML

- & (ampersand)
 - reserved character, 224
 - symbol for, 224
- ‘ (apostrophe), symbol for, 224
- > (greater than), symbol for, 224
- < (left angle bracket), reserved character, 224
- < (less than), symbol for, 224
- “ (quotation marks), symbol for, 224
- <- -> (comments), 225
- alternatives to
 - HTML, 238–240
 - JavaScript, 238
 - JSON (JavaScript Object Notation), 236–238
- & (ampersand) entity, 224
- APIs, 235
- &apos (apostrophe) entity, 224
- Atom Publishing Protocol, 235
- Atom Syndication Format, 235
- attributes, 223–224
- basics, 221–225
- benefits of, 235–236
- binary data, 229–230

- BMP (Basic Multilingual Plane), 227
- CDATA sections, 230
- character data, 224–225
- character encoding, 227–229
- code points, 227
- converting from binary data
 - automatically, 363–370
 - manually, 358–363
 - from Microsoft Word, 358
 - viewing raw data, 361
- converting from relational data
 - with ADO.NET, 343–348
 - client-side transformation, 343–348
 - controlling XML format, 340–343
 - with Oracle XSQL, 348–358
 - overview, 334–339
 - querystring parameters, 353–354
 - reformatting date, 353
 - renaming `<ROWSET>` and `<ROW>` elements, 352
 - renaming XML elements, 351–352
 - retrieving data as XML, 339–340
 - server-side transformation, 339–343
 - with SQL Server, 339–343
 - XSLT transformation, 354–358
- data formats, 235
- databases, 328–331
- datatypes, 231–232
- declaration, 225
- default namespace, 227
- drawbacks, 236
- DTD (Document Type Definition), 230
- elements, 222–223
- entities, 224, 230
- escape characters, 224–225
- files
 - displaying on pages, 71–75
 - extracting information from, 71–75
 - loading, 71–75
- `>` (greater than) entity, 224
- with Java, 313–317
- `<` (less than) entity, 224
- namespaces, 225–227
 - pipeline language, 325–328
 - PIs (Processing Instructions), 230
 - platform, serving XML dynamic content over HTTP, 325–328
 - protocols, 235
 - “ (quote marks) entity, 224
 - related technologies
 - XForms, 234
 - XHTML, 234
 - XInclude, 234
 - xml:base specification, 234
 - xml:id specification, 234
 - XPath, 233–234
 - XQuery, 234
 - XSLT, 233–234
 - renaming elements, 351–352
 - reserved characters, 224–225
 - root element, 222–223
 - schema languages, 232–233
 - surrogate pairs, 229
 - transforming into (X)HTML, 30–33
 - UBL (Universal Business Language), 235
 - Unicode, 227–229
 - unnecessary features, 230–231
 - UTF-8 encoding, 228–229
 - UTF-16 encoding, 228–229
 - W3C XML Schema, 232–233
 - WXS, 232–233
- XML, serving over HTTP**
 - character encoding, 310
 - dynamic content
 - with ASP.NET, 317–320
 - with C#, 317–320
 - example, 323–324
 - JAXP (Java API for XML Processing), 315–317
 - JSP, 314–315
 - life cycle, 312
 - with Perl, 322
 - with PHP, 322
 - principles, 312
 - with Python, 322
 - with Ruby on Rails, 321–322

XML, serving over HTTP (continued)

- serialization APIs, 315–317
- server-side XSLT, 322–324
- servlets, 313–314
- XML pipeline language, 325–328
- XML platform, 325–328
- XML with Java, 313–317
- with eXist, 329–330
- versus HTML, 309–310
- JSON, 332
- media types, 309
- static content, 310–311
- Subversion, 331
- version control, 331
- WebDAV, 331
- well-formed XML, 310
- XML databases, 328–331
- XQuery, 328–331
- XML Path Language (XPath). See XPath (XML Path Language).**
- XML Shareable Playlist Format (XSPF), 438**
- XML user interface language (XUL). See XUL (XML user interface language).**
- xml:base specification, 234**
- `XMLHttpRequest` **object**
 - example, 72–75
 - history, 71–72
- xml:id specification, 234**
- XOXO, 272**
- XPath (XML Path Language)**
 - and XML, 233–234
 - and XSLT, 111–112
- XPIN, 271**
- XQuery**
 - screen scraping example, 410–411
 - serving XML over HTTP, 328–331
 - and XML, 234
- XSLT (XSL Transformations)**
 - `<? ?>` (processing instructions), 122–124
 - in a browser, 122–124
 - dynamic typing, 113
 - “Hello World” example, 112–114

- instructions *versus* literals, 113
- listing channel items, 119–122
- namespaces, 113
- RSS 1.0 channel to HTML, 114–117
- styles, 117–118
- templates, 113
- transformation, 354–358
 - and XML, 233–234
 - and XPath, 111–112

XSPF (XML Shareable Playlist Format), 438

XSQL

- converting relational data to XML, 348–358
- page examples, 350–358
- required downloads, 349
- setting up, 348–350
- XSQL page, examples, 350–358

XSS (cross-site scripting), 481–482

XUL (XML user interface language). See also

OpenLaszlo; XAML.

- basics, 180–182
- `<button>` element, 181
- capabilities, 179
- description, 179
- directing browser to content, 180–182
- localization, 184–186
- manifest files, 180–182
- modifying browser behavior, 182–184
- namespaces, 181
- online resources, 179
- `onLoad` function, 183
- `onMenuItemCommand` function, 183–184

Y

Yahoo! UI (YUI) Library, 75–78

YUI (Yahoo! UI) Library, 75–78

`YUIConnectionManager` **object, 77–78**

Z

z (Close path) command, 135

ZVON.org, 51