index

Symbols

<!DOCTYPE> declaration 79, 110
$db variable 105
$(document).ready function 256, 263

Numerics

3D drawing
  complex models for 301
  overview 297–298

A

abort() method 319, 332–333
about:flags page 395
accessibility 9–10
Accessible Rich Internet Applications. *See* ARIA
accuracy attribute 327
action, forms 52
activeCues array 401
addElement() method 321
addEventListener() method 86, 100, 102–103, 126
agile planning board example
  building process for 114–115
  business logic for 119–123
  handling updates in browser 123
  handling updates on server 124–125
  overview 113–114
  prerequisites for 114
  template page 118–119
Airplane Mode 161
Aliens game
  collision detection 224
  CSS file for 205–206
  defense shields 221–222
  game engine base 214–215
  Game Over screen 211
  heads-up display 225–227
  html page for 204
  keyboard controls 219–221
  lasers 223
  mouse controls 219–221
  overview 202–204
  player ship 218
  screen transitions 215–216
  Start screen 210–211
  UFOs
    animating 229
    creating big UFO 217–218
    dynamic movement of 230–231
    flocks of 227
    making UFOs shoot 232
    paths for 228
altitude attribute 327
altitudeAccuracy attribute 327
<animate> tag 202, 206, 208
animation
  Canvas Ricochet game 173
  for SVG 208
  UFOs in SVG Aliens game 229
Apache web server 347–350
API method 19, 26, 140–141
.appcache extension 158
app.js file 59, 73, 365–368, 370–371
application cache manifest file 132
  configuring MIME types for 157–158
  creating 158–160
  detecting changes in 160–162
arc() method 178
arcTo() method 179

ARIA (Accessible Rich Internet Applications) 4, 9–10
arrays 283
<aside> elements 8–9
attributes
  data-* attributes 46–47
  for form elements 12–13
  for input types 41–43
  max attribute 46
  min attribute 46
  pattern attribute 49–50
  required attribute 44
audio
  Audio API 21–22, 313
  <audio> element 22, 242
  browser compatibility 239
  converting between formats 252
  custom controls for 255–256
  Firefox browser 248
  Media Element interface
    events in 245–246
    overview 244
    properties of 245
  multiple formats with <source> element
    browser compatibility 248
    discovering which video is playing 249–251
  Opera browser 248
  prerequisites for 239–240
  reference links for 427
  telestrator jukebox application
    adding <canvas> element 253–254
    adding clear button 264
    adding frame around video 257–260
    adjusting video opacity 260
    capturing mouse movement 263
    custom playback controls 255–256
    determining state of media resources 245
    displaying captured path over video 263
    grayscale video 261–262
    layout for 240–241
    loading list of videos 243
    overview 238–240
    playing video on canplaythrough event 246
    starting video when selected 244
    switching between videos 244
    using event handlers to change video 245
    <video> element 241–242
auto-complete, using cross-document messaging 127–129
autofocus attribute 41, 43
autoplay attribute 324

B

background
  Canvas Ricochet game 174–175
  Full-Screen API 414
ball, Canvas Ricochet game
  creating 178–179
  moving 182
Ball.collide() method 184
Ball.draw() method 194
Ball.move() method 182
Battery API 418–420
bitmap graphics vs. SVG 200–202
bounding box 231
Brick.draw() method 191
bricks, Canvas Ricochet game
  coloring 177–178
  creating 175
  removing hit bricks 184–185
Bricks.collide() method 191
browsers
  compatibility
    Canvas API 169–170
    using <source> element 248
    video 239
  development versions 379–380
  for Geometry Destroyer game 270–271
  WebSocket protocol support 362–363
buffered attribute 324
buffers
  for color 287–288
  for dimension 287–288
  displaying shape data using 288
  for shape 287–288
build() method 216–217
bullets, Geometry Destroyer game 299–300
business logic 119–123
bypassing form validation 51

C

CACHE section 158–160
cached event 318
calculations in forms
  data-* attributes 46–47
  functions to calculate total values 53–54
  min and max attributes 46
  number input type 46
  <output> element 47, 57
  retrieving price values 56–57
  retrieving value of input fields 55
  valueAsNumber property 54
canplay event 245
canplaythrough event 245–246
canPlayType() method 324

- Canvas API
  - Canvas Ricochet game
    - animation for 173
    - background image 174–175
    - coloring bricks 177–178
    - creating ball 178–179
    - creating game bricks 175
    - creating paddle 179
    - enabling collision detection 184
    - enabling edge detection 183
    - engine object for 171–173
    - Game Over screen for 194–195
    - keyboard controls for 186, 188–189
    - making ball move 182
    - mouse control for 187
    - moving paddle 182
    - removing hit bricks 184–185
    - score counter for 190–191
    - storing high scores online 192
    - testing 180
    - touch support for 187
    - Welcome screen for 193–194
  - contents of canvas element 169
  - libraries for ImpactJS 195
  - overview 21, 165–166, 313
  - reference links for 426
  - setting context 166–169
  - vs. SVG
    - community support 232–233
    - documentation 232–233
    - JavaScript integration 233
    - pros and cons of 233–234
  - verifying browser support 169–170
- <canvas> element 20–21, 32, 168–169, 234, 253–254
- CanvasRenderingContext2D interface 168
- card_cvv2 field 49
- card_number field 49
- Cascading Style Sheets. *See* CSS
- channel messaging 129
  - adding JavaScript to first page 375
  - adding JavaScript to second page 376–377
  - creating example pages 375
  - cross-domain test environment 373–374
  - installing local development web server 374
- chargingchange 419
- chargingTime 418
- chargingtimechange 419
- chat application example
  - chat form 105–106
  - connecting to stream in browser 111
  - creating database 105
  - creating server stream 109–111
  - implementing login process 107–108
  - login form 106–107
  - overview 102–105
  - sending chat messages to server 108
  - storing messages in database 108–109
- chat.js file 109, 127
- chat.sql file 344
- checkEnclosure() method 230
- checking event 318
- checkIntersection() method 230
- checkValidity() method 60, 66, 318
- Chrome developer tools 379
- class attribute 135–137, 140
- clear() method 130, 323
- clearData() method 322
- clearWatch() method 326
- client-side validation 13
- cmp() method 328
- codecs, discovering which used in video 386
- collapsible content 14–15
- collide() method 184, 217
- collision detection
  - displaying shape data 290
  - Geometry Destroyer game 303
  - for items in canvas 184
  - SVG Aliens game 224
- color buffers 287–288
- combobox example 9
- Command+Shift+H keyboard shortcut 350
- community support
  - for Canvas 232–233
  - for SVG 232–233
- Configuration page, MySQL installer 342
- Constraint Validation API 317
  - *See also* validating forms
- contact details section, order form example 43
- contenteditable attribute 25, 79, 81
- context, setting for canvas 166–169
- continue() method 328
- controller attribute 324
- controls attribute 242, 324
- controls, video playback 255–256
- converting video formats 252
- coords attribute 327
- core.js file 279
- CORS (Cross Origin Resource Sharing) 324
- cp.core.draw() method 290
- Create File zone 97
- createEmptyItem() function 148
- createFormSubmit() method 91, 93
- createIndex() method 130, 146, 330
- createObjectStore() method 130, 146, 329
- createReader() method 334
- createWriter() method 333
- Cross Origin Resource Sharing. *See* CORS

cross-document messaging 102
  auto-complete using 127–129
  overview 24, 125–126
  using postMessage method 126–127
cross-domain environment 373–374
cross-domain messaging 125, 127
crossOrigin attribute 324
CSS (Cascading Style Sheets) 10
  controlling visibility of views using 137
  and HTML5 18
  overview 314
  pseudo-classes 61–62
Ctrl+Shift+_ keyboard shortcut 349
Ctrl.init() method 186–187
ctx.beginPath() method 179
ctx.closePath() method 179
cube particles, Geometry Destroyer game
  creating 305
  customizing 306–307
cube() method 302
cuechange event 404
currentSrc attribute 249–251, 324
currentTime attribute 255, 324
cursors 130, 150
customError attribute 318

D

data-* attributes 45–47
dataset property 56
Date object 54
defaultMuted attribute 325
defaultPlaybackRate attribute 256, 325
defense shields, SVG Aliens game 221–222
delete_worker() method 122
delete() method 154–155, 328
deleteDatabase() method 130, 155–156, 328
deleteTask() function 154
designMode property 25, 76, 78–81
<details> element 15, 74
Developer Default option 341
Device Orientation API
  locking orientation 417
  overview 415–417
dimension buffers 287–288
direction attribute 328
DirectoryReader object 88
dischargingTime 419
dischargingtimechange 419
displayBrowserFileList() function 88, 98
displayFileSystem() function 88–89
<div> elements 6–7, 10
document.execCommand() method 26
document.getElementById() method 57
document.getItems() method 323
DOM (Document Object Model)
  and HTML5 19–20
  pulling shader data from 285–286
downloading event 318
Drag and Drop API 70, 96–97
  exporting files using 98–99
  importing files using 97–98
  overview 22–23
  reference for 321–322
drag event 322
dragend event 322
dragenter event 322
dragFile() function 98
draggable attribute 23, 70
dragleave event 322
Dragonfly 379
dragover event 97–98, 322
dragstart event 70, 98, 322
draw_welcome() function 408
draw() function 254, 260–261, 263–264, 419
drop event 322
dropDatabase() function 155–156, 160
dropEffect() method 321
duration attribute 255, 325
dynamic content, serving with Node.js
  creating static template with placeholders 367
  mixing dynamic content into template 368
  testing in browser 368–369

E

edge detection 183
edges() method 183
Editing API 320
effectAllowed() method 321
element.dataset.personName 56
email input type 41–43, 67
enableHighAccuracy attribute 326
encoding video with FFmpeg
  to MP4/h264 with AAC 387–388
  to MP4/h264 with MP3 388
  to Ogg/Theora 389
  to WebM/VP8 388–389
ended attribute 325
enemy, Geometry Destroyer game
  collision detection for 303
  creating 304–305
  generating random properties for 302
  overview 300
engine, custom WebGL
  base engine logic 277
  browser support for example 270–271
  entity helper methods 280–281
  entity storage 278
  index.html 271–272

engine, custom WebGL *(continued)*
  overview 269–270
  requestAnimationFrame() function 275
  shape entities with 3D data 279
  Simple JavaScript Inheritance script 275
  style.css 273
  sylvester.js 276
  time-saving scripts for 274–275
  webgl_util.js 276
error attribute 325
error event 319, 332–333
event-driven applications
  vs. polling 357–360
  server-side choices for 360
EventEmitter.js 114, 119–120
events
  changing videos using 245
  Media Element interface 245–246
  Text Track API 404–405
EventSource interface 25, 102, 111
execCommand() method 70, 78, 81, 320
exiting full-screen mode 415
Expiry Date field 65
exporting files using Drag and Drop API 98–99
eXtensible Messaging and Presence Protocol. *See* XMPP

F

failure of form validation
  detecting with invalid event 60–61
  styling invalid elements 61–62
FALLBACK section 159
fallbackValidation() function 68
favicon.ico 368
FFmpeg tool
  determining supported codecs 387
  discovering codecs used on video 386
  encoding with
    to MP4/h264 with AAC 387–388
    to MP4/h264 with MP3 388
    to Ogg/Theora 389
    to WebM/VP8 388–389
  obtaining 386
File API 30–31
File Browser view 72–73, 92, 97
File Editor view 72–73, 77–78, 86, 91, 94
File Reader API 30–31
File System API 70, 72–73
  creating files 90–91
  creating persistent filesystem 86–87
  deleting files 90–91
  getting list of files 87–89
  importing files from computer 92–94
  loading files in editor 89–90
  overview 30–31
  reference for 331–335
  saving files 94–96
  viewing files 90–91
File Writer API 30–31, 94–96
FileSaver() method 333
fileSystem field 86
Firebug 379
Firefox browser 248, 270–271
FLV (Flash video) format 21
follow_mouse() function 412, 416, 423
forever-frame hack 105
<form> element 41
formaction attribute 36, 38, 51–52
formats, video
  browser compatibility using multiple 248
  converting between 252
  discovering which is being played 249–251
formnovalidate attribute 36, 38, 51
forms
  action for 52
  fallbacks for unsupported browsers
    detecting features with Modernizr 63–64
    month-picker fallback 64–65
    validating form without constraint validation API 65–68
  input types
    attributes for 41–43
    email input type 41–43
    month input type 49
    pattern attribute 49–50
    required attribute for 44
    tel input type 41–43
  new features
    client-side validation 13
    new form attributes 12–13
    new form input types 11
  order form example
    basic document structure 40–41
    contact details section 43
    login details section 44
    order details section 47
    overview 38–39
    payment details section 50
    prerequisites for 39–40
  performing calculations in
    data-* attributes 46–47
    functions to calculate total values 53–54
    min and max attributes 46
    number input type 46
    <output> element 47, 57
    retrieving price values 56–57
    retrieving value of input fields 55
    valueAsNumber property 54
  reference links for 425

forms *(continued)*
validating
bypassing to save form details 51
custom validation 59–60
detecting failure with invalid event 60–61
styling invalid elements 61–62
fragment shader 283
frame around video 257–260
framebuffer 283
fs.readFile() method 366
Full-Screen API
entering full-screen mode 414
exiting full-screen mode 415
styling background 414
future HTML developments
Battery API 418–420
Device Orientation API
locking orientation 417
overview 415–417
Full-Screen API
entering full-screen mode 414
exiting full-screen mode 415
styling background 414
Mouse Event Capture API 411–413
Pointer Lock API 420–423
real-time media access and communication
getUserMedia() function 391–393
overview 390–391
WebRTC specification 394
test bed for exploring 406–411
Text Track API
adding multiple track elements 397–400
adding track element 396–397
checking to see if track is loaded 403–404
events for 404–405
loading text tracks in advance 403
styling for 405–406
Vibration API 417–418

G

Game Over screen
for Canvas Ricochet game 194–195
for SVG Aliens game 211
Game.draw() method 190
Game.init() method 190, 193
game.js file 280
Game.levelLimit() method 191
Game.levelUp() method 191
Game.restartGame() method 195
Game.runGame() method 193
Game.setup() method 173, 186
gd.core.animate() method 305
gd.core.draw() method 288
gd.core.shader.get() method 286
gd.core.shader.init() method 286
gd.core.shader.store() method 286
gd.gl.vertexAttribPointer() method 289
gd.template.Cube.init() method 306
gd.template.Entity.color() method 287
gd.template.Entity.indices() method 287
gd.template.Entity.shape() method 287
gd.template.Polygon 301–304
Geolocation API
inserting map of user location 83–84
overview 29, 82–83, 315
reference for 326–327
Geometry Destroyer game
3D cube particle 305
3D Polygon enemy 300
animating player entity 298
capturing user input 294
collision detection for enemies 303
creating enemies 304–305
creating player entity 296
customizing cubes 306–307
generating random enemy properties 302
HUD 295
player bullets 299–300
square particles 308
get_task() method 122
getAsFile() method 322
getAsString() method 322
getAttribute() method 56
getAttributeNS() method 213
getBBox() method 231
getBlob() method 335
getContext() method 381
getCueAsHTML() method 402
getCurrentPosition() method 70, 82, 326
getData() method 322
getDirectory() method 334
getEnclosureList() method 230
getFile() method 89–90, 334
getImageData() method 261–262
getIntersectionList() method 230
getItem() method 130, 140, 323
getUserMedia() method 391–393
Global Positioning System. *See* GPS
globalAlpha property 260
globalCompositeOperation property 258
go() function 407–408, 410, 414, 420, 422
Google Chrome 270–271
GPS (Global Positioning System) 326
gradient() method 177
graphics cards 283–284
grayscale video 261–262

H

HAVE_CURRENT_DATA property 246
HAVE_ENOUGH_DATA property 246
HAVE_FUTURE_DATA property 246
HAVE_METADATA property 246
HAVE_NOTHING property 246
headers, HTTP 355–357
heading attribute 327
heads-up display. *See* HUD
height attibute 242
Hello World application using Node.js 365–366
history of HTML5 310–311
home page, MySQL Workbench 344
hp (hit points) 221
.htaccess file 157
<html> element 5, 382
HTML5 (HyperText Markup Language 5)
- Audio and Video APIs 21–22
- Canvas API 21
- collapsible content without JavaScript 14–15
- cross-document messaging 24
- and CSS3 18
- and DOM 19–20
- Drag and Drop API 22–23
- enhancing accessibility using ARIA roles 9–10
- File API 30–31
- File Reader API 30–31
- File System API 30–31
- File Writer API 30–31
- forms
  - client-side validation 13
  - new form attributes 12–13
  - new form input types 11
- Geolocation API 29
- history of 310–311
- IndexedDB API 29–30
- Internet Explorer support
  - overview 10
  - printing new elements properly 11
  - rendering new elements properly 10
- and JavaScript 19–20
- meters 14
- microdata 16–18
- offline web applications 27–28
- progress bars 14
- rich-text editing of documents 25–26
- semantic elements in 6–9
- server-sent events 25
- specifications for 316
- structure of documents 5–6
- SVG 31
- technologies not officially part of 312
- tools for
  - browser development versions 379–380
  - Chrome developer tools 379
  - Dragonfly 379
  - Firebug 379
  - HTML5 Boilerplate 382
  - HTML5 Shiv 379–380
  - IE developer tools 379
  - jsFiddle 383–384
  - Modernizr 380–382
  - Safari developer tools 379
- Web Storage API 25–27
- WebGL 32
- WHATWG vs. W3C 311–312
- *See also* future HTML developments

HTML5 Boilerplate 382
HTML5 Shiv 379–380
HTTP headers 355–357
httpd.conf file 347–348, 350
httpd.userdir.conf file 350
HUD (heads-up display) 190
- Geometry Destroyer game 295
- SVG Aliens game 225–227

Hud.update.level() method 227
HyperText Markup Language 5. *See* HTML5

I

IDBOpenDBRequest interface 330
IDE (integrated development environment) 379
IETF (Internet Engineering Task Force) 360
if statement 169
ImpactJS library 195
importing files
- using Drag and Drop API 97–98
- using File System API 92–94
- with XLink 209

IndexedDB API 132
- adding data to 152–154
- connecting to 145–148
- deleting data 154–155
- detecting support for 144–145
- dropping database 155–156
- overview 29–30
- reference for 327–330
- searching 150–152
- updating data 154–155

index.php file 106–107
init() method 182, 216–217, 276, 280, 299, 306
innerHTML property 57
input types
- attributes for 41–43
- email input type 41–43
- month input type 49
- number input type 46

input types *(continued)*
  pattern attribute 49–50
  required attribute for 44
  tel input type 41–43
<input> element 36, 38, 40, 47, 59, 65, 331
INSERT statement 153
insertImage() function 83
insertTask() function 153
integrated development environment. *See* IDE
Internet Engineering Task Force. *See* IETF
Internet Explorer
  developer tools 379
  supporting new elements in
    overview 10
    printing new elements properly 11
    rendering new elements properly 10
invalid event 36, 58, 60–61
IP (Internet Protocol) 355
isDirty variable 80

J

JavaScript
  collapsible content without using 14–15
  and HTML5 19–20
  interaction with Canvas 233
  interaction with SVG
    vs. Canvas 233
    overview 212–214
  libraries 315–316
  navigating between views using 137–139
JavaScript Object Notation. *See* JSON
jQuery 315–316
jsFiddle 383–384
JSON (JavaScript Object Notation) 19

K

key attribute 328
key() method 323
keyboard controls
  Canvas Ricochet game 186, 188–189
  SVG Aliens game 219–221
kind attribute 397, 399–401

L

lasers, SVG Aliens game 223
Laser.update() method 224
latency 357
latitude attribute 327
levelchange 419
<li> element 98, 249
libraries, Canvas API
  ImpactJS 195

lineTo() method 179
links
  for Canvas API 426
  for messaging 425–426
  for mobile applications 426
  for SVG 426
  for video and audio 427
  for WebGL 427
list directory command. *See* ls command
load event 332
load() method 325
loadeddata event 245
loadedmetadata event 245
loadend event 332
loadFile() function 89–90
loadSettings() function 140–141
loadstart event 332
loadTasks() function 146, 150–151
localStorage
  deleting data from 141–143
  reading data from 140–141
  saving data to 141
localStorage interface 27, 139–140
location.hash property 133, 138
locking orientation 417
login details section, order form example 44
longitude attribute 327
loop attribute 325
ls (list directory) command 347

M

Mac OS X
  MySQL
    configuring PHP for 352–353
    installing 351–352
  PHP
    configuring MySQL for 352–353
    editing Apache configuration files 347–350
    overview 345–346
    serving web files from sites directory 350–351
    using Terminal app 346–347
makeGradient() method 177
maps, user location on 83–84
<mark> elements 8–9
markAsComplete event 149, 154
matrices 288
max attribute 13, 45–46
maximumAge attribute 327
media
  adjusting opacity of 260
  browser compatibility 239
  converting between formats 252
  custom controls for 255–256
  and Firefox browser 248

media *(continued)*
  rayscale 261–262
  Media Element API
    events in 245–246
    overview 244
    properties of 245
  multiple formats with <source> element
    browser compatibility 248
    discovering which video is playing 249–251
  and Opera browser 248
  prerequisites for 239–240
  telestrator jukebox application
    adding <canvas> element 253–254
    adding clear button 264
    adding frame around video 257–260
    adjusting video opacity 260
    capturing mouse movement 263
    custom playback controls 255–256
    determining state of media resources 245
    displaying captured path over video 263
    grayscale video 261–262
    layout for 240–241
    loading list of videos 243
    overview 238–240
    playing video on canplaythrough event 246
    starting video when selected 244
    switching between videos 244
    using event handlers to change video 245
    <video> element 241–242
Media Element API
  events in 245–246
  overview 244
  properties of 245
  reference for 324
mediaGroup attribute 325
<menu> element 399
messaging
  agile planning board example
    building process for 114–115
    business logic for 119–123
    handling updates in browser 123
    handling updates on server 124–125
    overview 113–114
    prerequisites for 114
    template page 118–119
  chat application example
    chat form 105–106
    connecting to stream in browser 111
    creating database 105
    creating server stream 109–111
    implementing login process 107–108
    login form 106–107
    overview 102–105
    sending chat messages to server 108
    storing messages in database 108–109
  cross-document messaging
    auto-complete using 127–129
    overview 125–126
    using postMessage method 126–127
  reference links for 426
  server-sent events 112
  WebSockets 115–117
<meter> element 14
meters 14–15
microdata
  API reference 323
  overview 16–18
Microsoft Internet Explorer. *See* Internet Explorer
Microsoft Windows 7. *See* Windows 7
MIME types 31, 157–158
mime.types file 158
min attribute 13, 45–46
mobile applications
  IndexedDB API
    adding data to 152–154
    connecting to 145–148
    detecting support for 144–145
    dropping database 155–156
    searching 150–152
    updating and deleting data 154–155
  My Tasks application example
    adding new tasks 153
    controlling visibility of views using CSS 137
    creating views with <section> elements 136
    deleting tasks 154
    document structure for 135
    implementing navigation with JavaScript 137–139
    navigation bar 135
    overview 134–135
    searching task view list 152
    task list view 148–149
    updating tasks 154
  offline applications
    configuring MIME type on server 157–158
    creating application cache manifest file 158–160
    detecting changes in manifest file 160–162
  reference links for 426
  tools for 378
  Web Storage API
    deleting data from localStorage 141–143
    reading data from localStorage 140–141
    saving data to localStorage 141
Modernizr 380–382
month input type 49, 64–65
monthpicker.js file 41, 64
mouse controls
  Canvas Ricochet game 187
  SVG Aliens game 219–221

Mouse Event Capture API 411–413
movePaddle() method 187
moveTask message 124
Mozilla Firefox. *See* Firefox browser
MP4/h264 with AAC encoding 387–388
MP4/h264 with MP3 encoding 388
multiEntry flag 146
multimedia 313
multiple attribute 92
muted attribute 325
My Tasks application example 132–133, 161
  adding new tasks 153
  controlling visibility of views using CSS 137
  creating views with <section> elements 136
  deleting tasks 154
  document structure for 135
  implementing navigation with JavaScript 137–139
  navigation bar 135
  overview 134–135
  searching task view list 152
  task list view 148–149
  updating tasks 154
MySQL
  on Mac OS X
    configuring PHP for 352–353
    installing 351–352
  on Windows 7
    creating databases 343–346
    downloading 340–341
    installing 341–343
    running scripts 343–346
mysql.sock file 352

N

navigating between views using JavaScript 137–139
NETWORK section 159
NETWORK_EMPTY property 245
NETWORK_IDLE property 246
NETWORK_LOADING property 246
NETWORK_NO_SOURCE property 246
networking
  event-driven applications
    vs. polling 357–360
    server-side choices for 360
  HTTP headers 355–357
  latency 357
  overview 354–355
  throughput 357
  WebSocket protocol
    browser support for 362–363
    overview 361–362
    vs. WebSocket API 360–361
networkState attribute 245, 325
node_modules directory 370
Node.js 102, 112–115, 117–118, 129
  creating socket using 115–117
  creating web apps with 370–372
  Hello World application using
    creating 365
    running 365–366
  overview 315
  requirements for 364–365
  routing with 369–370
  serving mixed static and dynamic content with
    creating static template with placeholders 367
    mixing dynamic content into template 368
    testing in browser 368–369
  serving static files with
    loading file from disk 366
    running application 367
    sending file to browser 367
noupdate event 318
novalidate attribute 51
number input type 45
  overview 46
  retrieving value of 55

O

objectStore() method 152, 154
obsolete event 319
offline applications
  API reference 318–319
  configuring MIME type on server 157–158
  creating application cache manifest file 158–160
  detecting changes in manifest file 160–162
  overview 27–28
Ogg/Theora video format 251, 389
oncuechange property 404
onmessage event 116, 126, 128, 375–376
onsuccess property 331
onupgradeneeded event 330
opacity of video 260
Open Connection to Start Querying heading 344
open() method 130
openCursor() method 130, 150, 329–330
OpenGL ES (Open Graphics Library for Embedded Systems) 32, 284
Opera browser 248
options object 146
or operators 144
order form example
  basic document structure 40–41
  contact details section 43
  login details section 44
  order details section 47

order form example *(continued)*
overview 38–39
payment details section 50
prerequisites for 39–40
orientationchange event 140
<output> element 40, 47, 57

P

<p> tag 188
paddle, Canvas Ricochet game
creating 179
moving 182
Paddle.move() method 182–183
parseFloat function 55
paths
drawing in SVG 209–210
for UFOs in SVG Aliens game 228
pattern attribute 13, 49–50
patternMismatch attribute 318
pause() method 256, 325
paused attribute 325
payment details section, order form example 49–50
PDO database 109
persistent filesystem 86–87
PHP
files 106–107
on Mac OS X Mountain Lion
configuring MySQL for 352–353
editing Apache configuration files 347–350
overview 345–346
serving web files from sites directory 350–351
using Terminal app 346–347
on Windows 7
configuring Windows 7 IIS 336–337
confirming installation 339
downloading 337
installing 338–339
placeholder attribute 13, 41, 43
planner.get_words() method 128
plan.on() method 123
play() method 325
playbackRate attribute 256, 325
played attribute 325
Pointer Lock API 420–423
pointerlockchange 420
pointerLockElement 420
pointerlockerror 420
polling vs. event-driven applications 357–360
Polygon.collide() method 308
postMessage() method 100, 126–127, 375
preload attribute 325
primaryKey attribute 328
printing new HTML5 elements correctly in IE 11
progress bars 14
progress event 318, 332–333
pseudo-classes 61–62
putImageData() method 261
python command 134

Q

quantum entanglement 375
queryCommandEnabled() method 320
queryCommandIndeterm() method 320
queryCommandState() method 70, 82, 320
queryCommandSupported() method 320
queryCommandValue() method 320
queryUsageAndQuota() method 70
Quota Management API 70, 85–87

R

randomMeta() method 302
rangeOverflow attribute 318
rangeUnderflow attribute 318
RDBMS (relational database management system) 145
readAsArrayBuffer() method 332
readAsText() method 70, 89, 332
readEntries() method 334
readyState attribute 245–246, 325
real-time media access and communication
getUserMedia() function 391–393
overview 390–391
WebRTC specification 394
reference links
for Canvas API 426
for forms 425
for messaging 426
for mobile applications 426
for SVG 426
for video and audio 427
for WebGL 427
relational database management system. *See* RDBMS
releaseCapture() method 411
remove() method 90
removeItem() method 130, 141, 323
removeRecursively() method 334
requestAnimationFrame() method 173, 240, 274–275, 305, 407
requestFullscreen() method 414
requestPointerLock() method 420
requestQuota() method 70, 266

required attribute 13, 41, 43–44, 65, 68
resetSettings() function 155–156
rich-text editing of documents 25–26, 81
Ricochet game (Canvas)
  animation for 173
  background image 174–175
  coloring bricks 177–178
  creating ball 178–179
  creating game bricks 175
  creating paddle 179
  enabling collision detection 184
  enabling edge detection 183
  engine object for 171–173
  Game Over screen for 194–195
  keyboard controls for 186, 188–189
  making ball move 182
  mouse control for 187
  moving paddle 182
  removing hit bricks 184–185
  score counter for 190–191
  storing high scores online 192
  testing 180
  touch support for 187
  Welcome screen for 193–194
rotation logic 291
routing with Node.js 369–370
run.js file 271, 278, 280–282

S

Safari developer tools 379
Save Settings button 143
saveBtn.addEventListener() method 53
Scalable Vector Graphics. *See* SVG
scores, game
  counter for 190–191
  storing online 192
screen transitions 215–216
Screen.gameover() method 194
script contexts 125
<script> element 19, 116, 375–376
searching using IndexedDB API 150–152
<section> element 136
seek() method 333
seekable attribute 325
seeking attribute 326
self.applicationCache() method 319
semantic elements 4, 6–9
server-sent events. *See* SSE
session_id() method 107–108
session_start() method 110
sessionStorage interface 27, 139
setCapture() method 411–412
setCustomValidity() method 36, 59, 318
setData() method 322
setDragImage() method 321
setInterval() method 173, 305
setItem() method 130, 323
setTimeout() method 305
setversion() method 329
shaders
  creating via OpenGL ES 284
  pulling shader data from DOM 285–286
shapes
  buffers for 287–288
  creating in SVG 206–208
Ship object 219
showTask() function 148–150
showUI argument 81
Simple JavaScript Inheritance script 275
SimpleHTTPServer module 374
source attribute 328
<source> element 248
speed attribute 327
spinbox component 46
SQL database 150, 156
SQLite database 30
src attribute 242, 244, 248, 250, 326
srclang attribute 399, 401
SSE (server-sent events) 25, 102–103, 129, 314, 360
  chat application example
    chat form 105–106
    connecting to stream in browser 111
    creating database 105
    creating server stream 109–111
    implementing login process 107–108
    login form 106–107
    overview 102–105
    sending chat messages to server 108
    storing messages in database 108–109
  when to use 112
Start screen 210–211
startDate attribute 326
state management 76–78
static files
  loading file from disk 366
  running application 367
  sending file to browser 367
stepMismatch attribute 318
Storage APIs 132, 314–315
storageInfo field 86
streams
  connecting to stream in browser 111
  creating server stream 109–111
String.Replace() function 368
<style> element 253
style.css file 73, 205
sudo (switch user and do) command 348
<summary> element 15, 74

Super HTML5 Editor example
  designMode property 79–80
  Drag and Drop API
    exporting files using 98–99
    importing files using 97–98
  execCommand method 81
  file browser view 74
  file editor view 75–76
  File System API
    creating files 90–91
    creating persistent filesystem 86–87
    deleting files 90–91
    getting list of files 87–89
    importing files from computer 92–94
    loading files in editor 89–90
    saving files 94–96
    viewing files 90–91
  File Writer API 94–96
  Geolocation API
    inserting map of user location 83–84
    overview 82–83
  main document structure 74
  navigating between views 76–78
  overview 72–74
  reference links for 425–426
  rich-text editing toolbar 81
  state management for 76–78
SuperEditor() function 77
SVG (Scalable Vector Graphics)
  adding shapes 206–208
  adding text 208
  Aliens game
    animating UFOS 229
    big UFO 217–218
    collision detection 224
    CSS file for 205–206
    defense shields 221–222
    dynamic movement of UFOs 230–231
    game engine base 214–215
    Game Over screen 211
    heads-up display 225–227
    keyboard controls 219–221
    lasers 223
    making UFOs shoot 232
    mouse controls 219–221
    overview 202–204
    player ship 218
    screen transitions 215–216
    Start screen 210–211
    svg tags in html 204
    UFO flocks 227
    UFO paths 228
  animation 208
  API overview 313
  vs. bitmap graphics 200–202
  vs. Canvas
    community support 232–233
    documentation 232–233
    JavaScript integration 233
    pros and cons of 233–234
  drawing with paths 209–210
  importing files with XLink 209
  JavaScript interaction with 212–214
  overview 31, 199–200
  reference links for 426
  <svg> element 202–204, 207, 210, 212
  viewBox parameter 211–212
  Wilson smiley example 201–202
swapCache() method 130, 160, 318–319
switch user and do command. *See* sudo command
sylvester.js file 276

T

TCP (Transmission Control Protocol) 355
tel input type 41–43
telestrator jukebox application
  adding <canvas> element 253–254
  adding clear button 264
  adding frame around video 257–260
  adjusting video opacity 260
  capturing mouse movement 263
  custom playback controls 255–256
  determining state of media resources 245
  displaying captured path over video 263
  grayscale video 261–262
  layout for 240–241
  loading list of videos 243
  overview 238–240
  playing video on canplaythrough event 246
  starting video when selected 244
  switching between videos 244
  using event handlers to change video 245
  <video> element 241–242
template page, agile planning board example 118–119
Terminal app 346–347
test bed for exploring future HTML developments 406–411
text property 401, 403
Text Track API 395–396, 400–401
  adding multiple track elements 397–400
  adding track element 396–397
  checking to see if track is loaded 403–404
  events for 404–405
  loading text tracks in advance 403
  styling for 405–406
text, in SVG 208
text/event-stream type 109
<text> element 208

<textarea> element 25
throughput 357
timeout attribute 326
timestamp attribute 327
title attribute 50
tooLong attribute 318
tools
  for HTML5 applications
    browser development versions 379–380
    Chrome developer tools 379
    Dragonfly 379
    Firebug 379
    HTML5 Boilerplate 382
    HTML5 Shiv 379–380
    IE developer tools 379
    jsFiddle 383–384
    Modernizr 380–382
    Safari developer tools 379
  for mobile web applications 378
touch support 187
toURL method 90–91, 99
<track> element
  adding 396–397
  adding multiple 397–400
transaction() method 329
transcoder 252
Transmission Control Protocol. *See* TCP
triangle assembler 283
truncate() method 333
typeMismatch attribute 318

U

UFOs in SVG Aliens game
  animating 229
  creating big UFO 217–218
  dynamic movement of 230–231
  flocks of 227
  making UFOs shoot 232
  paths for 228
UI (user interface) 378
update_xy() function 417
update() method 215, 231–232, 298, 328
updateBrowserFilesList() function 88–89, 91
updateready event 160, 162, 318
upgradedNeeded event 145
user input 294
user interface. *See* UI

V

valid attribute 318
validateForm() function 68
validating data 125
validating forms
  bypassing to save form details 51
  custom validation 59–60
  detecting failure with invalid event 60–61
  new features for 13
  reference links for 425
  styling invalid elements 61–62
validationMessage attribute 318
validity attribute 318
value property 54
valueAsDate property 54
valueAsNumber property 38, 53–55
valueMissing attribute 318
Vector Markup Language. *See* VML
vertex shader 283
Vibration API 417–418
video
  adjusting opacity of 260
  API overview 21–22, 313
  browser compatibility 239
  converting between formats 252
  custom controls for 255–256
  encoding with FFmpeg tool
    to MP4/h264 with AAC 387–388
    to MP4/h264 with MP3 388
    to Ogg/Theora 389
    to WebM/VP8 388–389
  Firefox browser 248
  grayscale 261–262
  Media Element interface
    events in 245–246
    overview 244
    properties of 245
  multiple formats with <source> element
    discovering which video is playing 249–251
    for browser compatibility 247–248
  Opera browser 248
  prerequisites for 239–240
  reference links for 427
  telestrator jukebox application
    adding <canvas> element 253–254
    adding clear button 264
    adding frame around video 257–260
    adjusting video opacity 260
    capturing mouse movement 263
    custom playback controls 255–256
    determining state of media resources 245
    displaying captured path over video 263
    grayscale video 261–262
    layout for 240–241
    loading list of videos 243
    overview 238–240
    playing video on canplaythrough event 246
    starting video when selected 244

video *(continued)*
switching between videos 244
using event handlers to change video 245
<video> element 22, 241–242, 396, 399
viewBox property 203, 211–212, 230
views
controlling visibility of using CSS 137
creating with <section> elements 136
navigating between using JavaScript 137–139
VML (Vector Markup Language) 213
volume attribute 326

W

W3C (World Wide Web Consortium) 311–312
WAI (Web Accessibility Initiative) 9
watchPosition() method 326
Web Hypertext Application Technology Working Group. *See* WHATWG
Web Messaging API 314
Web Socket API 314
Web Storage API
deleting data from localStorage 141–143
overview 26–27
reading data from localStorage 140–141
reference for 323
saving data to localStorage 141
WebGL (Web Graphics Library)
3D drawing basics 297–298
building engine for
base engine logic 277
browser support for example 270–271
entity helper methods 280–281
entity storage 278
index.html 271–272
overview 269–270
requestAnimationFrame() function 275
shape entities with 3D data 279
Simple JavaScript Inheritance script 275
style.css 273
sylvester.js 276
time-saving scripts for 274–275
webgl_util.js 276
complex 3D models 301
creating buffers 287–288
displaying shape data on screen
adding matrix helpers 291
adding rotation logic 291
binding and drawing shapes 289
collision detection 290
using matrices and buffers 288
Geometry Destroyer game
3D cube particle 305
3D Polygon enemy 300
animating player entity 298
capturing user input 294
collision detection for enemies 303
creating enemies 304–305
creating player entity 296
customizing cubes 306–307
generating random enemy properties 302
HUD 295
player bullets 299–300
square particles 308
graphics cards 283–284
overview 32, 315
reference links for 427
shaders for 3D data
creating via OpenGL ES 284
pulling shader data from DOM 285–286
webgl_util.js file 276
WebM/VP8 encoding 388–389
WebRTC specification 394
WebSocket API
agile planning board example
building process for 114–115
business logic for 119–123
handling updates in browser 123
handling updates on server 124–125
overview 113–114
prerequisites for 114
template page 118–119
creating with Node.js 115–117
overview 25
vs. WebSocket protocol 360–361
WebSocket protocol
browser support for 362–363
overview 361–362
vs. WebSocket API 360–361
WebSocket-Node module 115, 117
websocket-sample.js file 117
WebVTT (Web Video Text Tracks) file format 396
Welcome screen 193–194
WHATWG (Web Hypertext Application Technology Working Group) 19, 311–312
width attribute 242
willValidate attribute 318
Wilson smiley example 201–202
wilson.svg file 201
window event 185
window.location.reload() method 160
Windows 7
MySQL
creating databases 343–346
downloading 340–341
installing 341–343
running scripts 343–346

Windows 7 *(continued)*
  PHP
    configuring Windows 7 IIS 336–337
    confirming installation 339
    downloading 337
    installing 338–339
World Wide Web Consortium. *See* W3C
write() method 70, 333
writeend event 333
writestart event 333
ws.onmessage 123

X

XHR (XML HTTP Request) object 4, 314
XLink 209
XMPP (eXtensible Messaging and Presence Protocol) 360

Y

yieldForStorageUpdates() method 324