

INDEX

A

- `<a>` element, 54
- accessibility. *See also* WAI-ARIA
 - canvas element, 139
 - dragging and dropping, 184–185
 - multimedia, 110–113
 - outlining algorithm, 36–37
- Accessible Rich Internet Applications. *See* WAI-ARIA
- `addEventListener` method, 106–110, 199, 208
- `<address>` element, 58
- animating paintings, 134–137
- APIs, retained-mode *versus* immediate mode, 124
- `<applet>` element, 60
- ARIA (Accessible Rich Internet Applications).
 See WAI-ARIA
- `aria-*` attribute, 63
- `aria-grabbed` attribute, 185
- `aria-required` attribute, 76
- `aria-valuenow` attribute, 81–82
- `<article>` element, 20–21, 37–42, 52, 54, 58, 111
 - block-level links, 38
 - comments as nested articles, 29–30
- Asian languages, 55
- `<aside>` element, 17, 19–20, 33, 52, 54
- `attributes` attribute, 63
- Audacity software, 101
- `<audio>` element, 54, 94, 96, 99–100
- `autocomplete` attribute, 74, 78
- `autofocus` attribute, 75
- `autoplay` attribute, 95

B

- `` element, 59
- Baranovskiy, Dmitry, 124
- base64 encoded assets, 133
- `beginPath` method, 122–123
- `<big>` element, 60
- `<object>` element, 92–93
- `<blink>` element, 60
- block-level elements, 38, 54
- `<blockquote>` element, 28, 34–35
- `<body>` element, 3–4, 5, 27–28, 34
- boldface, `` element, 59
- bug reports, 12
- `<button>` element, 54, 68

C

- Camem, Kroc, 100
- `cancelEvent` function, 179
- `canplaythrough` and `canplay` events, 108
- `canPlayType` method, 102–103
- `<canvas>` element/canvases, 54
 - accessibility, 139
 - animating paintings, 134–137
 - basics, 118–119
 - capturing images, 126–129
 - data URLs, saving to, 132–133
 - drawing applications, 115–116
 - drawing state, 137
 - fill styles, gradients and patterns, 118–122
 - Harmony application, 115, 117
 - MS Paint replication, 115–116
 - paths, 122–124
 - pixels, pushing, 130–132
 - rectangles, 118
 - gradients and patterns, 118–120
 - rendering text, 138–139
 - resizing canvases, 122
 - transformation methods, 124–126
- case sensitivity, `pattern` attribute, 78
- `<center>` element, 60
- character encoding, UTF-8, 2
- `charset="utf-8"` attribute, XHTML and XML *versus* HTML5, 2
- `checkValidity` attribute, 86
- `checkValidity` method, 85–86
- Chisholm, Wendy, 51
- `cite` attribute, 28
- `<cite>` element, 58
- classes
 - attributes, 6, 8
 - names, Google index research, 6
- `clear` attribute, 147
- `clearInterval` method, 127
- `clearRect` method, 125
- `clearWatch` method, 190
- codecs, 98–99
- `color` input type, 74
- Comet, 212, 215
- `<command>` element, 62, 65
- comments as nested articles, 29–30

Contacts API, 70
 <content> element, 9
 content models, 54
 contentEditable attribute, 61
 contentWindow object, 199
 context object, canvas attribute, 126
 contextmenu attribute, 62
 controls attribute, 54, 95–96
 cookies, 142–143
 Coordinated Universal Time (UTC), 26
 coords object, 191
 copyrights, <small> element, 18, 24, 60
 Cotton, Paul, xii
 createElement method, 121
 createPattern method, 119–121, 126
 createRadialGradient method, 120
 Crockford, Douglas, 148
 CSS (Cascading Style Sheets), 10
 <body> element requirement, 11
 display:inline, 54
 headers and footers for body and articles, 27–28
 IE, 5, 11–12
 outlines, 35–36
 WAI-ARIA, 50
 CSS Basic User Interface module, 83
 CSS Media Queries specification, 102

D

data-* attribute, 62, 112
 data storage
 cookies, 142
 Web SQL Databases, 142, 152–162
 Web Storage API, 142–151
 data URLs, 132–133
 <datalist> element, 74–75
 date input type, 70–71
 dates, machine-readable, 26
 datetime attribute, 26
 datetime input type, 71
 Davis, Daniel, 55
 <dd> element, 57
 definition lists, 57
 element, 54
 delete method, 68
Designing with Progressive Enhancement: Building the Web that Works for Everyone, 51
 <details> element, 34, 52–54
 <device> element, 94
 disclaimers, <small> element, 18, 24, 60
 display:block, 12
 display:inline, CSS, 54
 <div> element, HTML 4, 7–8

<dl> element, 57
 DOCTYPE, 2
 <!doctype html> tags, 2
 dragend event, 184
 draggable attribute, 62
 dragging and dropping
 accessibility, 184–185
 basics, 176–179
 custom drag icons, 183
 dragged data, interoperability, 180–182
 enabling elements for dragging, 182–183
 DragonFly plug-in, 150
 dragover event, 178
 dragstart event, 179, 183–185
 draw function, 136
 drawImage method, 126–130
 dropEffect method, 185
 <dt> element, 57
 durationchange event, 108

E

Eich, Brendan, xii
 element, 54–55, 58–60
 email input type, 69–70, 82
 <embed> element, 54, 64, 92–93
 embedded content models, 54
 emphasis effect, 54–55, 58–59
 enableHighAccuracy method, 194
 end method, 107
 error handling, 192–193
 event object, 198–199
 executeSql method, 154, 158–161

F

“fat footers,” 19–20
 Faulkner, Steve, 50
 ffmpeg library, 101
 <fieldset> element, 34, 68, 86
 <figcaption> element, 53
 <figure> element, 34, 53
 fill styles, gradients and patterns, 118–122
 fillRect method, 119
 fillStyle method, 119–121
 fillText method, 138–139
 Firebug plug-in, 149
 Firefogg software, 101
 Firefox Contacts addon, 70
 flow content models, 54
 element, 60
 <footer> element, 16, 18–20, 28
 forEach method, 156

- form attribute, 68
- `<form>` element/forms
 - `<button>` element, 68
 - comments, 79
 - `<datalist>` element, 74–75
 - date pickers, 83
 - delete, 68
 - `<fieldset>` element, 68
 - form fields, 83
 - get, 68
 - `<input>` element, 68
 - onchange, 81
 - type=...autocomplete, 74, 78
 - type=...autofocus, 75
 - type=color, 74
 - type=date, 70–71
 - type=datettime, 71
 - type=email, 68–69, 82
 - type=...list, 74–75
 - type=...max, 74, 78
 - type=...min, 74, 78
 - type=month, 71
 - type=...multiple, 69, 74, 76
 - type=number, 72, 82
 - type=...pattern, 74, 76–78
 - type=...placeholder, 75–76
 - type=range, 72–73, 80
 - type=...required, 69, 76
 - type=search, 73
 - type=...step, 74, 78–79
 - type=tel, 73, 82
 - type=text, 68–69
 - type=time, 71
 - type=url, 70, 82
 - type=week, 72
 - update, 68
 - validation
 - built-in, 68
 - JavaScript, 68
 - `<keygen>` element, 68
 - `<label>` element, 68
 - `<meter>` element, 68, 80
 - new types, 68
 - `<object>` element, 68
 - `<output>` element, 68
 - post, 68
 - `<progress>` element, 68, 80
 - `<select>` element, 68
 - sliders with values, 80–83
 - `<textarea>` element, 68
 - validation elements, 85–86
 - formats, consistent use, 3

- formnovalidate attribute, 87
- frames, removed from HTML5, 60
- furigana/ruby, 55

G

- geolocation API, xii, 187–195
- get method, 68
- getAttribute method, 112
- getCurrentPosition method, 189–194
- getData method, 178–180
- getImageData method, 130–132
- getItem method, 146–148, 151
- getTime method, 156
- getTweets method, 156
- “The Guardian” case study, 42–47

H

- h1..h6 elements, 54
- H.264 specification, 98–101
- Harmony application, 115, 117
- `<head>` element, 2–4, 12
- `<header>` element, 13–15
- heading content models, 54
- height attribute, 96–97
- `<hgroup>` element, 13, 33–34
- Hickson, Ian, iii, xi–xiii, 6, 175
- hidden attribute, 62
- highlighter pen effect, 54–55
- hiragana alphabet, 55–56
- Hiroshi Ichikawa, 212
- `<hr>` element, 59–60
- HSLA color picker, 88–89
- `<html>` tags
 - importance, 4–5
 - optional tags, 3–4
 - primary language declaration, 4–5
- HTML5
 - `<content>` element, 9
 - history, x
 - `<http://html5.validator.nu>` tag, 5
 - philosophies, xiii
 - W3C specification, x
 - WHATWG (Web Hypertext Application Technology Working Group) specification, x–xiv
 - XML and XHTML, xi–xii, xvi, 2–3,
 - “The HTML5 <ruby> element in words of one syllable or less,” 55
- html5 shiv, 54
- html5canvas library, 118
- HTMLElement object, 112
- `<http://html5.validator.nu>` tags, 5
- Hyatt, David, xii

I

`<i>` element, 59
 Ichikawa, Hiroshi, 212
 IDs, names in Google index research, 6
`<iframe>` element, 54, 60
`` element, 54, 94
`importScripts` method, 207, 210
 “Incite a riot,” 58
 inline elements, 54
`<input>` element
 forms, 68
 `onchange` attribute, 81
 `type` attribute
 `autocomplete`, 74, 78
 `autofocus`, 75
 `color`, 74
 `date`, 70–71
 `datetime`, 71
 `email`, 68–69, 82
 `list`, 74–75
 `max`, 74, 78
 `min`, 74, 78
 `month`, 71
 `multiple`, 69, 74, 76
 `number`, 72, 82
 `pattern`, 74, 76–78
 `placeholder`, 75–76
 `range`, 72–73, 80
 `required`, 69, 76
 `search`, 73
 `step`, 74, 78–79
 `tel`, 73, 82
 `text`, 68–69
 `time`, 71
 `url`, 70, 82
 `week`, 72
`<ins>` element, 54
`INSERT` statements, 156–157
`insertId` attribute, 158
 interactive content models, 54
 Internet Archive, 101
 “Introduction to WAI-ARIA,” 51, 184
 italics, `<i>` element, 59
`item` attribute, 63
`itemprop` attribute, 63

J

Japanese language, 55–56
 JavaScript
 `<body>` element requirement, 11
 degrees to radians conversion, 120
 element validation, 85–86

`focus` command, `tabindex` attribute, 63
 form validation, 68
 Modernizr library, 82
 IE application of CSS to HTML5, 11–12
 IE Print Protector, 12
 library, 75
 media API, 102–104
 Modernizr library, 82
 outlines, 31
 `pattern` attribute, 77
 polyfilling, 75
 PPK on JavaScript, 112
 jQuery library, 134
 jQuery Visualize, 139
 JSON library, 148

K

Keith, Jeremy, 58
`key` method, 146–147
`<keygen>` element, 54, 64–65, 68
 Koch, Peter-Paul, 112, 141–142

L

`<label>` element, 54, 68
 Langridge, Stuart, 54
 legacy browsers
 backwards compatibility, 82–83
 `<body>` element requirement, 11
 input type problems, 68–79
 multimedia elements, 100–101
 `<script>` element, JavaScript default, 11
 styling, 12
 videos, 94–98
 legal restrictions, `<small>` element, 18, 24, 60
 Lemon, Gez, 51, 184
 Levithan, Steven, 76
`list` input type, 74–75
 lists
 definition lists, 57
 ordered lists, 56–57
 unordered lists, 16
`load` method, 102–103
`loadeddata` event, 108–109, 128
`loadstart` event, 108
`localStorage` method, 143–144, 146, 149–150, 200
`loop` attribute, 97

M

machine-readable data
 dates and times, 16
 `microdata` attribute, 65

MAMA crawler, Opera, 6
`<mark>` element, 54–55
`<marquee>` element, 60
`max` attribute, 74, 78
`maximumAge` method, 194
media. *See also* `<audio>` element; `<video>` element
accessibility, 110–113
attributes, 102–104
codecs, 98–100
H.264 specification, 98–101
handheld devices, 101–102
legacy browsers, 100–101
software, 101
`<source>` elements, multiple, 99–100
custom controls, 102–110
events, 102–104, 106–108
HTML5 shortcomings, 94
Internet Archive, 101
methods, 102–104
royalty-free, 101
`media` attribute, 102
`<menu>` element, 54, 62, 65
`message` property, 193
Messaging API, 198–200
`<meta charset=utf-8>` tags, 2
`<meta>` tags, XHTML and XML *versus* HTML5, 2–3
metadata content models, 54
`<meter>` element, 65, 68, 80
`microdata` attribute, 65
Microsoft Word 2007 outline view, 30
Mill, Bill, 134
`min` attribute, 74, 78
Miro Video Converter, 101
Modernizr library, 82
`month` input type, 71
`moveTo` method, 123
MS Paint replication, 115–116
multimedia. *See* media
`multiple` attribute, 69, 74, 76

N

Nas, Will, 73
`<nav>` element, 15–18, 33, 54
Newhouse, Mark, 16
Nitot, Tristan, 130–131
`novalidate` attribute, 87
`number` input type, 72, 82
NVDA screen reader, 51

O

`<object>` element, 54, 68
offline
`applicationCache`, 164, 171–172, 174
browser-server process, 168–171
CACHE MANIFEST, 164–167
FALLBACK, 165–167, 172–173
killing caches, 174
NETWORK, 167
serving manifests, 168
Ogg Theora and Vorbis codecs, 98, 101
OggConvert software, 101
`` element, 16, 56–57
`onchange` attribute, 81
`ondragover` event, 177
`ondrop` event, 177–178
`onforminput` event, 80
`oninputchange` event, 88–89
`onload` event, 121
`onmessage` event handler, 199, 202–206, 209–210, 213–215
`open` attribute, 53
`openDatabase` method, 154–155
ordered lists, 56–57
outlines/outlining algorithm
accessibility, 36–37
`<article>` element, 37–42
case study, 42–47
`<hgroup>` element, 33–34
JavaScript implementation, 31
Microsoft Word 2007 outline view, 30
`<section>` element, 31–33, 37–41, 41–42
sectioning content, 31
sectioning roots, 34–35
styling with CSS, 35–36
tool at gsnedders.html5.org/outliner/, 31–32
`<output>` element, 68, 80–81

P–Q

The Paciello Group, ARIA information, 50–51
paragraph-level thematic breaks, `<hr>` element, 59–60
Parker, Todd, et al, 51
path API/paths, 122–124
`pattern` attribute, 74, 76–78
`pause` method, 102–103
Pfeiffer, Silvia, 113
phrasing content models, 54
Pieters, Simon, 12
`placeholder` attribute, 75–76
`play` method, 102–103
`playbackRate` attribute, 109–110

polyfilling, 75
 post method, 68
 poster attribute, 96
 postMessage method, 198–199, 202–210, 213
PPK on JavaScript, 112
 preload attribute, 97, 109
 processing.js library, 134
 <progress> element, 65, 68, 80
 progress event, 108
 pubdate attribute, 27
 public-key cryptography, 65
 putImageData method, 132

R

radians, 120
 range input type, 72–73, 80
 Raphael library, 124
 rectangles, gradients and patterns, 118–120
 regular expressions, 76–77
 removeItem method, 147
 required attribute, 69, 76
 Resig, John, 12, 134
 restore method, 137
 reversed attribute, 57
 RGBA color picker, 88
 role attribute, 63
 role=main tags, WAI-ARIA, 9
 rotate method, 124–126
 Rouget, Paul, 130–131
 rowAffected attribute, 158
 rows attribute, 158
 <rp> element, 55–56
 <rt> element, 55–56
 Ruby, Sam, xiv
 <ruby> element, 55–56

S

save method, 137
 saveTweets method, 156
 scalar measurements, 65
 scale method, 124
 scoped attribute, 65
 screen readers
 HTML5 and ARIA, 51
 problems, 64
 <script> element, 11
 search input type, 73
 Searchhi script, 54
 <section> element, 18, 33, 37–42, 54, 85–86
 sectioning content, 18, 31
 models, 54

sectioning roots, 34–35
 <select> element, 54, 68
 SELECT statements, 158
 sessionStorage method, 143–151
 setAttribute method, 112
 setCustomValidity method, 84–85
 setData method, 179–181
 setDragImage method, 183
 setInterval method, 125, 127, 203
 setItem method, 146–148, 151
 setOnline method, 173
 setTimeout method, 203
 sidebars, 17–18
 Silverlight, 118
 <small> element, 18, 24
 <source> element, 99–100
 spellcheck attribute, 63
 SQLite, 152
 src attribute, 98
 Stachowiak, Maciej, xii
 start attribute, 56
 start method, 107
 step attribute, 74, 78–79
 strokeRect method, 119
 strokeStyle method, 119
 element, 55, 59
 <style scoped> element, 65
 subject attribute, 63
 <summary> element, 52
 SVG (Scalable Vector Graphics) API, x, 54, 124,
 swapCache method, 171–172
 syntax, consistent use, 3

T

tabindex (+"-1") attribute, 63–64, 185
 “Taming Lists,” 16
 <td> element, 34
 tel input type, 73, 82
 testOnline method, 173
 <textarea> element, 54, 68, 85–86
 time
 machine-readable, 26
 UTC (Coordinated Universal Time), 26
 <time> element, 16, 26–27
 time input type, 71
 timeout method, 194
 timestamp object, 191
 timeupdate event, 111, 128
 TinyOgg software, 101
 toDataURL method, 132–133
 transaction method, 161–162
 transform method, 124

`translate` method, 124–126, 137–138

Twitter API, 155–161

2D canvas API, 115, 117, 124

`type` attribute

- `<input>` element, 54
 - `autocomplete`, 74, 78
 - `autofocus`, 75
 - `color`, 74
 - `date`, 70–71
 - `datetime`, 71
 - `email`, 68–69, 82
 - `list`, 74–75
 - `max`, 74, 78
 - `min`, 74, 78
 - `month`, 71
 - `multiple`, 69, 74, 76
 - `number`, 72, 82
 - `pattern`, 74, 76–78
 - `placeholder`, 75–76
 - `range`, 72–73, 80
 - `required`, 69, 76
 - `search`, 73
 - `step`, 74, 78–79
 - `tel`, 73, 82
 - `text`, 68–69
 - `time`, 71
 - `url`, 70, 82
 - `week`, 72
- `<menu>` element, 54

U

`` element, 16

Universal Design for Web Applications, 51

unordered lists, 16

`update` method, 68

`updateSeekable` function, 108

`url` input type, 70, 82

`usemap` attribute, 54

UTC (Coordinated Universal Time), 26

UTF-8 character encoding, 2

V

`valid` attribute, 86

validation

ARIA, 49

avoiding, 86–87

built-in for forms, 68

custom messages, 84–85

elements with JavaScript, 85–86

`<http://html5.validator.nu>` tag, 5

pros and cons, 5

`validity` attribute, 86

van Kesteren, Anne, xiv, 92, 102,

`<video>` element, 54

attributes, 95–98

legacy browsers, 100–101

reasons needed, 92–93

sources, 99–100

“Video for Everybody!”, 100

VLC software, 101

VoiceOver screen reader, 51

W

W3C

Geolocation API, 187

HTML5 specification, xiv

WAI-ARIA (Web Accessibility Initiative’s Accessible Rich Internet Applications) suite, 48–49

attributes

`aria-required`, 76

`aria-valuenow`, 81–82

document landmarks and structure, 49–50

HTML5, combining with, 50

information not built into HTML5, 50

resources, 50–51

`role=main` tags, 9

screen readers, 51

specification, 51

transitional accessibility, 81–82

`watchPosition` method, 189–194

Web Applications 1.0, xi-xii

“A Web Developer’s Responsibility,” 12

Web Sockets API, x, 212–215

Web SQL Databases, 142, 152–162, 208

Web Storage API, x, 142–151

Web Workers API, 198, 200–211

WebKit browsers, 82

`week` input type, 72

WHATWG (Web Hypertext Application Technology Working Group), 111

`width` attribute, 96

`willValid` attribute, 86

Wilson, Chris, xiv

`window` object, 198–199

`ws://` server protocol, 213

X-Z

XHTML *versus* XML and HTML5, 2–3

XML *versus* HTML5 and XHTML, 2–3

`XMLHttpRequest` object, 198, 203, 210, 212