

Index

A

- abort() method, 264
- age validator application, 156-157
- Ajax Construction Kit* live CD
 - backing up data, 275
 - contents, 271-272
 - installing on hard drives, 278
 - launching, 272-273
 - network setup, 273-274
 - overview, 271
 - running as virtual client, 278
 - running from Web server, 278
 - running on Mac OS X, 277-278
 - running on Windows PCs, 276
 - tar archives, 278
- Ajax in pop culture*
 - Ajax, Canada, 171
 - Ajax character in movie *The Warriors*, 189
 - Ajax cleanser, 139
 - Ajax Corporation, 9
 - Ajax Duckman (TV character), 231
 - Ajax the Greek hero, 97
 - Ajax mountain, 205
 - Ajax the racehorse, 121
 - Ajax toilet, 159
 - Ajax video game, 49
 - USS Ajax, 25
- ajaxkit.js file, 26, 34-35, 76, 140
 - Completer application, 122
 - Forecaster application, 190
 - Informer application, 172

- News Feeder application, 98
- Rater application, 232
- ajaxSendRequest() function, 34, 41, 239
- ajaxUpdateState() function, 35
- ajaxUpdateStatus() function, 42
- animated “loading” images, 169
- anticipating data entry. *See*
 - Completer application
- applications. *See specific applications*
- asynchronous communications, 12, 19
- Atom news feeds, 98
- auto-complete application. *See*
 - Completer application

B

- backing up
 - Ajax Construction Kit* live CD data, 275
 - XAMPP server, 275
- blogs, Booker application
 - (e-book viewer), 73-74
- <body> tag (HTML)
 - Completer application
 - (auto-completer), 129
 - Forecaster application, 197-198
 - Imaginator application, 164
 - informer.html Web page, 179-180
 - News Feeder application, 108
 - Rater application, 241-242
 - shipper.html Web page, 216-218
 - snapper.html Web page, 174

`book_alicewonderland.xml` file, 26

`book_frankenstein.xml` file, 26

`book_mobydick.xml` file, 26

`book_warofworlds` file, 26

`<bookData>` tag (XML), 56-57

Booker application (e-book viewer)

`booker.html` Web page, 26, 39-43, 61

`ajaxSendRequest()` function, 41

`ajaxUpdateStatus()` function, 42

code listing, 62-63

`handleRequest()` function, 42, 65-67

`loadBook()` function, 41-43

`loadChapter()` function, 64-65

`nextChapter()` function, 67-68

`prevChapter()` function, 67

`replaceText()` function, 41

`booker.php` server script, 68-70

client requests, 53

client/server conversation, 55-57

customizing, 43-44

data-driven blog tool, 73-74

e-book files, 39

files, 25-26, 50

loading data in smaller chunks, 50-51

overview, 38

requests, 31-32

server responses, 53-55

testing, 70-71

XML e-books, 58-61

`booker.html` Web page, 26, 39-43, 61

`ajaxSendRequest()` function, 41

`ajaxUpdateStatus()` function, 42

code listing, 62-63

`handleRequest()` function, 42, 65-67

`loadBook()` function, 41-43

`loadChapter()` function, 64-65

`nextChapter()` function, 67-68

`prevChapter()` function, 67

`replaceText()` function, 41

`booker.php` script, 68-70

C

calculating shipping charges, 20-23. *See also*

Shipper application (shipping calculator)

cascading style sheets. *See* CSS

CDs. *See* live CDs

`<channel>` tag (XML), 106

`<chapter>` tag (XML), 59

client/server conversations, 55-57. *See also* requests; responses

Completer application, 127-128

Forecaster application, 194-196

Informer application, 178-179

News Feeder application, 103-106

Picker application, 80-83

Rater application, 236-240

Shipper application, 211-215

Validator application, 145-146

commands, `modprobe`, 274

COMPLETE state (requests), 30

Completer application (auto-completer)

`ajaxkit.js` file, 122

challenges of anticipating data

entry, 122-124

client requests, 126-127

client/server conversation, 127-128

`completer.html` Web page, 122, 129-133

`<body>` tag, 129

`getCompletion()` function, 130-131

`handleCompletionRequest()`

function, 131-132

`onkeyup` event handler, 130

`onload` event handler, 130

`showBiography()` function, 132-133

`completer.php` server script, 122, 133-134

context-sensitive search boxes, 136-137

`domkit.js` file, 122

overview, 121

server responses, 127

testing, 134-135

user interface, 124-126

when to use auto-complete, 135-136

`completer.html` Web page, 122, 129-133

`<body>` tag, 129

`getCompletion()` function, 130-131

`handleCompletionRequest()`

function, 131-132

`onkeyup` event handler, 130

`onload` event handler, 130

`showBiography()` function, 132-133

`completer.php` server script, 122, 133-134

configuring networking for *Ajax*

Construction Kit live CD, 273-274

context-sensitive search boxes, 136-137

`cookiesreset.html` Web page, 232, 244-246

CSS (cascading style sheets)

`infowinkit.css` file, 182

`newsfeeder.css` style sheet, 98, 112-114

- overview, 27
- ratingkit.css file, 242

customizing Booker application (e-book viewer), 43-44

D

data entry, anticipating. *See* Completer application

<date> tag (XML), 105

<description> tag (XML), 195-196

diagnostic bar, turning on/off, 44, 62

documents (XML), 58

domkit.js file, 26, 35-36, 76, 140

- Completer application, 122
- Forecaster application, 190
- Imaginator application, 160, 166
- Informer application, 172
- News Feeder application, 98
- Rater application, 232

Duckman, Ajax (TV character), 231

E

e-book viewer. *See* Booker application

<ebook> tag (XML), 59

elements (XML). *See* tags

embedding news feeds in Web pages, 117-118

enabling diagnostic bar, 62

event handlers

- onload, 108
- onreadystatechange, 269

event object, 181

Extensible Markup Language. *See* XML

F

file_get_contents() function, 115

files

- Ajax toolkit
 - ajaxkit.js, 34-35
 - domkit.js, 35-36
 - infowinkit.js, 37
 - modalingkit.js, 36-37
 - ratingkit.js, 37-38
 - validatekit.js, 37
- Booker application, 25-26, 50
 - booker.html, 39-43
 - customizing, 43-44
 - e-book files, 39

- Completer application, 121
 - completer.html, 129-133
 - completer.php, 133-134
- Forecaster application, 189
 - forecaster.html Web page, 196-198
 - forecaster.php, 194, 198-199
- Imaginator application, 160
 - domkit.js file, 166
 - modalingkit.css file, 165
 - modalingkit.js file, 165
 - modalingkit_gray.png file, 166
 - modalingki_wait.gif file, 163
- Imaginator application, 161
- Informer application, 172
 - informer.html Web page, 179-182
 - infowinkit.css file, 182
 - infowinkit.js, 175
 - infowinkit.js file, 182
 - skateXXX.html Web pages, 182-184
- News Feeder application, 98
 - newsfeeder.css style sheet, 112-114
 - newsfeeder.html, 107-112
 - newsfeeder.php, 114-115
 - wait.gif, 100-101
- Picker application, 75
- Rater application, 232
- Validator application, 140
- XML files, 58

five-star rating system. *See* Rater application

Flickr, 16-18

Forecaster application (weather forecast window)

- ajaxkit.js file, 190
- challenges of gathering dynamic weather information, 190-191
- changing city, 203
- client requests, 193
- client/server conversation, 194-196
- detailed weather reports, 203-204
- domkit.js file, 190
- forecaster.html Web page, 190, 196-198
- forecaster.php server script, 190, 194, 198-199
- infowinkit.css file, 190
- infowinkit.js file, 190
- infowinkit_wait.gif file, 190
- overview, 189
- server responses, 193
- testing, 200-202
- user interface, 191-193

forecaster.html Web page, 190, 196-198
 forecaster.php server script, 190,
 194, 198-199
 functions. *See specific functions*
 future of Ajax, 259

G

Garrett, Jesse James, 257
 getAllResponseHeaders() method, 265
 getCityState() function, 150
 getClientHeight() function, 36
 getClientLeft() function, 36
 getClientTop() function, 36
 getClientWidth() function, 36
 getCompletion() function, 130-131
 getCookieData() function, 36
 getElementById() function, 111, 149
 getResponseHeader() method, 266
 getShipRate() function, 218-219
 getText() function, 36
 Gmail validation, 140
 Google
 Gmail validation, 140
 Google Maps, 18, 258
 Google Suggest, 15, 122

H

handleCityStateRequest() function, 150-151
 handleCompletionRequest()
 function, 131-132
 handleFeedRequest() function, 110-111
 handleRequest() function, 42, 65-67, 220
 handleStockInfoRequest() function, 89-90
 handleStockListRequest() function, 87-88
 hard drives, installing *Ajax Construction*
 Kit live CD on, 278
 Harrington, Sir John, 159
 history
 of Ajax, 257-258
 of RSS, 98
 HTML (Hypertext Markup Language).
 See specific pages (for example,
 newsfeeder.html)
 HTML (Hypertext Markup Language) tags.
 See tags (HTML)

I

The Iliad, 97
 Imaginator application (image viewer)
 adding images to, 168
 animated “loading” images, 169
 challenges of loading images
 dynamically, 160-161
 domkit.js file, 160, 166
 imaginator.html Web page, 160, 163-165
 modaling.js file, 161
 modalingkit.css file, 160, 165
 modalingkit.js file, 160-161, 165
 modalingkit_gray.png file, 160, 166
 modalingkit_wait.gif file, 160, 163
 overview, 159
 smokiesXX.jpg files, 160
 smokiesXX_thumb.jpg files, 160
 testing, 166-167
 user interface, 162-163
 imaginator.html Web page, 160, 163-165
 IMDb (Internet Movie Database), 95
 Informer application (popup windows)
 advantages of popups, 172-173
 ajaxkit.js file, 172
 client requests, 177
 client/server conversation, 178-179
 domkit.js file, 172
 informer.html Web page, 172, 179-182
 infowinkit.css file, 172, 182
 infowinkit.js file, 172, 175, 182
 infowinkit_wait.gif file, 172
 modifying content of, 186-187
 overview, 171-172
 server responses, 177-178
 skateXXX.html Web pages, 172, 182-184
 skateXXX.jpg files, 172
 testing, 184-186
 user interface, 176-177
 informer.html Web page, 172, 179-182
 infowinkit.css file
 Forecaster application, 190
 Informer application, 172, 182
 infowinkit.js file, 37
 Forecaster application, 190
 Informer application, 172, 175, 182
 infowinkit_wait.gif file
 Forecaster application, 190
 Informer application, 172

initForm() function, 219
initInfoWindow() function, 37, 176
initModal() function, 36, 161, 165
initRatings() function, 37, 233
<input> tag (HTML), 63
input validation
 age validation, 156-157
 Gmail validation, 140
 NFL quarterback rating validation, 157-158
 overview, 139
 Validator application. *See*
 Validator application
INTERACTIVE state (requests), 30
Internet Movie Database (IMDb), 95
The Invisible Ad advertisement service, 16
irrational exuberance, 259

J-K

JavaScript
 event object, 181
 overview, 27
 XMLHttpRequest object
 abort() method, 264
 getAllResponseHeaders() method, 265
 getResponseHeader() method, 266
 onreadystatechange event, 269
 open() method, 262-263
 overrideMimeType() method, 264-265
 overview, 261
 readyState property, 267
 request lifecycle, 30-32
 responseText property, 268
 responseXML property, 268
 send() method, 263-264
 setRequestHeader() method, 265
 status property, 267
 statusText property, 268
Jung, Carl, 11

Kontera service, 174

L

launching *Ajax Construction Kit*
 live CD, 272-273
** tag (HTML)**, 243
lifecycles of requests, 30-32
<link> tag (XML), 105

live CDs

Ajax Construction Kit live CD
 backing up data, 275
 contents, 271-272
 installing on hard drives, 278
 launching, 272-273
 network setup, 273-274
 overview, 271
 running as virtual client, 278
 running from Web server, 278
 running on Mac OS X, 277-278
 running on Windows PCs, 276
 tar archives, 278

 Live Linux CD, 4

loadBook() function, 41-43

loadChapter() function, 64-65

LOADED state (requests), 30

loadFeed() function, 108-109

loading

 data in small chunks, 50-51
 images dynamically. *See*
 Imaginator application

LOADING state (requests), 30

loadStockInfo() function, 86-89

loadStockList() function, 86

logs, server, 70

Lyricsfly.com, 122

M

Mac OS X, 277-278

Maps (Google), 18, 258

methods (XMLHttpRequest object)

abort(), 264
 getAllResponseHeaders(), 265
 getResponseHeader(), 266
 open(), 262-263
 overrideMimeType(), 264-265
 send(), 263-264
 setRequestHeader(), 265

Microsoft Remote Scripting technology, 257

modal windows, 162

modalimg.js file, 161

modalimgkit.css file, 160, 165

modalimgkit.js file, 36-37, 160-161, 165

modalimgkit_gray.png file, 160, 166

modalimgkit_wait.gif file, 160, 163

modprobe command, 274

Mossadegh, Mohammed, 75

movie collection application, creating from
 Picker application, 94-96
 MySQL database rating systems, 252-253

N

Netflix, 16
 networks, configuring for *Ajax*
Construction Kit live CD, 273-274
 news aggregators, 99
 News Feeder application. *See also*
 RSS news feeds
 adding news feeds to, 117
 ajaxkit.js file, 98
 client requests, 101
 client/server conversation, 103-106
 creating news feeds, 118
 domkit.js file, 98
 embedding news feeds in Web
 pages, 117-118
 files, list of, 98
 news aggregators, 99
 newsfeeder.css style sheet, 98, 112-114
 newsfeeder.html Web page, 98, 107-112
 <body> tag, 108
 getElementById() function, 111
 handleFeedRequest()
 function, 110-111
 loadFeed() function, 108-109
 onload event handler, 108
 newsfeeder.php server script, 98, 114-115
 newspaper.gif file, 98
 querying news feeds dynamically, 98-100
 server responses, 102
 testing, 115-116
 user interface design, 100-101
 wait.gif file, 98-101
 newsfeeder.css style sheet, 98, 112-114
 newsfeeder.html Web page, 98, 107-112
 <body> tag, 108
 getElementById() function, 111
 handleFeedRequest() function, 110-111
 loadFeed() function, 108-109
 onload event handler, 108
 newsfeeder.php server script, 98, 114-115
 newspaper.gif file, 98
 nextChapter() function, 67-68
 NFL quarterback rating validator
 application, 157-158
 nitRatings() function, 244

O

objects
 Event, 181
 XMLHttpRequest
 abort() method, 264
 getAllResponseHeaders() method, 265
 getResponseHeader() method, 266
 onreadystatechange event, 269
 open() method, 262-263
 overrideMimeType() method, 264-265
 overview, 261
 readyState property, 267
 request lifecycle, 30-32
 responseText property, 268
 responseXML property, 268
 send() method, 263-264
 setRequestHeader() method, 265
 status property, 267
 statusText property, 268
 onblur event handler, 150
 onkeyup event handler, 130
 online shopping carts, 12, 20-23
 onload event handler
 Completer application
 (auto-completer), 130
 Informer application (popup windows), 180
 News Feeder application, 108
 Picker application (stock picker), 86
 Validator application, 149
 onreadystatechange event, 269
 open() method, 262-263
 Operation Ajax, 75
 <option> tag (XML), 100
 Orkut, 258
 overrideMimeType() method, 264-265

P

PHP
 booker.php script, 68-70
 completer.php server script, 122, 133-134
 forecaster.php script, 190, 194, 198-199
 newsfeeder.php script, 98, 114-115
 overview, 29, 57
 quoter.php server script, 91-92
 ratingkit.php server script, 232
 server.php server script, 221-223
 ziplookup.php server script, 140, 151-152

Picker application (stock picker)

- client requests, 79
- client/server conversation, 80-83
- creating movie collection application
 - from, 94-96
- files, 75
- goals of, 76-78
- picker.html Web page, 76, 85-91
 - code listing, 85-86
 - handleStockInfoRequest()* function, 89-90
 - handleStockListRequest()* function, 87-88
 - loadStockInfo()* function, 86-89
 - loadStockList()* function, 86
 - onload event handler, 86
 - replaceText()* function, 90
- quoter.php server script, 91-92
- server responses, 79-80
- stocks.xml file, 83-85
- testing, 93-94
- user interface, 78-79

picker.html Web page, 76, 85-91

- code listing, 85-86
- handleStockInfoRequest()* function, 89-90
- handleStockListRequest()* function, 87-88
- loadStockInfo()* function, 86-89
- loadStockList()* function, 86
- onload event handler, 86
- replaceText()* function, 90

popularity of Ajax, 258**popup windows**

- advantages of, 172-173

Informal application

- client requests, 177*
- client/server conversation, 178-179*
- modifying content of, 186-187*
- server responses, 177-178*
- testing, 184-186*
- user interface, 176-177*

- Kontera service, 174

- overview, 171-172

- size of, 187

- Snap Preview service, 174-175

- weather forecast window

- (Forecaster application)

- ajaxkit.js file, 190*
- challenges of gathering dynamic weather information, 190-191*
- changing city, 203*

- client requests, 193*

- client/server conversation, 194-196*

- detailed weather reports, 203-204*

- domkit.js file, 190*

- forecaster.html Web page, 190, 196-198*

- forecaster.php server script, 190, 194, 198-199*

- infowinkit.css file, 190*

- infowinkit.js file, 190*

- infowinkit_wait.gif file, 190*

- overview, 189*

- server responses, 193*

- testing, 200-202*

- user interface, 191-193*

- prevChapter() function, 67**

properties of XMLHttpRequest object

- readyState, 267

- responseText, 268

- responseXML, 268

- status, 267

- statusText, 268

- <pubDate> tag (XML), 105**

Q-R

- querying news feeds dynamically, 98-100**

- quoter.php script, 76**

- quoter.php server script, 91-92**

Rater application (rating system)

- adding to Web pages, 250

- ajaxkit.js file, 232

- challenges, 232-234

- changing images in, 252

- client requests, 235-236

- client/server conversation, 236-240

- cookieset.html Web page, 232, 244-246

- domkit.js file, 232

- item IDs, 250-252

- MySQL databases, 252-253

- overview, 231-232

- rater.html Web page, 232, 240-244

- <body> tag, 241-242*

- initRatings() function, 244*

- tag, 243*

- setRating() function, 244*

- tag, 243*

- tag, 243*

- ratingkit.css file, 232, 242

- ratingkit.js file, 232

- ratingkit.php server script, 232
- server responses, 236
- star.gif file, 232
- star_sm.gif file, 232
- testing, 246-249
- user interface, 234-235
- rater.html Web page**, 232, 240-244
 - <body> tag, 241-242
 - initRatings() function, 244
 - tag, 243
 - setRating() function, 244
 - tag, 243
 - tag, 243
- rating systems. See Rater application**
- ratingkit.css file, 232, 242
- ratingkit.js file, 37-38, 232
- ratingkit.php server script, 232
- readyState property (XMLHttpRequest object)**, 267
- real-time validation. See input validation**
- real-world applications**, 14-15
 - Flickr, 16-18
 - Google Maps, 18, 258
 - Google Suggest, 15
 - The Invisible Ad advertisement service, 16
 - Netflix, 16
- Really Simple Syndication. See RSS news feeds**
- relationship between Ajax technologies**, 26
- reloading Web pages**, 12-13
- Remote Scripting technology**, 257
- replaceText() function**, 35-36, 41, 90
- requests**
 - Booker application, 53
 - Completer application, 126-127
 - Forecaster application, 193
 - Informer application, 177
 - lifecycle, 30-32
 - News Feeder application, 101
 - Picker application, 79
 - Rater application, 235-236
 - Shipper application, 209-211
 - Validator application, 144-145
 - XMLHttpRequest object
 - abort() method, 264
 - getAllResponseHeaders() method, 265
 - getResponseHeader() method, 266
 - onreadystatechange event, 269
 - open() method, 262-263
 - overrideMimeType() method, 264-265
 - overview, 261
 - readyState property, 267
 - responseText property, 268
 - responseXML property, 268
 - send() method, 263-264
 - setRequestHeader() method, 265
 - status property, 267
 - statusText property, 268
- responses**
 - Booker application, 53-55
 - Completer application, 127
 - Forecaster application, 193
 - Informer application, 177-178
 - News Feeder application, 102
 - Picker application, 79-80
 - Rater application, 236
 - Shipper application, 211
 - Validator application, 145
- responseText property (XMLHttpRequest object)**, 268
- responseXML property (XMLHttpRequest object)**, 268
- restaurant analogy**, 10-11
- RSS news feeds**
 - creating, 118
 - embedding in Web pages, 117-118
 - history of, 98
 - news aggregators, 99
 - News Feeder application
 - adding news feeds to, 117
 - ajaxkit.js file, 98
 - client requests, 101
 - client/server conversation, 103-106
 - domkit.js file, 98
 - files, list of, 98
 - newsfeeder.css style sheet, 98, 112-114
 - newsfeeder.html Web page, 98, 107-112
 - newsfeeder.php server script, 98, 114-115
 - newspaper.gif file, 98
 - server responses, 102
 - testing, 115-116
 - user interface design, 100-101
 - wait.gif file, 98-101
- overview, 97, 192
- querying dynamically, 98-100
- Scientific American Web site, 103-106
- USA Today Web site, 102

running Ajax Construction Kit live CD

- installing on hard drives, 278
- on Mac OS X, 277-278
- tar archives, 278
- as virtual client, 278
- from Web server, 278
- on Windows PCs, 276

S**Scientific American news feeds, 103-106****scripts**

- booker.php, 68-70
- completer.php, 122, 133-134
- forecaster.php, 190, 194, 198-199
- newsfeeder.php, 98, 114-115
- quoter.php, 91-92
- ratingkit.php, 232
- server.php server script, 221-223
- ziplookup.php, 140

<select> tag (HTML), 63, 228**send() method, 263-264****server.php server script, 221-223****servers**

- conversations. *See* client/server conversations
- logs, 70
- responses. *See* responses
- role of, 29
- scripts. *See* scripts
- XAMPP server, backing up, 275

setRating() function, 233, 244**setRequestHeader() method, 265****Shipper application (shipping calculator)**

- accessing USPS production server, 226-229
- challenge of retrieving live shipping charges, 206-207
- client requests, 209-211
- client/server conversation, 211-215
- overview, 205-206
- server responses, 211
- server.php server script, 221-223
- shipper.html Web page, 215-221
 - <body> tag, 216-218*
 - getShipRate() function, 218-219*
 - handleRequest() function, 220*
 - initForm() function, 219*
 - <select> element, 228*
 - updateTotal() function, 220*

testing, 223-225

user interface, 207-208

wait.gif image, 208

shipper.html Web page, 215-221*<body> tag, 216-218**getShipRate() function, 218-219**handleRequest() function, 220**initForm() function, 219**<select> element, 228**updateTotal() function, 220***shipping calculator. *See* Shipper application****shopping carts, 12, 20-23****showBiography() function, 132-133****showInfoWindow() function, 37, 176, 181****size of popup windows, 187****skateXXX.html Web pages, 172, 182-184****skateXXX.jpg files, 172****smokiesXX.jpg files, 160****smokiesXX_thumb.jpg files, 160****Snap Preview service, 174-175****snapper.html Web page, 172-174**** tag (HTML), 243****star.gif file, 232****star_sm.gif file, 232****status property (XMLHttpRequest**

object), 267

statusText property (XMLHttpRequest

object), 268

stock picker. *See* Picker application**<stock> tag, 82, 85****stocks.xml file, 76, 83-85****strtolower() function, 134****style sheets. *See* CSS (cascading style sheets)****Suggest (Google), 15****Symbol Lookup, 84****synchronicity, 11****Synchronicity, 11****T****tags (HTML)***<body>**Completer application, 129**Forecaster application, 197**Imaginator application, 164**informer.html Web page, 179-180**News Feeder application, 108**Rater application, 241-242*

- shipper.html* Web page, 216-218
- snapper.html* Web page, 174
- <head>, 198
- <input>, 63
- , 243
- <select>, 63, 228
- , 243
- , 243
- tags (XML)
 - <bookData>, 56-57
 - <channel>, 106
 - <chapter>, 59
 - <date>, 105
 - <description>, 195-196
 - <ebook>, 59
 - <link>, 105
 - <option>, 100
 - <pubDate>, 105
 - <stock>, 82, 85
 - <title>, 105
- testing
 - Booker application, 70-71
 - Complete application, 134-135
 - Forecaster application, 200-202
 - Imaginator application, 166-167
 - Informer application, 184-186
 - News Feeder application, 115-116
 - Picker application, 93-94
 - Rater application, 246-249
 - Shipper application, 223-225
 - Validator application, 153-155
- <title> tag (XML), 105
- toolkit
 - ajaxkit.js file, 34-35
 - domkit.js file, 35-36
 - infowinkit.js file, 37
 - modalingkit.js file, 36-37
 - overview, 33-34
 - ratingkit.js file, 37-38
 - validatekit.js file, 37
- TRYNT Movie IMDb Web Service, 95
- turning on/off diagnostic bar, 44, 62

U

- , tag (HTML), 243
- UNINITIALIZED state (requests), 30
- updateTotal() function, 220
- USA Today news feeds, 102

user interfaces

- Completer application, 124-126
- Forecaster application, 191-193
- Imaginator application, 162-163
- Informer application, 176-177
- News Feeder application, 100-101
- Picker application, 78-79
- Rater application, 234-235
- Shipper application, 207-208
- Validator application, 143-144
- USPS Web Tools, 21, 226-228
- USS Ajax, 25

V

- validateAge() function, 156-157
- validateDate() function, 37, 142
- validateEmail() function, 37, 142
- validateInteger() function, 37, 142, 149
- validatekit.js file, 37, 140-143
- validateNonEmpty() function, 37, 142
- validateNumber() function, 37, 142
- validatePhone() function, 37, 142
- validateZipCode() function, 37, 142, 150
- validating input. *See* input validation
- Validator application (input validation)
 - ajaxkit.js file, 140
 - applying validation functions to other user-input forms, 156-158
 - client requests, 144-145
 - client/server conversation, 145-146
 - data types, 141
 - domkit.js file, 140
 - overview, 139
 - server responses, 145
 - testing, 153-155
 - user interface, 143-144
 - validatekit.js file, 140-143
 - validator.html Web page, 140, 147-151
 - code listing*, 147-148
 - getCityState()* function, 150
 - getElementById()* function, 149
 - handleCityStateRequest()* function, 150-151
 - onblur* event handler, 150
 - onload* event handler, 149
 - validateInteger()* function, 149
 - validateZipCode()* function, 150
 - wait.gif file, 140, 143

- ziplookup.php script, 140
- ziplookup.php server script, 151-152
- validator.html Web page, 140, 147-151
 - code listing, 147-148
 - getCityState() function, 150
 - getElementById() function, 149
 - handleCityStateRequest() function, 150-151
 - onblur event handler, 150
 - onload event handler, 149
 - validateInteger() function, 149
 - validateZipCode() function, 150
- value attribute (<option> tag), 100
- version numbers of Web, 12
- viewing images. *See* Imaginator application

W

- wait.gif file, 98, 100-101, 140, 143
- The Warriors*, 189
- weather forecast window. *See* Forecaster application
- Web pages, reloading, 12-13. *See also* specific pages (for example, newsfeeder.html)
- Web servers
 - running *Ajax Construction Kit* live CD from, 278
 - XAMPP server, backing up, 275
- Web sites
 - Flickr, 16-18
 - Google Maps, 18, 258
 - Google Suggest, 15
 - The Invisible Ad advertisement service, 16
 - Netflix, 16
 - Orkut, 258
 - shopping carts, 20-23
- Web version numbers, 12
- webserviceX.net, 151
- windows
 - modal windows, 162
 - popup windows. *See also* Informer application
 - advantages of*, 172-173
 - Kontera service*, 174
 - overview*, 171
 - size of*, 187
 - Snap Preview service*, 174-175
 - weather forecast windows. *See* Forecaster application
- Windows, running *Ajax Construction Kit* live CD on, 276

X

- XAMPP server, backing up, 275
- XML (Extensible Markup Language)
 - documents, 58
 - files, 58
 - overview, 28-29
 - tags
 - <bookData>, 56-57
 - <channel>, 106
 - <chapter>, 59
 - <date>, 105
 - <description>, 195-196
 - <ebook>, 59
 - <link>, 105
 - <option>, 100
 - <pubDate>, 105
 - <stock>, 82, 85
 - <title>, 105
 - XML e-books (Booker application), 58-61
 - .xml file extension, 61
 - XMLHttpRequest object
 - abort() method, 264
 - getAllResponseHeaders() method, 265
 - getResponseHeader() method, 266
 - onreadystatechange event, 269
 - open() method, 262-263
 - overrideMimeType() method, 264-265
 - overview, 261
 - readyState property, 267
 - request lifecycle, 30-32
 - responseText property, 268
 - responseXML property, 268
 - send() method, 263-264
 - setRequestHeader() method, 265
 - status property, 267
 - statusText property, 268

Y-Z

- Yahoo!'s Symbol Lookup, 84
- ZIP code validator. *See* Validator application
- ziplookup.php server script, 140, 151-152



THIS BOOK IS SAFARI ENABLED

INCLUDES FREE 45-DAY ACCESS TO THE ONLINE EDITION

The Safari® Enabled icon on the cover of your favorite technology book means the book is available through Safari Bookshelf. When you buy this book, you get free access to the online edition for 45 days.

Safari Bookshelf is an electronic reference library that lets you easily search thousands of technical books, find code samples, download chapters, and access technical information whenever and wherever you need it.

TO GAIN 45-DAY SAFARI ENABLED ACCESS TO THIS BOOK:

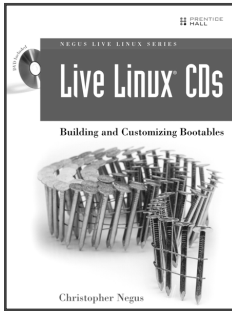
- Go to <http://www.prenhallprofessional.com/safarienabled>
- Complete the brief registration form
- Enter the coupon code found in the front of this book on the "Copyright" page

If you have difficulty registering on Safari Bookshelf or accessing the online edition, please e-mail customer-service@safaribooksonline.com.



PRENTICE
HALL

Also Available in the Negus Live Linux Series



Live Linux® CDs

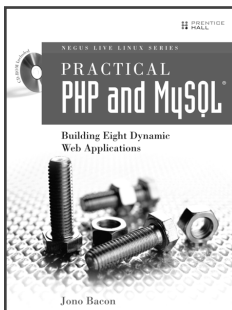
Building and Customizing Bootables

Christopher Negus | 0132432749 | ©2007

Create Custom Versions of Linux That Run “Live,” Without Installation!

“Live” Linux® CDs let users run Linux on any PC, without affecting the operating system and data already present there. Live Linux distributions, such as Knoppix, are now among the most popular versions of Linux. What’s more, because Linux is open source, you can customize your own Live Linux distribution for virtually any purpose. *Live Linux® CDs* is the first start-to-finish guide to creating, building, and remastering your own Live Linux distributions.

Bestselling *Linux Bible* author, Christopher Negus, walks you step-by-step through building complete Linux systems that run from CDs, DVDs, flash drives, and other bootable media. First learn exactly how Live Linux works and then walk through creating Live Linux distributions based on five different systems: KNOPPIX (Debian), Fedora/Kadischi, Gentoo, Slax (Slackware), and Damn Small Linux. Working from complete examples on the accompanying DVD-ROM, customize all these specialized bootable Linuxes.



Practical PHP and MySQL®

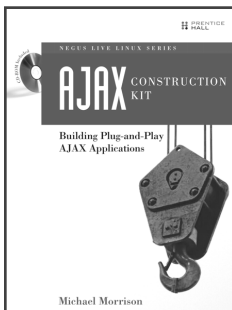
Building Eight Dynamic Web Applications

Jono Bacon | 0132239973 | ©2007

Build Dynamic Web Sites Fast, with PHP and MySQL...Learn from Eight Ready-to-Run Applications!

Suddenly, it’s easy to build commercial-quality Web applications using free and open source software. With this book, you learn from eight ready-to-run, real-world applications—all backed by clear diagrams and screenshots, well-documented code, and simple, practical explanations.

Leading open source author Jono Bacon teaches the core skills you need to build virtually any application. You discover how to connect with databases, upload content, perform cascading deletes, edit records, validate registrations, specify user security, create reusable components, use PEAR extensions, and even build Ajax applications.



Ajax Construction Kit

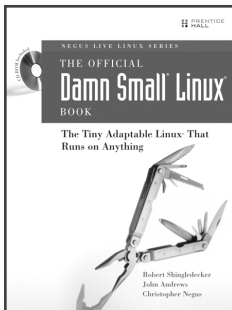
Building Plug-and-Play Ajax Applications

Michael Morrison | 0132350084 | ©2008

Supercharge Your Sites with Ajax Right Now...No Scripting Expertise Needed!

You’ve heard how great Ajax is—how it can help make your Web sites more usable, more interactive, more responsive, more successful. *Ajax Construction Kit* lets you put Ajax to work right now, even if you’ve never written a script! Just learn a few essentials, check out a few examples, and then run the live CD and discover all the plug-and-play code you need to hit the ground running.

Ajax Construction Kit’s built-in applications work right out of the box. And with easy guidance from Michael Morrison, you’ll gradually deepen your understanding—learn how to customize, extend, and reuse these applications—and even build skills for creating new ones. Walk away an expert.



The Official Damn Small® Linux® Book

The Tiny Adaptable Linux That Runs on Anything

Robert Shingledecker, John Andrews, Christopher Negus | 0132338696 | ©2008

Make the Most of Today’s Smallest, Fastest Linux Distribution—Damn Small Linux!

Damn Small Linux (DSL) is a super-efficient platform for everything from custom desktops to professional servers. Now, DSL’s creator and lead developer have written the first definitive, practical guide to this remarkable system. *The Official Damn Small® Linux® Book* brings together everything you need to put DSL to work in just minutes. Simply learn a few essentials, boot the live CD-ROM, and master the rest...one step at a time, hands-on.

If you’re new to Linux, you can quickly discover how to use DSL to take your data on the road, safely running your programs and personal environment on nearly any computer. Easily adapt DSL to run on anything from an alternative device (Internet appliance, hand-held, diskless PC, or mini-ITX system) to an older PC that might otherwise be headed for a landfill.