

INDEX

Symbols

- = (assignment operator), 56
- \ (backslash), 33
- \b (backspace), 33
- \ " (double quotation mark), 32
- \e (escape), 33
- \f (form feed), 33
- << (left shift) operator, 81
- \n (newline), 33
- + sign, 42–44
- \r (return), 33
- >> (sign-propagating right shift) operator, 82
- \ ' (single quotation mark), 32
- \t (tab), 32
- \uXXXX escape sequence, 33
- \XXX escape sequence, 33
- \xXX escape sequence, 33
- >>>> (zero-fill right shift) operator, 82

A

- Absolute positioning, 531–36
- Access logs and status codes, 620–21
- Action attribute, 263
- ActiveX controls, 210–11
- <a href> tag, 340
- Alert dialog box, 47–50
- Alert() method, 47–50
- Alphabetic characters, checking for, 321–24
- Alphabetic data, checking for, 451–52
- Alternation, 441–46
- Anchors, 435–41
- anchors object, 345–46

- Animation, 575–78

- Arguments:

- described, 112
- override argument, 692
- passing, 114–15

- Arithmetic operators, 60–61

- Array objects:

- associative arrays, 151, 156
- concat() method, 159
- creating and populating an array simultaneously, 155
- declaring, 152–56
- elements of, 151
- length property, 157
- for loop used to populate, 154
- methods, 158–65
- multidimensional arrays, 151
- new constructor used to declare, 152–53
- overview, 151
- pop() method, 160
- properties of, 157–58
- push() method, 161
- shift() method, 162
- slice() method, 163–64
- splice() method, 164–65
- unshift() method, 162

- Assignment:

- autodecrement (--) operator, 64
- autoincrement (++) operator, 64

- Assignment operator (=), 56

- Associative arrays, 151, 156

- Associativity rules for operators, 56–59

Attributes, 127

- action* attribute, 263
- assigning, 482–85
- bottom* attribute, 533–36
- domain name attribute, 481
- expiration date attribute, 480–81
- get* attribute, 263
- language* attribute, 20
- left* attribute, 533–36
- method* attribute, 263
- name* attribute, 272, 480
- overview, 480
- path* attribute, 481
- post* attribute, 263
- right* attribute, 533–36
- secure attribute, 482
- src* attribute, 20
- top* attribute, 533–36
- type* attribute, 20, 515–18

Autodecrement (--) operator, 64

Autoincrement (++) operator, 64

B

Background graphics, 593

Background properties, 510–11

Backslash (\), 33

Bars:

- graphics and images, 597
- menu bars, collapsing, 241–43
- navigation bars, creating, 238–40
- toolbars, collapsing, 241–43

Backspace (*\b*), 33

Bitwise operators, 80–83

- bitwise shift operators, 81–82
- overview, 80–83

Bitwise shift operators

- left shift (<<) operator, 81
- overview, 81
- sign-propagating right shift (>>) operator, 82
- zero-fill right shift (>>>>) operator, 82

Blocks, 93

Blur() method, 365–67

<BODY> tag, 586

Boolean literals, 34–35

Border properties, 512–14

Bottom attribute, 533–36*Break* statement, 98, 101, 105–7

Browser object model (BOM):

- ActiveX controls, 210–11

frames:

- child frames, 232
- frame* object, 236–37
- HTML frames, creating, 232–36
- menu bars, collapsing, 241–43
- menus, creating, 238–40
- navigation bars, creating, 238–40
- overview, 231–32
- parent window, 232
- toolbars, collapsing, 241–43
- top* property used to keep Main Window out of a frame, 240–41

location object:

- methods, 245
- new page into frame, loading, 245–47
- overview, 244
- properties, 244
- reload()* method, 245
- replace()* method, 245
- URL of document currently loaded in window, accessing, 244–45

MIME types, 211–13

navigator object, 203–13

- browser sniffer, 205
- browser support for, 203
- Internet Explorer support for, 203
- name of browser, 205
- Netscape Navigator support for, 203
- plug-in* object, 207–10
- plug-ins, detecting, 207–10
- properties, 203–4
- version of browser, 205

overview, 201–2

window object:

- history* object, 247–49
- methods, 215–16
- opening windows, 216–20
- open()* method, 216–20
- overview, 214
- properties, 214–15
- resizing a window, 220–22
- screen* object, 250–51
- scrolling in windows, 228–30
- scrollTo()* method, 228–31
- setInterval()* method, 222–27
- setTimeout()* method, 222–26
- timed events, creating, 222–27

Browsers:

- compliance, 547–48

- cookies, used to remove, 492–93
 - CSS-enhanced browsers, 496
 - name of, 205
 - old or disabled browsers:
 - examples of a JavaScript-disabled browser, 10–11
 - examples of a JavaScript-enabled browser, 10–11
 - hiding from old browsers, 9–11
 - hiding within HTML comments, 9–11
 - and `<noscript>` tag, 11–12
 - support for *navigator* object, 203
 - version of, determining, 205
 - Browser sniffer, 205
 - Bugs to watch out for, 44
 - Bullets, 597
 - Button* input type, 288–89
 - Buttons:
 - event handlers, 354
 - form events, 377
 - naming forms and buttons, 272–78
 - radio buttons, 311–14
 - submitting fillout forms with *button* input type, 288–89
- C**
- Carpout* method, 693–94
 - Case keyword, 98
 - Case sensitivity, 17, 590–91
 - Case statement, 97–100
 - Ceil()* method, 175
 - CGI:
 - CGI.pm module
 - advantages of, 662
 - carpout* method, 693–94
 - checkbox()* method, 682–85
 - cookies, 696–701
 - defaults()* method, 692
 - elements, 676–703
 - fatalToBrowser* method, 693–94
 - function-oriented programming style, 663–64
 - HTTP header methods, 702–3
 - labels, 689–90
 - methods, 676–703
 - object-oriented programming style, 662–63
 - override* argument, 692
 - overview, 661
 - popup_menu()* method, 685–88, 690–92
 - processing input from forms, 664–76
 - programming styles, 662–64
 - radio_group()* method, 685–88
 - reset()* method, 692
 - submit()* method, 692
 - textfield()* method, 678–82
 - communication between server and script
 - environment variables, 622–23
 - HTML file with link to CGI script, 624–26
 - overview, 621–22
 - finding CGI applications, 621
 - form processing:
 - creating HTML form, 628–32
 - decoding query string, 639–40
 - decoding query string with Perl, 637–38
 - e-mail, 648–55
 - encoded query string, 636
 - extra path information, 656–58
 - GET* method, 633–35, 640–44
 - input types for forms, 627–28
 - Mail::Mailer* Perl module, 653–54
 - mailto:* protocol, 654–55
 - overview, 626–27
 - parsing form’s input with Perl, 638
 - POST* method, 644–48
 - SMTP server, 648–53
 - SSI (Server Side Includes), 658–61
 - HTML:
 - creating a Web page with, 611–15
 - HTTP headers, 617
 - overview, 611–12, 615–16
 - simple CGI script, 616–19
 - simple HTML document, 612–13
 - tags, 612, 614–15
 - HTTP server, 606–7
 - HTTP status codes and log files, 607–8
 - log files:
 - access logs and status codes, 620–21
 - error logs and STDERR, 619–20
 - overview, 605
 - resources, 581
 - root directory, file URLs and server’s, 611
 - URLs, 609–11
 - CGI.pm module:
 - advantages of, 662
 - carpout* method, 693–94
 - checkbox()* method, 682–85
 - cookies, 696–701
 - defaults()* method, 692
 - elements, 676–703

- CGI.pm module (*cont.*):
 - fatalsToBrowser* method, 693–94
 - function-oriented programming style, 663–64
 - HTTP header methods, 702–3
 - labels, 689–90
 - methods, 676–703
 - object-oriented programming style, 662–63
 - override* argument, 692
 - overview, 661
 - popup_menu()* method, 685–88, 690–92
 - processing input from forms, 664–76
 - programming styles, 662–64
 - radio_group()* method, 685–88
 - reset()* method, 692
 - submit()* method, 692
 - textfield()* method, 678–82
- Character class, 419–22
- Character formatting, 589
- CharAt()* method, 187–88
- Checkboxes, 314–19
- Checkbox()* method, 682–85
- Checkbox object:
 - input devices for forms, programming, 314–19
 - overview, 315
 - properties, 315, 316
- Child frames, 232
- Child nodes, 544
- Class:
 - creating a style, 522–23
 - properties, 405–6
 - and *prototype* object, 145–47
- CodingForums, 581
- Color changed with *className* property, 560–62
- Color properties, 504–5
- Comments, 498
 - adding, 590
 - block comments, 19
 - single-line comments, 19
- Common Gateway Interface. *See* CGI.
- Comparison operators:
 - equality and, 68
 - identity and, 69
 - numbers, comparing, 69–70
 - overview, 67
 - strings, comparing, 70–72
- Complex tag, 591
- Composite data types, 36–37
- Compound statement, 93
- Concatenation and variables, 42–44
- Concatenation operator, 66–67
- Concat()* method, 159
- Conditional constructs:
 - break* statement, 98
 - case* keyword, 98
 - if/else if* statement, 96–97
 - if/else* statement, 94–96
 - labels, 98
 - overview, 93–94
 - switch* statement, 97–100
- Conditional operator, 79–80
- Confirm dialog box, 53–54
- Constants, 191–92
- Constructor method, 401–5
- Constructor used to create objects, 128–29
- Contextual selectors, 527–30
- Continue* statement, 101, 105–7
- Control statements, 105–7
- Control structures, 93
- Conversion:
 - of data types, 83–90
 - of milliseconds, 169–71
 - of numbers, 191–92
- Cookie* object, 482
- Cookies:
 - attributes:
 - assigning, 482–85
 - domain name attribute, 481
 - expiration date attribute, 480–81
 - name attribute, 480
 - overview, 480
 - path attribute, 481
 - secure attribute, 482
 - browser used to remove, 492–93
- CGI.pm module, 696–701
 - components of, 479
 - cookie* object, 482
 - creating, 482–93
 - deleting, 490–93
 - escape()* function, 483–84
 - help with, 493–94
 - HTTP and, 477
 - and Internet Explorer, 478, 492–93
 - and Netscape Navigator, 477–78, 492–93
 - overview, 477–78
 - server, retrieving from, 488–89
 - turning off, 478
 - unescape()* function, 483–84

Core objects:

Array objects:

- associative arrays, 151, 156
- concat()* method, 159
- creating and populating an array simultaneously, 155
- declaring, 152–56
- elements of, 151
- length* property, 157
- for loop used to populate, 154
- methods, 158–65
- multidimensional arrays, 151
- new* constructor used to declare, 152–53
- overview, 151
- pop()* method, 160
- properties of, 157–58
- push()* method, 161
- shift()* method, 162
- slice()* method, 163–64
- splice()* method, 164–65
- unshift()* method, 162

Date object:

- conversion of milliseconds, 169–71
- methods, 167–69
- overview, 165–66
- prototype* property, 171–72

Math object:

- ceil()* method, 175
- floor()* method, 175, 176
- methods, 173
- overview, 172
- Pi, 174
- power of a number, 174
- properties, 172–73
- random()* method, 176
- random number generation, 176–77
- rounding up and down, 174–76
- round()* method, 175
- square root, 174

Number object:

- constants, 191–92
- conversion of numbers, 191–92
- formatting numbers, 192–93
- methods, 191
- overview, 190–91
- properties, 190
- toFixed()* method, 192–93
- overview, 151

String object:

- charAt()* method, 187–88
- extraction of substrings from a string, methods for, 187–88
- HTML methods, 182–83
- indexOf()* method, 185–87
- lastIndexOf()* method, 185–87
- length* property, 179–80
- methods, 182–89
- overview, 177–78
- position in a string, finding, 184–87
- properties, 179–81
- prototype* property, 179, 181
- replace()* method, 188–89
- search and replace methods, 188–89
- search()* method, 188–89
- splice()* method, 187
- split()* method, 187–88
- string manipulation methods, 184
- substring()* method, 187
- substr()* method, 187–88
- with* keyword, 197

Wrapper objects:

- Boolean* object, 193–94
- Function* object, 195–96
- Number* object, 190–93
- overview, 177
- String* object, 177–89

Credit card validation:

- expiration date of card, checking, 464–66
- length of characters, checking, 464–70
- Luhn formula and, 470–71
- overview, 464
- prefix value, checking, 464–70
- valid type, checking for, 464–70

CSS:

- background properties, 510–11
- border properties, 512–14
- color properties, 504–5
- comments, 498
- CSS-enhanced browsers, 496
- declaration block, 496–97
- font properties, 506–7
- grouping, 498–500
- how it works, 496–501
- image properties, 510–11
- margin properties, 512–14
- measurement, units of, 500–501
- overview, 496

CSS (*cont.*):

- properties, 501–14
- selectors, 496–97
- style object:
 - color changed with *className* property, 560–62
 - overview, 554–58
 - properties, 555–56
 - text positioned with *style* property, 558–60
- text alignment properties, 508–9

D

Data types:

- composite data types, 36–37
- conversion of, 83–90
- overview, 31
- primitive data types:
 - Boolean literals, 34–35
 - escape sequences, 32–34
 - null, 36
 - numeric literals, 31–32
 - overview, 31
 - quotes, 32
 - string literals, 32–34
 - typeof* operator, 35–36
 - undefined, 36
- variables converted to, 40–41

Date object:

- conversion of milliseconds, 169–71
- methods, 167–69
- overview, 165–66
- prototype* property, 171–72

Debugging:

- errors, 24
- in Explorer, 28–29
- functions, 124–25
- load-time errors, 24
- logical errors, 24
- in Netscape, 25–27
- overview, 24
- runtime errors, 24
- tools:
 - for debugging in Explorer, 28–29
 - javascript*: URL protocol, 24–25
 - JavaScript Console (Netscape), 25–27
- types of errors, 24

Declaration block, 496–97

Declaring:

- array objects, 152–56

functions, 112–13

variables, 38–39

Decoding query string, 639–40

Decoding query string with Perl, 637–38

Default colors, 593–96

Defaults() method, 692

Defining functions, 112–13

Delimiters, 412–14

Dialog boxes:

- alert dialog box, 47–50
- confirm dialog box, 53–54
- described, 47
- prompt dialog box, 50–52

`<div>` tag, 536–37

Document object:

- forms. *See* forms.
- HTML `<body>` tag attributes, 256
- JavaScript, using document's properties in, 259–60
- methods, 260–63
- overview, 255–56
- properties, 256–60
- write()* method, 259–60

Document object model (DOM):

- animation, 575–78
- browser compliance, 547–48
- document* object:
 - forms. *See* forms.
 - HTML `<body>` tag attributes, 256
 - JavaScript, using document's properties in, 259–60
 - methods, 260–63
 - overview, 255–56
 - properties, 256–60
 - write()* method, 259–60

DOM Level 1, 253–54

dot syntax, 255

event handling and, 563–66

getElementById() method, 548–50

getElementsByTagName() method, 550–51

hierarchy of, 254–55

nodes, 254

- child nodes, 544

- element properties, 547

- methods, 542–43

- nodeName* property, 544–46

- nodeType* property, 544–46

- overview, 541–43

- parent nodes, 544

- properties, 542–43
 - scrolling with, 551–54
 - siblings, 543
 - overview, 253–54, 540
 - visibility, setting, 569–75
 - visibility* property
 - drop-down menus, 569–72
 - overview, 569
 - tool tips, 573–74
 - W3C and, 253
 - z-index and dynamic positioning, 567–68
- Domain name attribute, 481
- DOM Level 1, 253–54
- Dot metacharacter, 417–18
- Dot (.) syntax, 128, 255
- Double quotation mark ("), 32
- Do/while* loop, 102–3
- Drop-down menus, 304–11, 569–72
- Dynamic HTML (DHTML):
- CSS:
 - background properties, 510–11
 - border properties, 512–14
 - color properties, 504–5
 - comments, 498
 - CSS-enhanced browsers, 496
 - declaration block, 496–97
 - font properties, 506–7
 - grouping, 498–500
 - how it works, 496–501
 - image properties, 510–11
 - margin properties, 512–14
 - measurement, units of, 500–501
 - overview, 496
 - properties, 501–14
 - selectors, 496–97
 - style* object, 554–62
 - text alignment properties, 508–9
 - DOM:
 - animation, 575–78
 - browser compliance, 547–48
 - event handling and, 563–66
 - getElementById()* method, 548–50
 - getElementsByTagName()* method, 550–51
 - nodes and, 541–47
 - overview, 540
 - visibility, setting, 569–75
 - z-index and dynamic positioning, 567–68
 - embedded style sheet:
 - overview, 515
 - type* attribute, 515–18
 - overview, 495
 - style sheets:
 - class, creating a style, 522–23
 - contextual selectors, 527–30
 - CSS. *See* Dynamic HTML, CSS.
 - embedded style sheet, 515–18
 - external style sheets, 520–21
 - ID selector, 524–25
 - inline style sheet, 518–19
 - overview, 495
 - positioning elements and layers, 530–40
 - * tag, 525–27
 - types of, 515–40
- E**
- ECMA, 7
- ECMAScript, 7
- escape (*&e*), 33
- Eich, Brendan, 1
- Element properties, 547
- Elements:
- of array objects, 151
 - CGI.pm* module, 676–703
 - positioning elements and layers
 - absolute positioning, 531–36
 - bottom* attribute, 533–36
 - <div>* tag, 536–37
 - left* attribute, 533–36
 - overview, 530–31
 - relative positioning, 537–39
 - right* attribute, 533–36
 - style sheets, 530–40
 - top* attribute, 533–36
 - z-index, 539–40
- Elements[]* array:
- forms* object, 269–70, 272
 - properties, 272
- E-mail:
- addresses, checking, 324–26, 461–64
 - form processing, 648–55
- Embedded style sheet:
- overview, 515
 - type* attribute, 515–18
- Embeds* object, 346–48
- Empty fields, checking for, 319–21, 446–48
- Encoded query string, 636
- Environment variables, 622–23
- Equality and comparison operators, 68

Errors:

- events, 393–94
- load-time, 24
- logical, 24
- logs and STDERR, 619–20
- runtime, 24

Escape() function, 483–84

Escape sequences, 32–34

- backslash (\), 33
- backspace (\b), 33
- double quotation mark (\"), 32
- escape (\e), 33
- form feed (\f), 33
- newline (\n), 33
- return (\r), 33
- single quotation mark (\'), 32
- tab (\t), 32
- \uXXXX, 33
- \XXX, 33
- \xXX, 33

European Computer Manufacturers Association, 7

Eval() method, 89–90Event handlers. *See also* Events.

- for buttons, 354
- creating, 354–56
- for documents, 354
- error events, 393–94
- event methods, 356–58
- event object, 394–96
- form events:
 - buttons, 377
 - event methods, 380–83
 - list of, 376
 - onBlur* event, 383–84
 - onChange* event handler, 385–86
 - onClick* event, 379–80
 - onFocus* event, 383–84
 - onSubmit* event, 387–90
 - overview, 376
 - this* keyword, 377–78

forms:

- onClick* event, 282–83, 288–89
- onReset* event, 285–86
- onSubmit* event, 283–85
- this* keyword, 287–88

frame events:

- blur()* method, 365–67
- focus()* method, 365–67
- list of, 361

- onBlur* event handler, 363–65
- onFocus* event handler, 363–65
- onLoad* event handler, 361–63
- onload()* method, 363
- onUnload* event handler, 361–63
- onunload()* method, 363
- overview, 360–61

as JavaScript methods, 356–60

key events:

- onKeyDown* event, 390–92
- onKeyPress* event, 390–92
- onKeyUp* event, 390–92
- overview, 390–92

link events, 375–76

for links, 354

list of, 353–54

mouse events:

- and images, 370–74
- list of, 368
- onMouseOut* event handler, 370–71
- onMouseOver* event handler, 370–71
- overview, 367–68
- rollovers, 370–71
- slide show creation, 371–74
- using, 368–70

overview, 351–54

return values, 358–60

submitting rollout forms, 281–88

window events:

- blur()* method, 365–67
- focus()* method, 365–67
- list of, 361
- onBlur* event handler, 363–65
- onFocus* event handler, 363–65
- onLoad* event handler, 361–63
- onload()* method, 363
- onUnload* event handler, 361–63
- onunload()* method, 363
- overview, 360–61

Event methods:

- event handlers, 356–58
- form events, 380–83

Event object, 394–96

Events. *See also* event handlers.example of *onClick* event, 4–5

form events:

- buttons, 377
- event methods, 380–83
- list of, 376

- onBlur* event, 383–84
 - onChange* event handler, 385–86
 - onClick* event, 379–80
 - onFocus* event, 383–84
 - onSubmit* event, 387–90
 - overview, 376
 - this* keyword, 377–78
 - frame events:
 - blur()* method, 365–67
 - focus()* method, 365–67
 - list of, 361
 - onBlur* event handler, 363–65
 - onFocus* event handler, 363–65
 - onLoad* event handler, 361–63
 - onload()* method, 363
 - onUnLoad* event handler, 361–63
 - onunload()* method, 363
 - overview, 360–61
 - function called from, 117–18
 - individually listed and defined, 5
 - key events:
 - onKeyDown* event, 390–92
 - onKeyPress* event, 390–92
 - onKeyUp* event, 390–92
 - overview, 390–92
 - link events, 375–76
 - mouse events:
 - and images, 370–74
 - list of, 368
 - onMouseOut* event handler, 370–71
 - onMouseOver* event handler, 370–71
 - overview, 367–68
 - rollovers, 370–71
 - slide show creation, 371–74
 - using, 368–70
 - onAbort* event, 5
 - onBlur* event, 5, 383–84
 - onChange* event, 5
 - onChange* event handler, 385–86
 - onClick* event, 5, 379–80
 - onError* event, 5
 - onFocus* event, 5, 383–84
 - onKeyDown* event, 390–92
 - onKeyPress* event, 390–92
 - onKeyUp* event, 390–92
 - onLoad* event, 5
 - onMouseOut* event, 5
 - onMouseOver* event, 5
 - onSubmit* event, 5, 387–90
 - onUnLoad* event, 5
 - window events:
 - blur()* method, 365–67
 - focus()* method, 365–67
 - list of, 361
 - onBlur* event handler, 363–65
 - onFocus* event handler, 363–65
 - onLoad* event handler, 361–63
 - onload()* method, 363
 - onUnLoad* event handler, 361–63
 - onunload()* method, 363
 - overview, 360–61
 - Exec()* method, 401–2, 404–5
 - Expiration date attribute, 480–81
 - Expiration date of credit card, checking, 464–66
 - Expressions, 55
 - External files, placing JavaScript within, 13
 - External images, 597
 - External style sheets, 520–21
 - Extraction of substrings from a string, methods for, 187–88
 - Extraneous characters in forms, removing, 453–56
 - Extra path information, 656–58
- ## F
- FatalToBrowser* method, 693–94
 - Flash 5, 597
 - Floating-point numbers, 88–89
 - Floor()* method, 175, 176
 - Focus()* method, 365–67
 - Font properties, 506–7
 - For/in* loop, 105
 - For* loop, 104–5, 154
 - Formatting numbers, 192–93
 - Form events:
 - buttons, 377
 - event methods, 380–83
 - list of, 376
 - onBlur* event, 383–84
 - onChange* event handler, 385–86
 - onClick* event, 379–80
 - onFocus* event, 383–84
 - onSubmit* event, 387–90
 - overview, 376
 - this* keyword, 377–78
 - Form feed (f), 33
 - Form processing:
 - creating HTML form, 628–32
 - decoding query string, 639–40

Form processing (*cont.*):

- decoding query string with Perl, 637–38
- e-mail, 648–55
- encoded query string, 636
- extra path information, 656–58
- GET method, 633–35, 640–44
- input types for forms, 627–28
- Mail::Mailer Perl module, 653–54
- mailto: protocol, 654–55
- overview, 626–27
- parsing form's input with Perl, 638
- POST method, 644–48
- SMTP server, 648–53
- SSI (Server Side Includes), 658–61

Forms:

- action attribute, 263
- event handlers:
 - onClick event, 282–83, 288–89
 - onReset event, 285–86
 - onSubmit event, 283–85
 - this keyword, 287–88
- forms object:
 - elements[] array, 269–70, 272
 - forms[] array, 268–69
 - <form> tag elements and properties, 274
 - methods, 272
 - name attribute, 272
 - naming forms and buttons, 272–78
 - overview, 268
 - properties, 271
- <form> tag elements and properties, 274
- get attribute, 263
- HTML and, 263–64
- input devices, programming:
 - checkboxes, 314–19
 - checkbox object, 314–19
 - drop-down menus, 304–11
 - overview, 294
 - password object, 298–301
 - radio buttons, 311–14
 - radio object, 311–14
 - select object, 304–11
 - textarea object, 301–4
 - text object, 294–98
- input types, 264–67
- method attribute, 263
- naming forms and buttons, 272–78
- overview, 263
- post attribute, 263

processing. *See* form processing.

referencing form elements by name, 275–78

simple form validation:

- alphabetic characters, checking for, 321–24
- e-mail addresses, checking, 324–26
- empty fields, checking for, 319–21
- overview, 319
- password fields, checking, 326–28

steps used to produce, 264

submitting fillout forms:

- with button input type, 288–89
- with event handlers, 281–88
- image, submitting form with, 280–81
- with JavaScript, 281–88
- overview, 279–80

pop-up window, displaying form's content in, 292–93

reset() method, 290–91

submit() method, 290–91

validation of. *See* form validation.

Forms[] array, 268–69

Forms object:

- elements[] array, 269–70, 272
- forms[] array, 268–69
- methods, 272
- name attribute, 272
- overview, 268
- properties, 271

<form> tag, 274

Form validation:

- alphabetic data, checking for, 451–52
- combining functions that validate each field of form, 471–75
- credit cards, checking for valid, 464–71
- e-mail addresses, checking for valid, 461–64
- empty fields, checking for, 446–48
- extraneous characters, removing, 453–56
- non-alphanumeric characters, removing, 456
- non-digits, removing, 455
- null fields, checking for, 446–48
- overview, 446
- parentheses, removing unwanted, 454
- phone numbers, checking for valid, 458–61
- Social Security numbers, checking for valid, 457–58
- spaces and dashes, removing, 453
- zip codes, checking for numeric, 448–50

Frame events:

- blur() method, 365–67

focus() method, 365–67
 list of, 361
onBlur event handler, 363–65
onFocus event handler, 363–65
onLoad event handler, 361–63
onload() method, 363
onUnload event handler, 361–63
onunload() method, 363
 overview, 360–61
 Frame object, 236–37
 Frames, 602
 child frames, 232
 frame object, 236–37
 HTML frames, creating, 232–36
 menu bars, collapsing, 241–43
 menus, creating, 238–40
 navigation bars, creating, 238–40
 overview, 231–32
 parent window, 232
 toolbars, collapsing, 241–43
 top property used to keep main window out of a
 frame, 240–41
 FreewareJava, 581
 Function-oriented programming style, 663–64
 Functions:
 arguments, 112
 calling, 112–13
 debugging, 124–25
 declaring, 112–13
 defining, 112–13
 event, calling a function from a, 117–18
 <*head*> tag, placement within, 112
 invoking, 112–13
 JavaScript, calling a function from, 119
 link, calling a function from a, 116–17
 overview, 111
 parameters, 114
 passing arguments, 114–15
 recursive functions, 123–24
 return statement, 120–23
 return values, 120–23
 <*script*> tag, placement within, 112
 typeof operator used to check, 112
 variables in, scope of, 119–20
G
Get attribute, 263
GetElementById() method, 548–50
GetElementByTagName() method, 550–51

GET method, 633–35, 640–44
 Global variables, 42
 Graphics. *See* Images and graphics.
 Grouping:
 CSS, 498–500
 metacharacters, 442–43

H

<*head*> tag, 112, 585, 586
 Headings, 586–87
Hello.html file, 15
 Help with cookies, 493–94
 Hierarchy of DOM, 254–55
History object, 247–49
 HTML:
 basics of, 7–8
 <*body*> tag, 7, 256, 586
 and case sensitivity, 17
 creating a Web page with, 611–15
 file with link to CGI script, 624–26
 and forms, 263–64
 <*form*> tag, 4
 frames, 232–36
 <*head*> tag, 7, 585–86
 hello.html file, 15
 HTTP headers, 617
 images, 329–31
 methods, 182–83
 overview, 611–12, 615–16
 <*script*> tag, 7, 9, 15, 19–20
 simple CGI script, 616–19
 simple HTML document, 612–13
 tags, 612, 614–15
 using the *.html* or *.htm* extensions, 15
 HTML documents:
 case sensitivity, 590–91
 character formatting, 589
 comments, adding, 590
 frames, 602
 graphics and images:
 background graphics, 593
 bars, 597
 bullets, 597
 complex tag, 591
 default colors, 593–96
 external images, 597
 finding, 592
 Flash 5, 597
 icons, 597

HTML documents, graphics and images (*cont.*):

- inline image, creating, 591
- sizing and placing, 591–92
- text alternative for, creating, 592
- JavaScript, 601–2
- linking:
 - creating a link, 589–90
 - overview, 589
 - to URLs, 590
- logical tags, 589
- metatags, 599–600
- minimal HTML document:
 - headings, 586–87
 - lists, 587
 - overview, 586
 - tables, 588
 - tags to separate text, 587
- overview, 583–84
- physical tags, 589
- source code, 602
- style sheets, 600–601
- tags:
 - `<body>` tag, 7, 256, 586
 - `<head>` tag, 7, 585, 586
 - `<HTML>` tag, 585
 - order of required elements and their tags, 585
 - overview, 584
 - required elements, 585
 - `<title>` tag, 585
- tips for, 602–3
- troubleshooting, 597–99
- `<HTML>` tag, 585
- `.html` or `.htm` extensions, 15
- HTTP:
 - cookies, 477
 - header methods, 702–3
 - headers, 617
 - server, 606–7
 - status codes and log files, 607–8

I

- Icons, 597
- Identity and comparison operators, 69
- ID selector, 524–25
- If/else if* statement, 96–97
- If/else* statements
 - conditional operator and, 79–80
 - described, 94–96
- Image()* constructor, 336

Image object:

- Image()* constructor, 336
- onClick* event used to randomly display images, 338–40
- overview, 331–32
- preloading images, 336
- properties, 332
- randomly displaying images, 338–40
- rollover with mouse event, 336–38
- src* property, replacing images dynamically with, 333–35
- Images and graphics:
 - background graphics, 593
 - bars, 597
 - bullets, 597
 - complex tag, 591
 - default colors, 593–96
 - external images, 597
 - finding, 592
 - Flash 5, 597
 - HTML and, 329–31
 - icons, 597
 - image* object:
 - Image()* constructor, 336
 - onClick* event used to randomly display images, 338–40
 - overview, 331–32
 - preloading images, 336
 - properties, 332
 - randomly displaying images, 338–40
 - rollover with mouse event, 336–38
 - src* property, replacing images dynamically with, 333–35
 - inline image, creating, 591
 - JavaScript and, 331–40
 - mouse events, 370–74
 - overview, 328–29
 - properties, 510–11
 - sizing and placing, 591–92
 - submitting form with, 280–81
 - text alternative for, creating, 592
- IndexOf()* method, 185–87
- Inheritance, 146–47
- Initializing variables, 38–39
- Inline image, 591
- Inline style sheet, 518–19
- Input devices for forms, programming
 - checkboxes, 314–19
 - checkbox* object, 314–19

- drop-down menus, 304–11
 - overview, 294
- password* object, 298–301
- radio buttons, 311–14
- radio* object, 311–14
- select* object, 304–11
- textarea* object, 301–4
- text* object, 294–98
- Input types for forms, 264–67, 627–28
- Instance properties, 405–6
- Internet Explorer:
 - cookies and, 478, 492–93
 - navigator* object, support for, 203
 - versions supported by JavaScript and JScript, 6
- Invalid names for variables, 38

J

- JavaScript, 601–2
 - attributes. *See* Attributes.
 - debugging, 24–29
 - testing with Explorer’s debugging window, 28–29
 - testing with Netscape’s JavaScript Console, 25–27
 - testing with the *javascript:* URL protocol, 24–25
 - defined, 1–2
 - and events. *See* Events.
 - example of Web page using, 2
 - external files, placing within, 13
 - generating HTML output, 21–23
 - HTML compared, 3
 - and HTML documents:
 - script execution, 16
 - where to embed within, 7–8, 15, 19
 - and images, 331–40
 - Java compared, 2
 - JScript compared, 6
 - methods. *See* Methods.
 - objects, support for, 127–28
 - and old or disabled browsers:
 - examples of a JavaScript-disabled browser, 10–11
 - examples of a JavaScript-enabled browser, 10–11
 - hiding from old browsers, 9–11
 - hiding within HTML comments, 9–11
 - and *<noscript>* tag, 11–12
 - overview, 1–3

- placement of, 7–8
- script setup, 15–30
- standardizing using ECMAScript, 7
- strings, 21
- submitting fillout forms, 281–88
- syntactical details, 17–20
 - case sensitivity, 17
 - comments, 19
 - free form and reserved words, 17–18
 - statements and semicolons, 18
 - whitespace, 17
- URLs, 340
- uses for, 3
- versions of, 6–7, 20

- Javascript:* URL protocol, 24–25
- JavaScript Central on Netscape’s devedge site, 581
- JavaScript Console (Netscape), 25–27
- JavaScript Planet, 581
- JScript, 6

K

- Key events:
 - onKeyDown* event, 390–92
 - onKeyPress* event, 390–92
 - onKeyUp* event, 390–92
 - overview, 390–92

L

- Labels, 98, 108–9, 689–90
- LastIndexOf()* method, 185–87
- Layers:
 - positioning elements and layers:
 - absolute positioning, 531–36
 - bottom* attribute, 533–36
 - <div>* tag, 536–37
 - left* attribute, 533–36
 - overview, 530–31
 - relative positioning, 537–39
 - right* attribute, 533–36
 - style sheets, 530–40
 - top* attribute, 533–36
 - z-index*, 539–40
 - Left* attribute, 533–36
- Left shift (<<) operator, 81
- Length of characters, credit card validation and checking, 464–70
- Length* property:
 - array objects, 157
 - String* object, 179–80

Link events, 375–76

Links:

`<a href>` tag, 340

`anchors` object, 345–46

calling a function from, 116–17

creating a link, 589–90

`embeds` object, 346–48

JavaScript URLs, 340

`links` object, 341–45

overview, 340, 589

to URLs, 590

`Links` object, 341–45

Literal notation, 400–401

LiveScript. *See* JavaScript.

Load-time errors, 24

Local variables, 42

`Location` object:

methods, 245

new page into frame, loading, 245–47

overview, 244

properties, 244

`reload()` method, 245

`replace()` method, 245

URL of document currently loaded in window,
accessing, 244–45

Log files:

access logs and status codes, 620–21

error logs and STDERR, 619–20

Logical AND `&&` operator, 73–74

Logical errors, 24

Logical NOT `!` operator, 76–77

Logical operators:

logical AND `&&` operator, 73–74

logical NOT `!` operator, 76–77

logical OR `//` operator, 75–76

overview, 72–73

Logical OR `//` operator, 75–76

Logical tags, 589

Loops:

`break` statement, 101, 105–7

`continue` statement, 101, 105–7

control statements, 105–7

`do/while` loop, 102–3

`for` loop, 104–5

`for/in` loop, 105

labels, 108–9

nested loops, 107–8

overview, 101

`while` loop, 101–2

Loosely typed language, JavaScript as, 40–41

Luhn formula and credit card validation, 470–71

M

`Mail::Mailer` Perl module, 653–54

`Mailto:` protocol, 654–55

Margin properties, 512–14

`Match()` method, 408–9

`Math` object:

`ceil()` method, 175

`floor()` method, 175, 176

methods, 173

overview, 172

Pi, 174

power of a number, 174

properties, 172–73

`random()` method, 176

random number generation, 176–77

rounding up and down, 174–76

`round()` method, 175

square root, 174

Measurement, units of, 500–501

Menu bars, collapsing, 241–43

Menus, creating, 238–40

Metacharacters:

alternation, 441–46

anchors, 435–41

capturing, 443–46

character class, 419–22

clustering, 442–43

dot metacharacter, 417–18

grouping, 442–43

list of, 415–17

metasymbols, 422–26

overview, 414–17

pattern matches, metacharacters to repeat,
426–35

quantifiers, 426

remembering, 443–46

zero-width assertions, 435–41

Metasymbols, 422–26

Metatags, 599–600

`Method` attribute, 263

Methods, 130–31, 542–43

array objects, 158–65

`blur()` method, 365–67

`carpout` method, 693–94

`ceil()` method, 175

`CGI.pm` module, 676–703

charAt() method, 187–88
checkbox() method, 682–85
concat() method, 159
Date object, 167–69
defaults() method, 692
 defined for an object, 134–39
document object, 260–63
 extraction of substrings from a string, methods
 for, 187–88
fatalToBrowser method, 693–94
floor() method, 175, 176
focus() method, 365–67
forms object, 272
getElementById() method, 548–50
getElementsByTagName() method, 550–51
 HTML methods, 182–83
 HTTP header methods, 702–3
indexOf() method, 185–87
lastIndexOf() method, 185–87
location object, 245
Math object, 173
Number object, 191
onload() method, 363
onunload() method, 363
open() method, 216–20
password object, 299
pop() method, 160
popup_menu() method, 685–88, 690–92
push() method, 161
radio_group() method, 685–88
random() method, 176
reload() method, 245
replace() method, 188–89, 245
reset() method, 692
round() method, 175
scrollTo() method, 228–31
 search and replace methods, 188–89
search() method, 188–89
setInterval() method, 222–27
setTimeout() method, 222–26
shift() method, 162
slice() method, 163–64
splice() method, 164–65, 187
split() method, 187–88
 string manipulation methods, 184
String object, 182–89
submit() method, 692
substring() method, 187
substr() method, 187–88

textarea object, 302
textfield() method, 678–82
text object, 295
toFixed() method, 192–93
unshift() method, 162
window object, 215–16
write(), 21–23
writeln(), 21–23
 Microsoft Script Debugger (MSSD), 29
 MIME types, 211–13
 Minimal HTML document:
 headings, 586–87
 lists, 587
 overview, 586
 tables, 588
 tags to separate text, 587
 Mouse events:
 and images, 370–74
 list of, 368
 onMouseOut event handler, 370–71
 onMouseOver event handler, 370–71
 overview, 367–68
 rollovers, 370–71
 slide show creation, 371–74
 using, 368–70
 Multidimensional arrays, 151
 Multiple selects, 309–11
 Murray, Joan, 603

N

Name attribute, 272, 480
 Name of browser, 205
 Naming forms and buttons, 272–78
 Navigation bars, creating, 238–40
Navigator object, 203–13
 browser sniffer, 205
 browser support for, 203
 Internet Explorer support for, 203
 name of browser, 205
 Netscape Navigator support for, 203
 plug-in object, 207–10
 plug-ins, detecting, 207–10
 properties, 203–4
 version of browser, 205
 Nested loops, 107–8
 Netscape Navigator:
 cookies and, 477–78, 492–93
 navigator object, support for, 203
 and *<noscript>* tags, 11–12

- Netscape Navigator (*cont.*):
 - plug-ins, detecting, 207–10
 - versions supported by JavaScript and JScript, 6
- New* constructor used to declare array objects, 152–53
- New* operator, 131
- New page into frame, loading, 245–47
- Newline (*\n*), 33
- nodeName* property, 544–46
- Nodes, 254
 - DOM:
 - child nodes, 544
 - element properties, 547
 - methods, 542–43
 - nodeName* property, 544–46
 - nodeType* property, 544–46
 - overview, 541–43
 - parent nodes, 544
 - properties, 542–43
 - scrolling with, 551–54
 - siblings, 543
 - NodeType* property, 544–46
- Non-alphanumeric characters, removing, 456
- Non-digits, removing, 455
- Null, 36
- Null fields, checking for, 446–48
- Null string, 32
- Number* object:
 - constants, 191–92
 - conversion of numbers, 191–92
 - formatting numbers, 192–93
 - methods, 191
 - overview, 190–91
 - properties, 190
 - toFixed()* method, 192–93
- Numbers, comparing, 69–70
- Numeric literals, 31–32

- O**
- Object()* constructor, 131–32
- Object literals, 140–41
- Object models, 128
- Object-oriented programming style, 662–63
- Objects:
 - attributes, 127
 - constructor used to create, 128–29
 - dot (.) syntax, 128
 - for/in* loop, 143–45
 - JavaScript support for, 127–28
 - manipulation of:
 - for/in* loop, 143–45
 - prototypes used to extend objects, 145–48
 - with* keyword, 141–43
 - methods defined for, 130–31, 134–39
 - methods defined in constructor, 136–38
 - object models, 128
 - overview, 127–28
 - properties, 127, 129–30, 138–39
 - prototypes used to extend objects, 145–48
 - user-defined objects:
 - creating, 133
 - methods defined for an object, 134–39
 - new* operator, 131
 - Object()* constructor, 131–32
 - object literals, 140–41
 - user-defined function used to create, 133
 - with* keyword, 141–43
- OnBlur* event, 383–84
- OnBlur* event handler:
 - frame events, 363–65
 - window events, 363–65
- OnChange* event handler, 385–86
- OnClick* event:
 - form events, 379–80
 - forms, 282–83, 288–89
 - used to randomly display images, 338–40
- OnFocus* event, 383–84
- OnFocus* event handler, 363–65
- OnKeyDown* event, 390–92
- OnKeyPress* event, 390–92
- OnKeyUp* event, 390–92
- OnLoad* event handler, 361–63
- Onload()* method, 363
- OnMouseOut* event handler, 370–71
- OnMouseOver* event handler, 370–71
- OnReset* event, 285–86
- OnSubmit* event, 283–85, 387–90
- OnUnload* event handler, 361–63
- Onunload()* method, 363
- Opening windows, 216–20
- Open()* method, 216–20
- Operands, 55
- Operators:
 - arithmetic operators, 60–61
 - assignment operator (=), 56
 - associativity rules for, 56–59
 - autodecrement (--) operator, 63–65
 - autoincrement (++) operator, 63–65

- bitwise operators, 80–83
- comparison operators, 67–72
- concatenation operator, 66–67
- conditional operator, 79–80
- logical operators, 72–78
- overview, 55
- precedence rules for, 56–59
- shortcut assignment operators, 62–63
- special operators, 91
- Override* argument, 692

P

- Parameters, 114
- Parentheses, removing unwanted, 454
- Parent nodes, 544
- Parent window, 232
- parseFloat()* method, 88–89
- parseInt()* method, 85–87
- Parsing form's input with Perl, 638
- Passing arguments, 114–15
- Password fields, checking, 326–28
- Password* object, 298–301
- Path attribute, 481
- Pattern matches, metacharacters to repeat, 426–35
- Phone numbers, checking for valid, 458–61
- Physical tags, 589
- Pi, 174
- Placement of JavaScript, 7–8
- Plug-in* object, 207–10
- Plug-ins, detecting, 207–10
- plus sign (+):
 - concatenation of variable and strings with, 42–44
 - as numeric operator, 43
 - as string operator, 42–43
- Pop()* method, 160
- Popup_menu()* method, 685–88, 690–92
- Pop-up window, displaying form's content in, 292–93
- Position in a string, finding, 184–87
- Positioning elements and layers:
 - style sheets:
 - absolute positioning, 531–36
 - bottom* attribute, 533–36
 - `<div>` tag, 536–37
 - left* attribute, 533–36
 - overview, 530–31
 - relative positioning, 537–39
 - right* attribute, 533–36
 - top* attribute, 533–36

- z-index, 539–40
- Post* attribute, 263
- POST* method, 644–48
- Power of a number, 174
- Precedence rules, 56–59
- Prefix value, checking, 464–70
- Preloading images, 336
- Primitive data types:
 - Boolean literals, 34–35
 - conversion methods:
 - eval()* method, 89–90
 - parseFloat()* method, 88–89
 - parseInt()* method, 85–87
 - escape sequences, 32–34
 - null, 36
 - numeric literals, 31–32
 - overview, 31
 - quotes, 32
 - string literals, 32–34
 - typeof* operator, 35–36
 - undefined, 36
- Processing input from forms, 664–76
- Programming styles, 662–64
- Prompt dialog box, 50–52
- Prompt()* method, 50–52
- Properties, 127, 129–30, 501–14, 542–43, 555–56
 - array objects, 157–58
 - color changed with *className* property, 560–62
 - document* object, 256–60
 - forms* object, 271
 - image* object, 332
 - length* property, 157, 179–80
 - location* object, 244
 - Math* object, 172–73
 - navigator* object, 203–4
 - nodeName* property, 544–46
 - nodeType* property, 544–46
 - Number* object, 190
 - password* object, 299
 - prototype* property, 171–72, 179, 181
 - RegExp()* object, 405–8
 - src* property, replacing images dynamically with, 333–35
 - String* object, 179–81
 - textarea* object, 302
 - text* object, 295
 - text positioned with *style* property, 558–60
 - top* property used to keep main window out of a frame, 240–41

Properties (*cont.*):

- visibility* property
 - drop-down menus, 569–72
 - overview, 569
 - tool tips, 573–74
- window* object, 214–15

Prototype object:

- class and, 145–47
- extending, 147–48
- inheritance and, 146–47
- overview, 145

Prototype property:

- Date* object, 171–72
- String* object, 179, 181

Push() method, 161**Q**

Quantifiers, 426

Quotes, 32

R

Radio buttons, 311–14

Radio_group() method, 685–88*Radio* object:

- input devices for forms, programming, 311–14
- methods, 312–13
- overview, 311–12
- properties, 312

Randomly displaying images, 338–40

Random() method, 176

Random number generation, 176–77

Recursive functions, 123–24

Referencing form elements by name, 275–78

RegExp() object:

- class properties, 405–6
- exec()* method, 401–2, 404–5
- instance properties, 405–6
- overview, 401–2
- properties, 405–8
- test()* method, 401–4

Regular expressions:

- constructor method, 401–5
- creating, 400–414
- form validation:
 - alphabetic data, checking for, 451–52
 - combining functions that validate each field of form, 471–75
 - credit cards, checking for valid, 464–71
 - e-mail addresses, checking for valid, 461–64

- empty fields, checking for, 446–48
- extraneous characters, removing, 453–56
- non-alphanumeric characters, removing, 456
- non-digits, removing, 455
- null fields, checking for, 446–48
- overview, 446
- parentheses, removing unwanted, 454
- phone numbers, checking for valid, 458–61
- Social Security numbers, checking for valid, 457–58
- spaces and dashes, removing, 453
- zip codes, checking for numeric, 448–50

literal notation, 400–401

metacharacters:

- alternation, 441–46
- anchors, 435–41
- capturing, 443–46
- character class, 419–22
- clustering, 442–43
- dot metacharacter, 417–18
- grouping, 442–43
- list of, 415–17
- metasymbols, 422–26
- overview, 414–17
- pattern matches, metacharacters to repeat, 426–35
- quantifiers, 426
- remembering, 443–46
- zero-width assertions, 435–41

overview, 399–400

RegExp() object:

- class properties, 405–6
- exec()* method, 401–2, 404–5
- instance properties, 405–6
- overview, 401–2
- properties, 405–8
- test()* method, 401–4

search patterns, options used for modifying, 400

string methods using, 408–14

String object

- match()* method, 408–9
- overview, 408
- replace()* method, 410–11, 453–56
- search()* method, 409–10
- split()* method, 412–14
- testing, 401–5

Relative positioning, 537–39

Reload() method, 245

Replace() method:
 location object, 245
 String object, 188–89, 410–11, 453–56

Required elements for tags, 585

Reserved words, 17–18

Reset() method:
 CGI.pm module, 692
 submitting fillout forms, 290–91

Resizing a window, 220–22

Return (*v*), 33

Return statement, 120–23

Return values, 120–23, 358–60

Right attribute, 533–36

Rollovers, 370–71

Rollover with mouse event, 336–38

Root directory, file URLs and server's, 611

Rounding numbers up and down, 174–76

Round() method, 175

Runtime errors, 24

S

Screen object, 250–51

Script execution, 16

Script setup:
 generating HTML, 21–23
 strings, 21
 writeIn() method, 21–23
 write() method, 21–23

HTML document and:
 generally, 15
 hello.html file, 15
 script execution, 16

syntax issues:
 case sensitivity, 17
 comments, 19
 reserved words, 17–18
 <*script*> tag, 19–20
 semicolons, 18
 statements, 18
 whitespace, 17

<*script*> tag, 19–20, 112

Scrolling in windows, 228–30, 551–54

ScrollTo() method, 228–31

Search and replace methods, 188–89

Search() method, 188–89, 409–10

Search patterns, options used for modifying, 400

Secure attribute, 482

Select object:
 input devices for forms, programming, 304–11

 methods, 305
 multiple selects, 309–11
 overview, 304
 properties, 304–5

Selectors, 496–97

Semicolons, 18

Server:
 communication between server and script:
 environment variables, 622–23
 HTML file with link to CGI script, 624–26
 overview, 621–22
 cookies retrieved from, 488–89
 HTTP server, 606–7
 root directory, file URLs and server's, 611
 SMTP server, 648–53
 SSI (Server Side Includes), 658–61

SetInterval() method, 222–27

SetTimeout() method, 222–26

Shift() method, 162

Shockwave Flash, checking for, 208

Shortcut assignment operators, 62–63

Siblings, 543

Sign-propagating right shift (>>) operator, 82

Simple CGI script, 616–19

Simple HTML document, 612–13

Single quotation mark (') , 32

Sizing and placing graphics and images, 591–92

Slice() method, 163–64

Slide show creation, 371–74

SMTP server and form processing, 648–53

Social Security numbers, checking for valid, 457–58

Source code, 602

Spaces and dashes in forms, removing, 453

<*span*> tag, 525–27

Special operators, 91

Splice() method:
 array objects, 164–65
 String object, 187

Split() method, 187–88, 412–14

Square root, 174

Src property, replacing images dynamically with,
 333–35

SSI (Server Side Includes), 658–61

Statements, 18

Stein, Lincoln, 661

String literals, 32–34

String manipulation methods, 184

String object:
 charAt() method, 187–88

String object (*cont.*):

- extraction of substrings from a string, methods
 - for, 187–88
 - HTML methods, 182–83
 - indexOf()* method, 185–87
 - lastIndexOf()* method, 185–87
 - length* property, 179–80
 - match()* method, 408–9
 - methods, 182–89
 - overview, 177–78, 408
 - position in a string, finding, 184–87
 - properties, 179–81
 - prototype* property, 179, 181
 - replace()* method, 188–89, 410–11, 453–56
 - search and replace methods, 188–89
 - search()* method, 188–89, 409–10
 - splice()* method, 187
 - split()* method, 187–88, 412–14
 - string manipulation methods, 184
 - substring()* method, 187
 - substr()* method, 187–88
 - with* keyword, 197
- Strings, 21
- bugs to watch out for, 44
 - comparing, 70–72
 - concatenation of, 21, 34
 - conversion to numbers, 85–87, 89–90
 - eval()* method used to convert strings to numbers, 89–90
 - parseInt()* method used to convert strings to numbers, 85–87
 - using quotes with, 21
- Style* object, 554–62
- CSS:
- color changed with *className* property, 560–62
 - overview, 554–58
 - properties, 555–56
 - text positioned with *style* property, 558–60
- Style sheets, 600–601
- CSS. *See* CSS.
- overview, 495
- positioning elements and layers
- absolute positioning, 531–36
 - bottom* attribute, 533–36
 - `<div>` tag, 536–37
 - left* attribute, 533–36
 - overview, 530–31
 - relative positioning, 537–39

- right* attribute, 533–36
 - top* attribute, 533–36
 - z-index*, 539–40
 - types of, 515–40
- Submit()* method:
- CGI.pm* module, 692
 - submitting fillout forms, 290–91
- Submitting fillout forms:
- with *button* input type, 288–89
 - with event handlers, 281–88
 - image, submitting form with, 280–81
 - with JavaScript, 281–88
 - overview, 279–80
 - pop-up window, displaying form content in, 292–93
 - reset()* method, 290–91
 - submit()* method, 290–91
- Substring()* method, 187
- Substr()* method, 187–88
- Switch* statement, 97–100
- Syntax issues:
- case sensitivity, 17
 - comments, 19
 - reserved words, 17–18
 - `<script>` tag, 19–20
 - semicolons, 18
 - statements, 18
 - whitespace, 17

T

- Tab (*v*), 32
- Tables, 588
- Tags:
- `<BODY>` tag, 586
 - `<HEAD>` tag, 585, 586
 - HTML, 612, 614–15
 - `<HTML>` tag, 585
 - order of required elements and their tags, 585
 - overview, 584
 - required elements, 585
 - to separate text, 587
 - `<TITLE>` tag, 585
- Test()* method, 401–4
- Text alignment properties, 508–9
- Text alternative for graphics and images, creating, 592
- Textarea* object, 301–4
- Textfield()* method, 678–82
- Text* object, 294–98

Text positioned with *style* property, 558–60
This keyword, 287–88, 377–78
 Timed events, creating, 222–27
 <*TITLE*> tag, 585
ToFixed() method, 192–93
 Toolbars, collapsing, 241–43
 Tool tips, 573–74
Top attribute, 533–36
Top property used to keep main window out of a frame, 240–41
 Troubleshooting, 597–99
 Turning off cookies, 478
Type attribute, 515–18
Typeof operator, 35–36, 112

U

Unescape() function, 483–84
Unshift() method, 162
 URLs, 244–45, 609–11
 User-defined function, 133
 User-defined objects:
 creating, 133
 methods defined for an object, 134–39
 new operator, 131
 Object() constructor, 131–32
 object literals, 140–41
 user-defined function used to create, 133
 \uXXXX escape sequence, 33

V

Validation of forms:
 alphabetic characters, checking for, 321–24
 e-mail addresses, checking, 324–26
 empty fields, checking for, 319–21
 overview, 319
 password fields, checking, 326–28
 Valid names for variables, 37–38
 Valid type of credit card, checking for, 464–70
 Variables:
 bugs to watch out for, 44
 concatenation and, 42–44
 conversion to data types, 40–41
 declaring, 38–39
 global variables, 42
 initializing, 38–39
 invalid names for, 38
 local variables, 42
 loosely typed language, JavaScript as, 40–41
 overview, 37

 scope of, 42, 119–20
 valid names for, 37–38
 Version of browser, 205
 Versions of JScript, 6
 Visibility, setting, 569–75
 Visibility property:
 drop-down menus, 569–72
 overview, 569
 tool tips, 573–74

W

W3C and DOM, 253
 Web resources, 581
While loop, 101–2
 Whitespace, 17
 Window events:
 blur() method, 365–67
 focus() method, 365–67
 list of, 361
 onBlur event handler, 363–65
 onFocus event handler, 363–65
 onLoad event handler, 361–63
 onload() method, 363
 onUnload event handler, 361–63
 onunload() method, 363
 overview, 360–61
Window object:
 history object, 247–49
 methods, 215–16
 opening windows, 216–20
 open() method, 216–20
 overview, 214
 properties, 214–15
 resizing a window, 220–22
 screen object, 250–51
 scrolling in windows, 228–30
 scrollTo() method, 228–31
 setInterval() method, 222–27
 setTimeout() method, 222–26
 timed events, creating, 222–27
With keyword, 197
 World Wide Web Consortium, 253
 Wrapper objects:
 Boolean object, 193–94
 Function object, 195–96
 Number object, 190–93
 overview, 177
 String object, 177–89

WriteIn() method, 21–23

Write() method, 21–23, 259–60

X

`\XXX` escape sequence, 33

`\xXX` escape sequence, 33

Z

Zero-fill right shift (`>>>>`) operator, 82

Zero-width assertions, 435–41

Z-index:

described, 539–40

and dynamic positioning, 567–68

Zip codes, checking for numeric, 448–50