

Index

NUMBERS

1&1 web hosting company, 762

A

activation objects, 85

ActiveXObject type, 524

additive operators, 56–58

addRule () method, 334–335

ADsafe, 771

advanced techniques, 589–616

custom events, 606–609

drag and drop, 609–615

functions. See functions, advanced

introducing, 589

summary of, 615–616

timers. See timers

AES cryptography, 763

Ajax (Asynchronous JavaScript + XML), 567–587

cross-domain requests and, 578–581

general libraries supporting, 759

JSON and, 581–586

Rico and, 761

summary of, 587

XHR objects in. See XHR (`XMLHttpRequest`) object

“Ajax: A New Approach to Web Applications”, 567

AjaxDoc, 770

alert () method, 213–214

algorithms, 650–651

almost standards mode, 19–21

AND bitwise operator, 47–48

AND logical operator, 52–53

Ant build tool, 661–662, 665

ANY_TYPE constant, 533

APIs (application programming interfaces), 669–702

HTML 5. See HTML 5

introducing, 669

Prototype library and, 760

Selectors, 669–672

summary of, 702

appendChild () method, 266–267

application logic/event handlers, 641–642

application programming interfaces. See APIs (application programming interfaces)

apply () method, 128–130

Aptana Debugger

as debugging tool, 767

overview of, 513

arc attribute, 13–14

arguments

in ECMAScript 4, 721–722

function, 74–75

functions accepting, 125

passing, 82–83, 99, 176

arguments object, 126–127

Array.isArray () method, 749

Array.prototype , 164

arrays

chunking of, 603–604

comprehensions in, 712–713

extras in, 706–708

literals, 100, 656–657

reductions for, 716

reference types and, 100–102

slicing, 737–738

string generics and, 708

assignments

destructuring, 713–714

operators, 62–63

asynchronous file loading, 528

Asynchronous JavaScript + XML (Ajax), 567–587

cross-domain requests and, 578–581

general libraries supporting, 759

JSON and, 581–586

Rico and, 761

summary of, 587

XHR objects in. See XHR (`XMLHttpRequest`) object

asynchronous XHR requests, 569–571

Attr type, 296–297

attributes

- of Element type, 285–286
- of EX4, 553–554
- in mutation events, 406

automatic tab forward, 449–450

B

BEA Systems, 547

beforeunload event, 409

best practices, 635–667

- for deployment, 660–666
- maintainability of code. See maintainable code
- overview of, 635
- summary of, 666–667

best practices, for performance, 647–660

- approaches to, 650–655
- array literals in, 656–657
- collecting garbage and, 93
- for DOM interactions, 657–660
- double interpretation and, 654–655
- introducing, 647–648
- iterative values in, 656
- loops, optimizing, 651–652
- loops, unrolling, 652–654
- multiple variable declarations in, 655–656
- object literals in, 656–657
- for property lookups, 650–651
- scopes and, 648–649
- speed considerations in, 655
- statement count in, 655–657

bind () function, 595

bitwise operators, 45–51

- AND, 47–48
- left shift, 49–50
- NOT, 47
- OR, 48
- overview of, 45–47
- signed right shift, 50
- unsigned right shift, 51
- XOR, 49

blocking

- characters, 446–447
- pop-up windows, 210–211

block-level scopes

- identifier lookups and, 89–90

- in JavaScript 1.7, 709–710
- let keyword and, 709–710
- variable declarations and, 88–89

blur in scripting forms, 439–441

BOM (Browser Object Model), 201–227

- defined, 3, 9–10
- history object in, 226–227
- introducing, 201
- location object in, 216–219
- navigator object in, 219–224
- screen object in, 224–226
- summary of, 227
- window object in. See window object

Boolean data types, 30–31

Boolean operators, 51–54

- logical AND, 52–53
- logical NOT, 51–52
- logical OR, 53–54
- overview of, 51

Boolean primitive wrappers, 131–132

break statement, 67–69

breakpoints

- in Drosera, 508–509
- in Firebug, 503–504, 766
- in Internet Explorer debugger, 498
- in Opera JavaScript debugger, 511

browser error reporting

- Chrome, 472–473
- Firefox, 467–469
- Internet Explorer, 465–467
- Opera, 470–472
- Safari, 469–470

Browser Object Model (BOM), 201–227

- defined, 3, 9–10
- history object in, 226–227
- introducing, 201
- location object in, 216–219
- navigator object in, 219–224
- screen object in, 224–226
- summary of, 227
- window object in. See window object

browsers

- debuggers and, 765
- errors and. See browser error reporting
- identifying, 245–247
- libraries for bridging differences, 759
- user-agent detection and, 233–234
- XHR objects and, 575–578

bubble phase (false), 370, 376

bubbling, events, 366–367, 403–405

build process, 660–662

built-in objects, 142–148

button property, 389–390

<button> element, 434–439

C

cached, 19

Caja, 771

call () method, 128–130

<canvas> element, 682–692

context transformations in, 688–691

images, working with, 691–692

introducing, 682

paths, drawing, 685–687

rectangles, drawing, 683–685

text, drawing, 687–688

capture phase (true), 370

capturing of events, 367

capturing-group matching, 118–121

Cascading Style Sheets (CSS)

Aptana Debugger and, 767

creating rules for, 334–335

decoupling, 640–641

deleting rules in, 335

in DOM Levels 2 and 3, 333–334

dynamic styles in, 309

jQuery and, 760

script.aculo.us and, 762

style sheets of, 331–333

`<style>` element and, 326–329

Yahoo! User Interface (YUI) library, 759

case methods, 137

case sensitivity, 23, 137

catch statements, 87–88

CDATASection type, 293–294

chaining, prototypes. See prototype

chaining

character codes, 395

character methods, 134

characters, blocking, 446–447

charset attribute, 13

children

of Document type in DOM, 264–267,
269–271

of Element type, 288–289

memory and performance issues of, 306

property, 298–299

in XML attributes, 554

in XML object hierarchy, 553

Chrome

custom error messages in, 479

debugging, 510, 513

document.doctype in, 270

DOM support in, 9

ECMAScript compliance of, 6

error reporting in, 472–473

inheritance in, 112

timer mechanism of, 600

user-agent detection, 238

Web Inspector as debugger for, 767

chunk () method, 603–604

circular references, 92

classes, ECMAScript 4, 726–729

dynamic, 728–729

prototypes of, 727

static members of, 729

virtual properties of, 727–728

classList property, 673–675

class-related additions, in HTML 5

classList property in, 673–675

getElementsByClassName() method
in, 673

introducing, 673

click events

in event bubbling, 366–367

in event delegation, 422–423

in event handlers, 368–372

and event object, 376–378

in mouse events, 383–389

client coordinates, 384–385

client detection, 229–259

capability detection for, 229–231

introducing, 229

quirks detection for, 231–232

summary of, 258–259

user-agent detection for. See user-agent
detection

client dimensions, 337–338

client-side storage, 617–634

cookies for. See cookies

globalStorage objects in, 631–632

Internet Explorer user data in, 627–628

introducing, 617

limits and restrictions on, 634

localStorage objects in, 632–633

overview of, 628

client-side storage (continued)

- sessionStorage objects in, 629–630
- storage events in, 633
- Storage type in, 628–629
- StorageItem type in, 633
- summary of, 632

clipboards, 447–449

cloning

- Internet Explorer ranges, 362
- nodes in DOM, 268
- ranges in DOM Levels 2 and 3, 358

close () method, 277–278

closures, expression, 715

code, maintainable, 635–647

- conventions for, 636
- defined, 636
- loose coupling and, 639–642
- naming functions and variables in, 637
- overview, 635
- programming, 642
- readability of, 636–637
- variable type transparency in, 637–639

code conventions, 636–639

- in naming functions and variables, 637
- readability and, 636–637
- variable type transparency and, 637–639

code injection, 144

collapsing

- Internet Explorer ranges, 361–362
- ranges in DOM Levels 2 and 3, 356–357

comma operators, 63

Command Line tab, 512

Comment type, 292–293

comments, 24

communications errors, 481, 485–486

compareBoundaryPoints () method, 361

complex selection, in ranges

- in DOM Levels 2 and 3, 351–353
- in Internet Explorer, 360

complex XSLT transformations, 539–543

compound assignment operators, 62–63

comprehensions, array, 712–713

compression. See also crunchers

- of files, 664–665
- HTTP, 665–666
- introducing, 663
- JSMIn for, 768

computed styles, 329–331

concat () method, 108, 135

conditional operators, 62

confirm () method, 214

conformance

- Document type in DOM and, 275–276
- in ECMAScript, 5

Console, in Internet Explorer debugger, 501–502

constant values, 650–651

constants, 704

constructor patterns

- durable, 169
- as functions, 154
- overview of, 152–153
- parasitic, 167–168
- problems with, 154–155
- prototype patterns and, 166

constructor properties, 120–121

constructor stealing, 175–177

- introducing, 175
- passing arguments in, 176
- problems with, 176
- prototype chaining and, 176–177

contains () method, 299–300

content manipulation, 300–306

context transformations, 688–691

contextmenu, 407–409

continue statement, 67–69

control statements, 25

conversion

- decimals in ECMAScript 3.1, 755
- reference types, 102–103
- types in ECMAScript 4, 725–726

convertToArray, 265

cookies, 617–627

- HTTP-only, 626
- in JavaScript, 619–622
- overview of, 617–618
- parts of, 618–619
- restrictions on, 618
- security of, 627
- size of, 626–627
- subcookies, 622–626

coordinates

- client, 384–385
- screen, 386

coupling, loose

- application logic/event handlers, 641–642
- CSS/JavaScript, 640–641
- HTML/JavaScript, 639–640

createElement () method, 287–288

createTextNode () method, 290–291

Crockford, Douglas

- as author of JSLint validator, 768
- as author of JSMIn cruncher, 768
- on durable objects, 169
- JavaScript JSON library of, 586
- JSLint by, 662
- on prototypal inheritance, 178

cross-browser

- event handlers, 373–375
- event objects, 381–382
- support for XML in JavaScript, 528–529
- XML DOM support, 528–529
- XPath, 536–538
- XSLT, 545–546

cross-domain requests, 578–581

cross-site request forgery (CSRF) attacks, 577–578

cross-site scripting (XSS) attacks, 578

crunchers. See also compression

- Dojo ShrinkSafe, 769
- JSMIn, 768
- YUI Compressor, 769

cryptography libraries

- JavaScript MD5, 763
- JavaScript, 763

CSRF (cross-site request forgery) attacks, 577–578

CSS (Cascading Style Sheets)

- Aptana Debugger and, 767
- creating rules for, 334–335
- decoupling, 640–641
- deleting rules in, 335
- in DOM Levels 2 and 3, 333–334
- dynamic styles in, 309
- jQuery and, 760
- script.aculo.us and, 762
- style sheets of, 331–333
- `<style>` element and, 326–329
- Yahoo! User Interface (YUI) library, 759

cssText property, 328–329

curry () function, 596–597

cursor trails, 609

custom events, 606–609

D

data attributes, 675–676

data types

- Boolean, 30–31

defined, 28

for error handling, 481, 483–484

functions, 72–76

null, 30

Null, 30

Number, 31–37

Object, 40–41

operators, 41

statements. See statements

String, 37

typeof operator, 28

undefined, 28–30

Undefined, 28–30

database storage, 694–696

dataTransfer object

additional members of, 700

drag and drop in HTML 5, 698–700

dropEffect property of, 698–699

EffectAllowed property of, 699–700

dates

in ECMAScript 4, 739–740

formatting methods, 112–113

reference types and, 109–111, 113–114

debugging techniques, 488

error handling, 488–492

introducing, 488

logging messages to consoles, 488–491

logging messages to pages, 491

throwing errors, 491–492

debugging tools, 492–513

Aptana Debugger, 513, 767

for Chrome, 510, 513

Chrome JavaScript Debugger, 513

Drosera, 507–510, 767

Microsoft Script Debugger, 765

Microsoft Script Editor, 766

Opera JavaScript debugger, 510–512

overview of, 492–496

Venkman (Firefox), 513, 767

Visual Studio (Internet Explorer), 513

Visual Studio.NET, 766

Visual Web Developer Express, 766

Web Inspector, 767

debugging tools, Firebug, 502–507

breakpoints in, 503–504

browser error reporting, 467–469

as debugging tool, 766

JavaScript Console in, 505–506

logging function calls and, 506–507

overview of, 502–503

debugging tools, Firebug (continued)

- `select` events in, 443
- stepping through code in, 504–505
- watches in, 505

debugging tools, IE Debugger, 496–502

- breakpoints in, 498
- code in, 499–500
- Console in, 501–502
- overview of, 496–497
- watches in, 500–501

decimals, in ECMAScript 3.1, 752–756

- comparisons of, 754
- conversion of, 755
- math, 755
- methods of, 755–756
- operations of, 753–754
- overview of, 752

declarations vs. expressions, 124

decoupling

- application logic/event handlers, 641–642
- CSS/JavaScript, 640–641
- HTML/JavaScript, 639–640

decrement operators, 41–43

deep types, 723

default prototypes, 171–172

defer attribute, 13, 22

deferred scripts, 16–17

delegation of events, 422–423

deleteRule () method, 335

deployment, 660–666

- build process in, 660–662
- compression for, 663–666
- introducing best practices for, 660
- validation and, 662–663

deprecated syntax, 18–19

dereferencing values, 93–94

descendants () method, 556–557

destructuring assignments, 713–714

detail in mouse events, 391

Develop menu, 469–470

developers, Visual Web Developer

Express, 766

devices, 396–397

Dhakar, Lokesh, 762

DHTML (Dynamic HyperText Markup Language), 7

dimensions, DOM Levels 2 and 3, 336–342

- client, 337–338

determining, 340–342

offset, 336–337

scroll, 339–340

divide operator, 55

document element, 262

document modes, 19–21

Document Object Model. See DOM (Document Object Model)

Document type, 269–278

- child nodes and, 269–271
- collections of, special, 275
- conformance detection for, 275–276
- in DOM Levels 2 and 3, 320–324
- element retrieval, 272–275
- introducing, 269–278
- properties of, information in, 271–272
- writing with, 277–278

documentation generators

- AjaxDoc, 770
- JsDoc Toolkit, 770
- YUI Doc, 770

document.exeCommand (), 459

DocumentFragment type, 294–295

DocumentType type

- in DOM, 294
- in DOM Levels 2 and 3, 322

Dojo Object Harness (DOH), 770

Dojo ShrinkSafe, 769

Dojo Toolkit, 760

DOM (Document Object Model), 261–315

- Attr type in, 296–297
- CDATASection type in, 293–294
- Comment type in, 292–293
- defined, 3
- Document type in. See Document type
- DocumentFragment type in, 294–295
- DocumentType type in, 294
- dynamic scripts with, 307–309
- dynamic styles with, 309–310
- Element type in, see Element type
- event object, 375–378
- general libraries supporting, 759
- introducing, 11, 261
- Level 0, 369–370
- Level 1, 8
- other languages and, 8–9
- overview of, 7–8
- script.aculo.us and, 762
- simulating events, 426

- summary of, 314–315
- tables and, 311–313
- Text type in, 289–292
- Web browsers supporting, 9
- working with, 307–314
- DOM extensions, 297–306**
 - children property, 298–299
 - contains () method, 299–300
 - for content manipulation, 300–306
 - innerHTML property, 302–304
 - innerText property, 300–302
 - introducing, 297
 - memory and performance of, 306
 - outerHTML property, 305–306
 - outerText property, 305
 - rendering modes with, 297
 - scrolling with, 298
- DOM interactions**
 - event delegation, using, 659
 - innerHTML, using, 658–659
 - live updates, minimizing, 657–658
 - NodeList objects, pitfalls of, 659–660
 - performance best practices, 657–660
- DOM Level 2, 317–364**
 - browsers supporting, 515–516
 - changes in, 317–318
 - defined, 8
 - Document in, 322–324
 - in Document type, 320–324
 - DocumentType type in, 322
 - in Element type, 321
 - event handlers, 370–372
 - frames in, 325–326
 - introducing, 317
 - in Node type, 319–320
 - NodeIterator type in, 344–347
 - ranges in. See ranges, DOM Levels 2 and 3
 - styles in. See styles, DOM Levels 2 and 3
 - summary of, 362–363
 - traversals in, 342–344
 - TreeWalker type, 347–349
 - XML namespaces in, 318–322
- DOM Level 3, 317–364**
 - changes in, 317–318
 - defined, 8
 - Document in, 322–324
 - in Document type, 320–324
 - DocumentType type in, 322
 - in Element type, 321
- frames in, 325–326
- introducing, 317
- in Node type, 319–320
- NodeIterator type in, 344–347
- ranges. See ranges, DOM Levels 2 and 3
- styles. See styles, DOM Levels 2 and 3
- summary of, 362–363
- traversals in, 342–344
- TreeWalker type, 347–349
- XML namespaces in, 318–322
- XPath, 530–533
- DOM Level 3 Load and Save, 518–523**
 - overview of, 518
 - parsing files from URIs, 521–522
 - parsing with context, 522
 - parsing XML in, 518–521
 - serializing XML and, 523
- DOM nodes**
 - cloneNode () method for, 268
 - hierarchy of, 261–262
 - inserting, 405
 - in Levels 2 and 3, 319–320, 324–325
 - manipulating, 266–268
 - NodeIterator type, 344–347
 - NodeLists, 264–265, 314
 - nodeName property, 264
 - nodeValue property, 264
 - normalizing, 268, 291–292
 - relationships of, 264–266
 - removing, 403–405
 - types of, 263–269
- DOM Storage, 628–634. See also client-side storage**
 - globalStorage objects in, 631–632
 - limits and restrictions on, 634
 - localStorage objects in, 632–633
 - overview of, 628
 - sessionStorage objects in, 629–630
 - storage events in, 633
 - Storage type in, 628–629
 - StorageItem type in, 633
- domain property, 271–272**
- DOMContentLoaded, 413**
- DOMMouseScroll, 411–412**
- DomParser type, 516–517**
- do-while statement, 64**
- drag and drop, 609–615**
 - adding custom events and, 613–615
 - fixing drag functionality, 611–613
 - overview of, 609–611

drag and drop, in HTML 5, 696–700

- additional members and, 700
- custom drop targets, 697–698
- `dataTransfer` object, 698–700
- `dropEffect` property in, 698–699
- `EffectAllowed` property in, 699–700
- events of, 696–697
- introducing, 696

drawing, with `<canvas>` element

- images and, 691–692
- paths, 685–687
- rectangles, 683–685
- text, 687–688
- transformations for, 688–691

`dropEffect` property, 698–699

Drosera debugger, 507–510

- breakpoints in, 508–509
- as debugging tool, 767
- error handling, 507–510
- overview of, 507–508
- stepping through code in, 509–510

Duff, Tom, 653

Duff's device, 653–654

durable constructor patterns, 169

durable objects, 169

dynamic classes, 728–729

Dynamic HyperText Markup Language (DHTML), 7

dynamic properties, 80

dynamic prototype patterns, 163–164, 166–167

dynamic scripts, 307–309

dynamic styles, 309–310

E

Eclipse IDE, 767

ECMA (European Computer Manufacturers Association), 2, 5

ECMAScript, 23–77

- conformance in, 5
- as Core, 3
- data types. See data types
- editions of, 4
- functions, 72–76
- introducing, 23
- keywords, 25–26
- operators. See operators
- overview of, 3–4, 11
- reserved words, 26

- statements. See statements
- summary of, 76
- syntax, 23–25
- variables, 26–27
- Web browsers supporting, 5–6

ECMAScript 3.1, 741–757

- `Array.isArray()` method in, 749
- decimals in, 752–756
- functions in, 748–749
- future of, 757
- native JSON support in, 750–752
- object creation in, 744–745
- object internals in, 741–742
- object lock-down methods in, 747–748
- object properties in, 743–744
- property names, retrieving, 746
- prototype information, accessing, 742–743
- static object methods in, 742
- `toISOString()` to `Date` objects in, 750
- `trim()` method in, 749–750
- usage subsets in, 756–757

ECMAScript 4, 717–741

- array slicing, 737–738
- classes in, 726–729
- `Date` object, 739–740
- error type detection, 737
- `for-each-in`, 734–735
- functions, 720–722, 724
- future of, 741
- generic functions, 722
- inheritance in, 730–732
- interfaces in, 730
- keyword usage and, 739
- language changes in, 734–741
- `Map` class, 740–741
- multiline strings, 737
- namespaces in, 732–733
- operator overloading, 735
- packages in, 734
- property enumerations, 738
- regular expressions, 738
- with statements, 735
- strict mode, 736
- string indexing, 737–738
- `this` statements, 736
- type definition in, 723–726
- variable typing, 717–720
- `Vector` class, 740

ECMAScript for XML (EX4), 547–565

- attributes of, 553–554
- enabling full, 564
- for-each-in loop, 563
- introducing, 547
- isXMLName (), 563–564
- Namespace type in, 550–551
- namespaces generally, 561–563
- node types in, 555–556
- parsing with, 560–561
- QName type in, 551–552
- querying, 556–557
- serializing with, 560–561
- summary of, 564–565
- typeof operator, 564
- types in, 547–552
- usage generally, 552–553
- XML construction and manipulation with, 558–560
- XML type in, 547–548
- XMLElement type in, 549–550

editions of JavaScript. See evolution of JavaScript**EffectAllowed property, 699–700****effects libraries**

- Lightbox, 762
- moo.fx, 762
- script.aculo.us, 762

Eich, Brendan, 2**element styles. See styles, DOM Levels 2 and 3****Element type, 279–289**

- attributes property, 285–286
- children of, 288–289
- creating, 287–288
- getAttribute(), 282–284
- HTMLElement, 280–282
- introducing, 279
- in Levels 2 and 3, 321
- setAttribute(), 284

elements

- in mouse events, 387–389
- retrieval of, 272–275

encoding URI, 142–143**engine detection, 240–244, 246****equality operators, 60–62****error handling, 465–513**

- Chrome error reporting, 472–473
- for communications errors, 481, 485–486
- for data type errors, 481, 483–484

- debugging techniques and, 488–492
- debugging tools generally, 492–496
- Drosera debugger, 507–510
- error event in, 480–481
- error type detection in, 737
- for errors in Internet Explorer, 492–496
- fatal vs. nonfatal errors in, 486–487
- finally clause for, 474–475
- Firefox debugger. See Firebug
- Firefox error reporting, 467–469
- introducing, 465, 473
- location of errors in, 481
- for log errors to server, 487–488
- Opera error reporting, 470–472
- Opera JavaScript debugger, 510–512
- Safari error reporting, 469–470
- strategies for, 481
- summary of, 513
- throw operator for, 477–479
- try-catch statement for, 474, 477, 479
- for type coercion errors, 481–482
- types of errors in, 475–477

error handling, Internet Explorer (IE), 492–496.**See also Internet Explorer (IE) debugger**

- invalid character, 494
- member not found, 494–495
- operation aborted, 493–494
- reporting, 465–467
- syntax, 495–496
- system cannot locate the resource specified, 496
- unknown runtime, 495

error reporting, browsers

- Chrome, 472–473
- Firefox, 467–469
- Internet Explorer, 465–467
- Opera, 470–472
- Safari, 469–470

Espresso Pages, 2**European Computer Manufacturers Association (ECMA), 2, 5****eval () method, 144****evaluate () method in XPath, 530–533****event handlers**

- cross-browser, 373–375
- decoupling, 641–642
- DOM Level 0, 369–370
- DOM Level 2, 370–372
- HTML, 368–369
- Internet Explorer, 372–373

event listeners. See event handlers

event object, 375–382

- cross-browser, 381–382
- DOM, 375–378
- Internet Explorer, 379–380

event simulation, 425–431

- DOM, 426
- HTML, 430
- Internet Explorer, 430–431
- keyboard, 428–429
- mouse, 426–428
- mutation, 430

events, 365–432

- bubbling, 366–367, 403–405
- capturing, 367
- delegation, 422–423
- DOM, 426
- drag and drop, 613–615, 696–697
- flow of, 365–368
- form-field, 440–441
- gesture, 420–421
- handling. See event handlers
- Internet Explorer, 430–431
- introducing, 365
- keyboard, 392–397, 428–429
- listeners, 368
- load, 398–400
- memory and performance of, 422
- mobile Safari, 417–421
- mutation, 402–407, 430
- object, see event object
- orientationchange, 418–419
- removing event handlers, 424–425
- resize, 401–402
- Safari events, 417–421
- scroll, 402
- simulating, 425–431
- summary of, 432
- touch, 419–420
- types of, 383
- UI, 383
- unload, 401

events, click

- in event bubbling, 366–367
- in event delegation, 422–423
- in event handlers, 368–372
- and event object, 376–378
- in mouse events, 383–389

events, HTML, 397–402, 430

- load, 398–400

- resize, 401–402

- scroll, 402

- unload, 401

events, keyboard, 392–397

- character codes in, 395
- on devices, 396–397
- introducing, 392–393
- key codes in, 393–394
- simulating, 428–429
- textInput, 395–396

events, mouse, 383–392

- accessibility and, 391–392
- button property in, 389–390
- client coordinates in, 384–385
- detail for information about, 391
- elements related to, 387–389
- in Internet Explorer, 391
- mobile Safari support for, 391
- modifier keys for, 387
- mousewheel, 410–412
- screen coordinates in, 386
- simulating, 426–428

events, mutation, 402–407

- attribute changes, 406
- node insertion, 405
- node removal, 403–405
- simulating, 430
- text changes, 406–407

events, proprietary, 407–417

- beforeunload, 409
- contextmenu, 407–409
- DOMContentLoaded, 413
- DOMMouseScroll, 411–412
- mousewheel, 410–412
- pageshow and pagehide, 416–417
- readystatechange, 413–416

EventTarget (), 606–608, 613–615

evolution of ECMAScript. See also ECMAScript 4

- ECMAScript 3.1 in, 757
- ECMAScript in, 3–6

evolution of JavaScript, 703–757

- introducing, 1–2, 703
- JavaScript 1.5, 704–705
- JavaScript 1.6, 706–708
- JavaScript 1.7, 709–714
- JavaScript 1.8, 714–716
- JavaScript 1.9, 717
- summary of, 757

EX4 (ECMAScript for XML), 547–565

- attributes of, 553–554
- enabling full, 564
- for-each-in loop, 563
- introducing, 547
- isXMLName (), 563–564
- Namespace type in, 550–551
- namespaces generally, 561–563
- node types in, 555–556
- parsing with, 560–561
- QName type in, 551–552
- querying, 556–557
- serializing with, 560–561
- summary of, 564–565
- typeof operator, 564
- types in, 547–552
- usage generally, 552–553
- XML construction and manipulation with, 558–560
- XML type in, 547–548
- XMLElement type in, 549–550

exec () method, 118**execution context and scope, 84–87****expressions**

- closures, 715
- generator, 715–716

Ext JS, 761**Extensible HyperText Markup Language (XHTML), 17–18****extensions, DOM, 297–306**

- children property, 298–299
- contains () method, 299–300
- for content manipulation, 300–306
- innerHTML property, 302–304
- innerText property, 300–302
- introducing, 297
- memory and performance of, 306
- outerHTML property, 305–306
- outerText property, 305
- rendering modes with, 297
- scrolling with, 298

F**factorial functions, 126–127****factory patterns, 152****false Boolean value, 30–31****fatal vs. nonfatal errors, 486–487****fields in forms. See form-fields****FIFO (first-in-first-out) structures, 105****filter argument in NodeIterator, 344–345****filtering input, 445–449****final classes and methods, 732****finally clause, 474–475****findText () method, 359****Firebug, 502–507**

- breakpoints in, 503–504
- browser error reporting, 467–469
- as debugging tool, 766
- JavaScript Console in, 505–506
- logging function calls and, 506–507
- overview of, 502–503
- select events in, 443
- stepping through code in, 504–505
- watches in, 505

Firebug Working Group, 766**Firefox**

- Aptana Debugger and, 767
- custom error messages in, 478
- debuggers and, 765
- document.doctype in, 270
- DOM storage in, 628
- DOM support in, 9
- draggesture in, 697
- ECMAScript compliance of, 6
- error event in, 480–481
- getUserData () method in, 325
- inheritance in, 111
- innerHTML in, 304
- JavaScript in, 10
- MimeType objects in, 224
- progress event in, 577
- XML prologs in, 548

Firefox debugger. See Firebug**first-in-first-out (FIFO) structures, 105****floating-point values, 32–33****flow of events, 365–368****flow-control statements. See statements****focus in scripting forms, 439–441****for statement, 65–66****for-each-in loop**

- in ECMAScript 4, 734–735
- in EX4, 563

for-in statement, 66–67**form-fields**

- events of, 440–441
- methods of, 439
- properties of, 437–439
- in scripting generally, 436–437

forms, scripting, 433–464

- events of form-fields, 440–441
- fields of, 436–437
- methods of form-fields, 439
- overview of, 433–434
- properties of form-fields, 437–439
- resetting, 435–436
- rich text editing in, 458–464
- select boxes and, 450–456
- serializing forms, 456–458
- submitting, 434–435
- text boxes and. See text boxes, scripting

Fox, George, 648

frames

- in BOM window object, 202–205
- in DOM Levels 2 and 3, 325–326

fromCharCode () method, 141

Fuchs, Thomas, 762

function binding, 594–595

function currying, 596–598

function throttling, 604–606

functions

- arguments of, 74–75, 83
- as constructor patterns in OOP, 154
- in ECMAScript 3.1, 748–749
- naming, 637
- overloading, 76
- overview of, 72–74

functions, advanced, 589–598

- binding, 594–595
- currying, 596–598
- lazy loading, 592–594
- scope-safe constructors, 589–592

functions, as reference types, 122–130

- declarations vs. expressions, 124
- internals, 126–127
- overloading and, 123–124
- overview of, 122–123
- properties and methods, 128–130
- as values, 125–126

functions, ECMAScript 4

- in ECMAScript 4, 720–722, 724
- generic, 722
- optional arguments, 722
- rest arguments, 721–722

future-proofing, 19

G

game system identification, 253–255

garbage collection, 90

- mark-and-sweep, 91
- memory in, 93–94
- overview of, 90–91
- performance and, 93
- reference counting, 91–93

Garrett, Jesse James, 567

Gecko, 235–237

- Gecko-based browsers, 767

general libraries

- Dojo Toolkit, 760
- Ext JS, 761
- jQuery, 760
- MochiKit, 761
- MooTools, 760
- overview of, 759
- Prototype, 760
- Yahoo! User Interface (YUI) library, 759

generator expressions, 715–716

generators, 710–711

gesture events, 420–421

GET requests, 573–574

getAttribute(), 282–284

getBoundingClientRect(), 340–342

getElementsByClassName() method

- in class-related additions in HTML 5, 673
- in HTML 5, 673–675

getElementById method, 274–275

getPropertyValue(), 328–329

getters, 704–705

getTime () method, 113

getUserData () method, 325

Global built-in objects, 142–146

- eval () method, 144
- overview of, 142
- properties of, 145
- URI-encoding methods in, 142–143
- window object accessing, 145–146

global execution context, 85–86

global lookups, 648–649

global scope

- in BOM window object, 201–202
- in window object, 145–146

global variables, 27

globalStorage objects, 631–632

go () method, 226

Google Chrome. See **Chrome Greenberg, Jeff, 653**

H

handler object, 594–595

handler registration, 223–224

handling events. See **event handlers**

hash property, 218

heap, for storing objects, 79

Hewitt, Joe, 766

hierarchy of nodes, in DOM

Attr type in, 296–297

CDATASection type in, 293–294

cloneNode () method in, 268

Comment type in, 292–293

Document type in. See **Document type**

DocumentFragment type in, 294–295

DocumentType type in, 294

Element type in. See **Element type**

manipulating nodes in, 266–268

node relationships in, 264–266

node types in, 263

normalize () method, 268

overview of, 261–262

Text type in, 289–292

history object, 226–227

history of ECMAScript. See **evolution of ECMAScript**

host environments for ECMAScript, 3

HTML (HyperText Markup Language)

decoupling with JavaScript, 639–640

DOM and, 7

event handlers, 368–369

history of, 693

methods, 141–142

simulating events, 430

HTML (HyperText Markup Language),

JavaScript in, 13–22

document modes, 19–21

introducing, 13

<noscript> elements, 21–22

<script> elements, 13–19

summary, 22

HTML 5, 672–701

<canvas> element in. See **<canvas> element**

changes to, 693

character set properties in, 672–673

classList property in, 673–675

class-related additions in, 673–675

data attributes, customizing, 675–676

database storage in, 694–696

drag and drop with, 696–700

future of, 701

getElementsByClassName () method in, 673

introducing, 672

media elements in, 677–682

messaging, cross-document, 676–677

offline support for, 692–693

WebSocket type in, 700–701

HTML events, 397–402

load, 398–400

resize, 401–402

scroll, 402

unload, 401

HTMLCollection object, 273–275, 314

HTMLElement type, 280–282

HTMLFormElement, 433–434. See **also scripting forms**

HTMLFrameElement, 325–326

HTMLSelectElement type, 450–452

HTTP headers, 571–573

HTTP-only cookies, 626

Hungarian notation, 638

HyperText Markup Language. See **HTML (HyperText Markup Language)**

I

identifier lookups, 89–90

identifiers, 24

IE (Internet Explorer)

browser error reporting, 465–467

comment nodes in, 293

custom error messages in, 478

debuggers and, 765

Document type constructor and prototype in, 269–270

DOM Level 2 and 3 in, 318

DOM storage in, 628

DOM support in, 9

ECMAScript compliance of, 6

error event in, 480

event handlers, 372–373

event object, 379–380

garbage collection in, 93

IE (Internet Explorer) (continued)

- `getAttribute ()` method in, 284
 - inheritance in, 111
 - JavaScript objects in, 41
 - `lastIndex` in, 119
 - literals in, 657
 - Microsoft Script Debugger, 765
 - mouse events in, 391
 - queue methods in, 105
 - `setAttribute ()` method in, 284
 - simulating events, 430–431
 - user data in client-side storage, 627–628
 - version 3, 234
 - versions 4-8, 234–235
 - window object in, 202
 - XML DOM support in, 523–527
 - XML in JavaScript in, 523–527
 - XPath in, 535–536
 - XSLT in, 539–543
- ### IE (Internet Explorer) debugger, 496–502
- breakpoints in, 498
 - code in, 499–500
 - Console in, 501–502
 - overview of, 496–497
 - watches in, 500–501
- ### IE (Internet Explorer) ranges, 358–362
- cloning, 362
 - collapsing, 361–362
 - complex selection in, 360
 - interacting with, 360–361
 - introducing, 358–359
 - simple selection in, 359
- ### if statement, 63–64
- ### iframes, 325–326, 463–464
- ### images, 691–692
- ### implementations, 3–10
- ### importNode () method, 322–324
- ### in operator, 159–161
- ### increment/decrement operators, 41–43
- ### indexing methods
- select boxes and, 450–454
 - string, 737–738
 - for string location, 136–137
- ### inheritance, 170–181
- constructor stealing in, 175–177
 - in ECMAScript 4, 730–732
 - introducing, 170
 - of methods, 111–112
 - parasitic, 178–179
 - parasitic combination, 179–181
 - prototypal, 178
 - prototype chaining in, 170–175
- ### inheritance, ECMAScript 4, 730–732
- final classes and methods in, 732
 - introducing, 730–731
 - superclass methods, overriding, 731–732
- ### initialization statements, 124
- ### initializing variables, 30, 89
- ### inline code vs. external files, 19, 22
- ### innerHTML property, 302–304
- ### innerText property, 300–302
- ### input filtering, 445–449
- ### <input> element, 434–439
- ### insertBefore () method, 267
- ### insertRule () method, 334–335
- ### instance methods, 118–119
- ### instance properties, 117
- ### instances in prototype chaining, 172
- ### intellectual property issues, 660
- ### interactions, DOM
- event delegation, using, 659
 - innerHTML, using, 658–659
 - live updates, minimizing, 657–658
 - NodeList objects, pitfalls of, 659–660
 - performance best practices, 657–660
- ### interfaces, 726, 730
- ### internal functions as reference types, 126–127
- ### Internet application libraries
- qooxdoo, 762
 - Rico, 761
- ### Internet Explorer (IE)
- browser error reporting, 465–467
 - comment nodes in, 293
 - custom error messages in, 478
 - debugger. See Internet Explorer (IE) debugger
 - debuggers and, 765
 - Document type constructor and prototype in, 269–270
 - DOM Level 2 and 3 in, 318
 - DOM storage in, 628
 - DOM support in, 9
 - ECMAScript compliance of, 6
 - error event in, 480
 - event handlers, 372–373
 - event object, 379–380
 - garbage collection in, 93
 - `getAttribute ()` method in, 284
 - inheritance in, 111
 - JavaScript objects in, 41

- lastIndex in, 119
- literals in, 657
- Microsoft Script Debugger, 765
- mouse events in, 391
- queue methods in, 105
- ranges in. See Internet Explorer (IE) ranges
- setAttribute () method in, 284
- simulating events, 430–431
- user data in client-side storage, 627–628
- version 3, 234
- versions 4-8, 234–235
- window object in, 202
- XML DOM support in, 523–527
- XML in JavaScript in, 523–527
- XPath in, 535–536
- XSLT in, 539–543

Internet Explorer (IE) debugger, 496–502

- breakpoints in, 498
- code in, 499–500
- Console in, 501–502
- overview of, 496–497
- watches in, 500–501

Internet Explorer (IE) ranges, 358–362

- cloning, 362
- collapsing, 361–362
- complex selection in, 360
- interacting with, 360–361
- introducing, 358–359
- simple selection in, 359

intervals, in window object, 211–213

invalid character error, 494

iPhone, 251–253, 412, 417–421

iPod Touch, 251–253, 412

Ippolito, Bob, 761

isSupported () method, 324

isXMLName (), 563–564

iterative values, 656

iterators, 711–712

J

JavaScript, 1–11

- editions of, 10–11
- history of, 1–2
- implementations in, 3–10
- introducing, 1
- Lightbox and, 762
- summary of, 11
- Yahoo! User Interface (YUI) library, 759

JavaScript 1.5, 704–705

- constants in, 704
- getters in, 704–705
- setters in, 704–705

JavaScript 1.6, 706–708

- array and string generics in, 708
- array extras in, 706–708

JavaScript 1.7, 709–714

- array comprehensions in, 712–713
- block-level scopes in, 709–710
- destructuring assignments in, 713–714
- generators in, 710–711
- iterators in, 711–712
- scopes in, 709–710

JavaScript 1.8, 714–716

- array reductions in, 716
- expression closures in, 715
- generator expressions in, 715–716

JavaScript 1.9, 717

JavaScript Console, 505–506

JavaScript Lint, 768

JavaScript MD5, 763

JavaScript Object Notation (JSON), 581

- with Ajax, 583–586
- native support, 750–752
- overview of, 581–583
- security in, 567

“JavaScript Performance Issues”, 648

JavaScript tools

- ADsafe, 771
- AjaxDoc, 770
- Aptana Debugger, 767
- Caja, 771
- Dojo Object Harness (DOH), 770
- Dojo ShrinkSafe, 769
- Drosera, 767
- Firebug, 766
- JavaScript Lint, 768
- JsDoc Toolkit, 770
- JSLint, 768
- JSMIn, 768
- JsUnit, 769
- Microsoft Script Debugger, 765
- Microsoft Script Editor, 766
- Venkman, 767
- Visual Studio.NET, 766
- Visual Web Developer Express, 766
- Web Inspector, 767
- YUI Compressor, 769
- YUI Doc, 770
- YUI Test, 769

JavaScript/CSS, 640–641

JavaScript/HTML, 639–640

JavaScript, 763

Johnston, Paul, 763

jQuery, 760

JsDoc Toolkit, 770

JSLint, 662–663, 768

JSMin, 768

JSON (JavaScript Object Notation), 581

with Ajax, 583–586

native support, 750–752

overview of, 581–583

security in, 567

JsUnit, 769

K

key codes, 393–394

keyboard events, 392–397

character codes in, 395

on devices, 396–397

introducing, 392–393

key codes in, 393–394

simulating, 428–429

textInput, 395–396

keywords

in ECMAScript 4, 739

in JavaScript, 25–26

King, Andrew B., 653

Konqueror, 238

L

labeled statements, 67

language attribute, 13

last-in-first-out (LIFO) structures, 104

lazy loading, 592–594

Lecomte, Julien, 661

left shift bitwise operator, 49–50

length property, 128

let keyword, 709–710

libraries, JavaScript

Dojo Toolkit, 760

Ext JS, 761

JavaScript MD5, 763

JavaScript, 763

jQuery, 760

Lightbox, 762

MochiKit, 761

moo.fx, 762

MooTools, 760

Prototype, 760

qooxdoo, 762

Rico, 761

script.aculo.us, 762

Yahoo! User Interface (YUI) library, 759

LIFO (last-in-first-out) structures, 104

Lightbox, 762

<link> element, 309–311

listeners, event. See event handlers

LiveConnect, 489–490

Load and Save, DOM Level 3, 518–523

overview of, 518

parsing files from URIs, 521–522

parsing with context, 522

parsing XML in, 518–521

serializing XML and, 523

load events

defined, 368

DOMContentLoaded, 413

overview of, 398–400

localeCompare () method, 140–141

localStorage objects, 632–633

location

of errors, 481

methods, 136–137

location object

manipulating, 217–219

overview of, 216

query string arguments in, 216–217

logging

errors to servers, 487–488

function calls in Firebug, 506–507

messages to consoles, 488–491

messages to pages, 491

logical Boolean operator, 52–53

logical NOT Boolean operator, 51–52

logical OR Boolean operator, 53–54

long-running script limits, 602

lookups, global, 648–649

loops, for-each-in

in ECMAScript 4, 734–735

in EX4, 563

loose coupling

application logic/event handlers,
641–642

CSS/JavaScript, 640–641

HTML/JavaScript, 639–640

LSParser, 518–522

LSSerializer object, 523

M**maintainable code, 635–647**

- conventions for, 636
- defined, 636
- loose coupling and, 639–642
- naming functions and variables in, 637
- overview, 635
- programming, 642
- readability of, 636–637
- variable type transparency in, 637–639

manipulating nodes, 266–268**manipulations methods**

- for reference types, 108–109
- `String` primitive wrapper type, 135–136

Map class, 740–741**mark-and-sweep garbage collection, 91****match () method, 138****matching capturing groups, 118–121****math, in ECMAScript 3.1, 755****Math objects, 146–149**

- `min ()` and `max ()` methods, 146–147
- other methods, 148–149
- properties of, 146
- `random ()` method, 148
- rounding methods, 147

Math.random method, 148**MD4, 763****MD5, 763****media elements, 677–682****member not found error, 494–495****memory and performance**

- collecting garbage in, 93–94
- DOM extensions for, 306
- events and, 422
- introducing, 79

messaging, cross-document, 676–677**methods**

- character, 134
- conversion, 102–103
- date-matting, 112–113
- date/time component, 113–114
- decimal, 755–756
- final classes, 732
- of form-fields, 439
- HTML, 141–142
- inherited, 111–112
- instance, 118–119
- manipulations, 108–109
- `Math` object and, 148–149

- `min ()` and `max ()`, 146–147
- object lock-down, 747–748
- properties, 128–130
- in prototype chaining, 173–174
- queue, 105
- reordering, 106–107
- rounding, 147
- stack, 104
- static object, 742
- string case, 137
- string location, 136–137
- string pattern-matching, 138–140
- string-manipulations, 135–136
- in styles in DOM Levels 2 and 3, 328–329
- superclass, 731–732
- URI-encoding, 142–143

Microsoft Script Debugger, 765**Microsoft Script Editor, 766****Miller, Matthias, 768****MimeType objects, 222–224****min () and max () methods, 146–147****minus operator, 44–45****mobile device identification, 251–253****mobile Safari. See Safari****MochiKit, 761****modifier keys, 387****modulus operator, 56****moo.fx, 762****MooTools**

- general libraries, 760
- `moo.fx` and, 762

Mosaic, 19**mouse events, 383–392**

- accessibility and, 391–392
- button property in, 389–390
- client coordinates in, 384–385
- detail for information about, 391
- elements related to, 387–389
- in Internet Explorer, 391
- mobile Safari support for, 391
- modifier keys for, 387
- mousewheel, 410–412
- screen coordinates in, 386
- simulating, 426–428

Mozilla, 10**Mozilla Foundation, 766****multiline strings, 737****multiple variable declarations, 655–656****multiplicative operators, 54–56****multiply operator, 55**

mutation events, 402–407

- attribute changes, 406
- node insertion, 405
- node removal, 403–405
- simulating, 430
- text changes, 406–407

N

NamedNodeMap type, 321–322

Namespace type, 550–551

namespaces

- best practices for, 645
- in ECMAScript 4 proposals, 732–733
- in EX4, 550–551, 561–563
- in XPath, 534–536

naming functions and variables, 637

NaN (Not a Number) value, 33–34

native JSON support, 750–752

native object prototype patterns, 164–165

navigator object

- handler registration in, 223–224
- overview of, 219–221
- plug-in detection in, 221–223

Netscape Communications Corporation, 617

Netscape Communicator, 234–235, 367

Netscape Navigator

- DOM support in, 9
- ECMAScript compliance of, 5–6
- JavaScript in, 10, 19
- overview of, 2
- user-agent detection in, 234

Nintendo Wii, 253–255

node types, EX4, 555–556

nodes, DOM

- cloneNode() method for, 268
- hierarchy of, 261–262
- inserting, 405
- in Levels 2 and 3, 319–320, 324–325
- manipulating, 266–268
- NodeIterator type, 344–347
- NodeLists, 264–265, 314
- nodeName property, 264
- nodeValue property, 264
- normalizing, 268, 291–292
- relationships of, 264–266
- removing, 403–405
- types of, 263–269

Nokia Nseries, 252–253

Nombas, 1–2

nonfatal errors, 486–487

normalizing nodes, 268, 291–292

<noscript> elements, 21–22

Not a Number (NaN) value, 33–34

NOT bitwise operator, 47

NOT logical operator, 51–52

null data type, 30

nullability, 719

Number types, 31–37

- converting to, 34–37
- floating-point values in, 32–33
- NaN value, 33–34
- overview of, 31–32
- range of values in, 33
- reference, 132–133

O

O(1) operations, 650–651

object () function, 178

object definitions, 97

object internals, 741–742

object literals, 98, 656–657

object lock-down methods, 747–748

Object reference type, 97–99

object-oriented programming (OOP), 151–182

- constructor patterns in, 152–155
- constructor stealing in inheritance, 175–177
- creating objects in, 151–152
- durable constructor patterns in, 169
- dynamic prototype patterns in, 166–167
- factory patterns in, 152
- introducing inheritance, 170
- parasitic combination inheritance, 179–181
- parasitic constructor patterns in, 167–168
- parasitic inheritance, 178–179
- prototypal inheritance, 178
- prototype chaining in inheritance, 170–175
- prototype patterns in. See prototype patterns, in OOP
- qooxdoo and, 762
- summary of, 182

Object.prototype, 171–172

objects

- creating, 744–745
- in DOM Storage, 629–633
- as instances of reference types. See reference types
- Math, 146–149

- native prototype patterns of, 164–165
 - NodeList, 659–660
 - ownership of, 643–644
 - properties of, 743–744
 - toISOString() to Date, 750
 - XHR (XMLHttpRequest). See XHR (XMLHttpRequest) object
 - objects, Global built-in, 142–146**
 - eval () method, 144
 - overview of, 142
 - properties of, 145
 - URI-encoding methods in, 142–143
 - window object accessing, 145–146
 - observer patterns, 606**
 - offset dimensions, 336–337**
 - onclick event handler**
 - bubbling and, 380
 - in memory and performance, 422–423
 - in mouse events, 383–385
 - overview of, 368–373
 - onload event handler, 368–372**
 - OOP (object-oriented programming), 151–182**
 - constructor patterns in, 152–155
 - constructor stealing in inheritance, 175–177
 - creating objects in, 151–152
 - durable constructor patterns in, 169
 - dynamic prototype patterns in, 166–167
 - factory patterns in, 152
 - introducing inheritance, 170
 - parasitic combination inheritance, 179–181
 - parasitic constructor patterns in, 167–168
 - parasitic inheritance, 178–179
 - prototypal inheritance, 178
 - prototype chaining in inheritance, 170–175
 - prototype patterns in. See prototype patterns, in OOP
 - quooxdo and, 762
 - summary of, 182
 - open () method, for XHR object, 569**
 - Opera**
 - custom error messages in, 479
 - dates in, 110
 - debuggers and, 765
 - document.doctype in, 270
 - DOM support in, 9
 - ECMAScript compliance of, 6
 - garbage collection in, 93
 - inheritance in, 112
 - load event in, 520
 - readystatechange event, 416
 - user-agent detection, 238–239
 - XPath in, 530
 - Opera JavaScript debugger, 510–512**
 - breakpoints in, 511
 - Command Line tab in, 512
 - error handling, 510–512
 - error reporting in, 470–472
 - overview of, 510–511
 - stepping through code in, 512
 - operation aborted error, 493–494**
 - operators, 41–63**
 - additive, 56–58
 - assignment, 62–63
 - bitwise, 45–51
 - Boolean, 51–54
 - comma, 63
 - conditional, 62
 - equality, 60–62
 - increment/decrement, 41–43
 - multiplicative, 54–56
 - overloading, 735
 - overview of, 41
 - plus and minus, 44–45
 - relational, 58–60
 - unary, 41
 - optional arguments, 722**
 - OR bitwise operator, 48**
 - OR operator, logical, 53–54**
 - orientationchange, 418–419**
 - outerHTML property, 305–306**
 - outerText property, 305**
 - overloading functions, 76, 123–124**
 - ownerDocument property, 266**
- ## P
- packages**
 - in Dojo Toolkit, 760
 - in ECMAScript 4, 734
 - page dimensions, 207**
 - pageshow and pagehide events, 416–417**
 - parameterized types, 724–725**
 - parasitic constructor patterns, 167–168**
 - parasitic inheritance, 178–181**
 - parent object, 203**
 - parsing**
 - of errors in XML documents, 525–526
 - with EX4, 560–561
 - XML. See parsing XML

parsing XML, 528–529

- with context, 522
- files from URIs, 521–522
- overview of, 518–521

partial text selection, 444–445

passing arguments

- in constructor stealing, 176
- variables in, 82–83

pattern limitations, 122

pattern-matching methods, 138–140

performance best practices, 647–660

- approaches to, 650–655
- array literals in, 656–657
- collecting garbage and, 93
- for DOM interactions, 657–660
- double interpretation and, 654–655
- introducing, 647–648
- iterative values in, 656
- loops, optimizing, 651–652
- loops, unrolling, 652–654
- multiple variable declarations in, 655–656
- object literals in, 656–657
- for property lookups, 650–651
- scopes and, 648–649
- speed considerations in, 655
- statement count in, 655–657

persisted property, 417

Persistent User State – HTTP Cookies, 617

persistent user data, 627

platform identification, 247–249

Playstation 3, 253–255

plug-in detection, 221–223

plus and minus operators, 44–45

pop () method, 104

pop-up windows

- blocking, 210–211
- overview of, 208–210
- security restrictions and, 210

POST requests, 574–575

primitive types. See data types

primitive values, 79–80

primitive wrapper types, 130–142

- Boolean, 131–132
- character method in, 134
- fromCharCode () method in, 141
- HTML methods in, 141–142
- localeCompare () method in, 140–141
- Number, 132–133
- overview of, 130–131

- string case method in, 137
- string location methods in, 136–137
- string pattern-matching methods in, 138–140
- String type generally, 134
- string-manipulations methods in, 135–136

profilers, Firebug, 766

programming maintainable code

- constants, using, 646–647
- globals, avoiding, 644–645
- introducing, 642
- null comparisons, avoiding, 645–646
- ownership of objects, respecting, 643–644

prompt () method, 214

properties

- providing web page information, 271–272
- of form-fields, 437–439
- of functions as reference types, 128–130
- of Global built-in objects, 145
- of Math objects, 146
- of styles in DOM Levels 2 and 3, 328–329

property enumerations, 738

property names, 746

proprietary events, 407–417

- beforeunload, 409
- contextmenu, 407–409
- DOMContentLoaded, 413
- DOMMouseScroll, 411–412
- mousewheel, 410–412
- pageshow and pagehide, 416–417
- readystatechange, 413–416

prototypal inheritance, 178

“Prototypal Inheritance in JavaScript”, 178

prototype chaining, 170–175

- constructor stealing combined with, 176–177
- default prototypes and, 171–172
- instances in, 172
- methods in, 173–174
- overview of, 170–171
- problems with, 174–175
- prototypes in, 172

prototype information, 742–743

Prototype library

- general libraries, 760
- moo.fx and, 762
- script.aculo.us as companion to, 762

prototype patterns, in OOP

- constructor patterns and, 166
- dynamic nature of, 163–164, 166–167
- native object, 164–165
- in operator and, 159–161
- overview of, 155–159
- problems with, 165–166
- syntax for, 162

prototype property, 128**prototypes, class, 727****push () method, 104****Q****QName type, 551–552****qooxdoo, 762****querying**

- in BOM location object, 216–217
- in EX4, 556–557

querySelector() method, 670–671**querySelectorAll() method, 671****queue methods, 105****quirks mode, 19–21, 338–340****R****random () method, 148****range of values, 33****RangeError, 476****ranges, DOM Levels 2 and 3, 349–362**

- cleaning up, 358
- cloning, 358
- collapsing, 356–357
- comparing, 357–358
- complex selection in, 351–353
- inserting content, 355–356
- interacting with content of, 353–355
- Internet Explorer and, 358–362
- overview of, 349–350
- simple selection in, 350–351

readability of code, 636–637**readystatechange event, 413–416****rectangles, 683–685****reductions, array, 716****reference counting, 91–93****reference types**

- Array, 100–102
- built-in objects, 142–148
- conversion methods for, 102–103

Date, 109–111

date-formatting methods, 112–113

date/time component methods, 113–114

functions. See functions, as reference types

Global built-in objects, 142–146

inherited methods and, 111–112

manipulations methods, 108–109

Math objects, 146–149

Object, 97–99

overview of, 97

primitive wrappers as. See primitive wrapper types

queue methods, 105

RegExp type, 115–121

reordering methods for, 106–107

stack methods for, 104

summary of, 149–150

reference values, 79–80**ReferenceError, 476****references, circular, 92****RegExp type, 115–121**

constructor properties of, 120–121

instance methods of, 118–119

instance properties of, 117

overview of, 115–117

pattern limitations in, 122

reference types and, 115–121

regular expressions, 738**relational operators, 58–60****relationships of nodes, 264–266****remainder (modulus) operator, 56****remote scripting, 567****removeChild () method, 267–268****rendering engine identification, 240–244****rendering modes, 297****reordering methods, 106–107****repeating timers, 600–602****replace () method, 138–139****replaceChild () method, 267–268****reserved words, 26****Resig, John, 760****resize event, 401–402****resizing functionality, 606****resolving namespace functions, 534–535****rest arguments, 721–722****retrieving selected text, 443****return statements, 475****reverse () method, 106**

Rhino JavaScript, 663–664

rich text editor, 458–464

- forms and, 463–464
- interacting with, 459–461
- overview of, 458–459
- selections with, 462–463

Rico, 761

right shift operators, 50–51

rounding methods, 147

Ruby on Rails, 760

Russel, Alex, 760

S

Safari

- clipboards and, 449
- custom error messages in, 478
- debugger. See Drosera debugger
- debuggers and, 765
- debuggers for, 767
- `document.doctype` in, 270
- DOM support in, 9
- ECMAScript compliance of, 6
- error reporting in, 469–470
- events in, generally, 417–421
- gesture events in, 420–421
- inheritance in, 112
- introducing, 417
- mouse events in, 391
- `orientationchange` in, 418–419
- touch events in, 419–420

`sayName ()` method, 154–158, 167, 169

scope, global

- in BOM window object, 201–202
- in window object, 145–146

scope chains, 84–90. See also block-level scopes

scopes

- augmentation of, 87–88
- for best performance, 648
- block-level scopes and, 88
- in execution context, 85–87
- execution context and, 84–87
- global lookups, avoiding, 648–649
- identifier lookups in, 89–90
- `let` keyword and, 709–710
- with statements, avoiding, 649
- variable declarations in, 88–89

scope-safe constructors, 589–592

screen coordinates, 386

screen object, 224–226

screen position properties, 205

Script Debugger, Microsoft, 765

Script Editor, Microsoft, 766

<script> elements, 13–19

- deferred scripts, 16–17
- deprecated syntax, 18–19
- in dynamic scripts, 307–309
- inline code vs. external files, 19
- overview of, 13–15, 22
- tag placements, 15–16
- in XHTML, 17–18

script.aculo.us, 762

scripting forms, 433–464

- events of form-fields, 440–441
- fields of, 436–437
- methods of form-fields, 439
- overview of, 433–434
- properties of form-fields, 437–439
- resetting, 435–436
- rich text editing in, 458–464
- select boxes and, 450–456
- serializing forms, 456–458
- submitting, 434–435

scripting text boxes, 441–450

- characters, blocking, 446–447
- clipboard and, 447–449
- input filtering in, 445–449
- partial text selection, 444–445
- retrieving selected text, 443
- `select` event for, 442–443
- tab forward, automatic, 449–450
- text selection, 442

scroll dimensions, 339–340

scroll event, 402

scrolling with extensions to `HTMLElement` type, 298

`search ()` method, 138

secure execution environments

- ADsafe, 771
- Caja, 771

security

- of cookies, 627
- in JSON, 567
- pop-up windows and, 210

select boxes, scripting, 450–451

- adding options, 454
- overview of, 450–452
- removing options, 455–456
- selecting options, 452–454

select event, 442–443

- selection, complex**
 - in DOM Levels 2 and 3, 351–353
 - in Internet Explorer, 360
- selection, simple**
 - in Internet Explorer ranges, 359
 - in ranges in DOM Levels 2 and 3, 350–351
- selection, text, 442**
- Selection object, 462–463**
- Selectors API (application program interface)**
 - future of, 672
 - introducing, 669–670
 - `querySelector()` method in, 670–671
 - `querySelectorAll()` method in, 671
- send () method, 569–570**
- serializing**
 - with EX4, 560–561
 - forms, 456–458
 - XML, 523, 526
- sessionStorage objects, 629–630**
- setAttribute () type, 284**
- setDate () method, 113**
- setters, 704–705**
- setUserData () method, 325**
- SHA-1, 763**
- shadowing properties, 157–158**
- ShrinkSafe, Dojo, 769**
- signed right shift bitwise operator, 50**
- simple selection**
 - in Internet Explorer ranges, 359
 - in ranges in DOM Levels 2 and 3, 350–351
- simple XSLT transformations, 539**
- simulating events, 425–431**
 - DOM, 426
 - HTML, 430
 - Internet Explorer, 430–431
 - keyboard, 428–429
 - mouse, 426–428
 - mutation, 430
- slice () method, 108**
- Slocum, Jack, 761**
- smartphones, 253**
- snapshot result type, 532**
- sort () method, 106–107**
- specialty libraries**
 - animation and effects, 762
 - cryptography, 763
 - Internet applications, 762
 - types of libraries, 759
- Speed Up Your Site, 653**
- SpiderMonkey, 768**
- splice () method, 109**
- split () method, 140**
- splitText () method, 292**
- splitting text nodes, 292**
- spoofing, 232**
- stack methods, 104**
- standards mode, 19–21**
- statement counts**
 - array literals for, 656–657
 - introducing, 655
 - iterative values for, 656
 - multiple variable declarations in, 655–656
 - object literals for, 656–657
- statements, 63–72**
 - break, 67–69
 - continue, 67–69
 - do-while, 64
 - for, 65–66
 - for-in, 66–67
 - if, 63–64
 - labeled, 67
 - overview of, 63
 - switch, 70–72
 - syntax, 25
 - this, 736
 - while, 65
 - with, 69–70
- static members, 729**
- static object methods, 742**
- Stephenson, Sam, 760**
- stopPropagation () method, 376, 378–382**
- storage, client-side, 617–634**
 - cookies for. See cookies
 - globalStorage objects in, 631–632
 - Internet Explorer user data in, 627–628
 - introducing, 617
 - limits and restrictions on, 634
 - localStorage objects in, 632–633
 - overview of, 628
 - sessionStorage objects in, 629–630
 - storage events in, 633
 - Storage type in, 628–629
 - StorageItem type in, 633
 - summary of, 632
- storage, database, 694–696**
- strict mode, 736**

String data types, 37–41

- character literals, 38
- converting to, 39–40
- nature of, 39
- overview of, 37

string generics, 708

string indexing, 737–738

string type, 134–142

- case methods, 137
- character methods, 134
- `fromCharCode ()` method, 141
- HTML methods, 141–142
- `localeCompare ()` method, 140–141
- location methods, 136–137
- manipulations methods, 135–136
- overview of, 134
- pattern-matching methods, 138–140

String.prototype, 164

strings in variable typing, 717–719

<style> element

- access to elements and, 326–327
- computed styles and, 329–331
- in dynamic styles, 309–311
- overview of, 326
- properties and methods of, 326–329
- style sheets and, 331–333

styles, DOM Levels 2 and 3

- accessing, 326–327
- computed, 329–331
- creating rules for, 334–335
- CSS rules for, 333–334
- deleting rules from, 326–335
- element dimensions and, 336–342
- properties and methods of, 328–329
- working with, 331

subcookies, 622–626

substrings

- creating values from, 135–136
- replacing, 138

subtract operator, 57–58

SubType prototype, 171–177, 179–181

super standards mode, 20–21

superclass methods, 731–732

SuperType prototype, 171–177, 179–181

switch statement, 70–72

synchronous file loading, 527

synchronous XHR requests, 569–570

syntax, 23–25

- case sensitivity, 23
- comments, 24

errors in Internet Explorer, 495–496

identifiers, 24

for prototype patterns in OOP, 162

statements, 25

tools for checking, 768

in XML attributes, 553

SyntaxError, 476

Syracuse University, 648

system cannot locate the resource specified error, 496

system dialogs, 213–215

T

tab forward, automatic, 449–450

tables, manipulating, 311–313

tag placements, 15–16, 22

tagName property, 279

TDD (test-driven development), 769

techniques, advanced, 589–616

custom events, 606–609

drag and drop, 609–615

functions. See functions, advanced

introducing, 589

summary of, 615–616

timers. See timers

techniques, debugging, 488

error handling, 488–492

introducing, 488

logging messages to consoles, 488–491

logging messages to pages, 491

throwing errors, 491–492

test () method, 119

test-driven development (TDD), 769

text, drawing, 687–688

text boxes, scripting, 441–450

characters, blocking, 446–447

clipboard and, 447–449

input filtering in, 445–449

partial text selection, 444–445

retrieving selected text, 443

`select` event for, 442–443

tab forward, automatic, 449–450

text selection, 442

text changes event, 406–407

text ranges, in Internet Explorer, 358–362

cloning, 362

collapsing, 361–362

complex selection in, 360

content of, interacting with, 360–361

- introducing, 358–359
 - simple selection in, 359
 - text selection, 442**
 - Text type, 289–292**
 - creating text nodes in, 290–291
 - normalizing text nodes in, 291–292
 - overview of, 289–290
 - splitting text nodes in, 292
 - textInput event, 395–396**
 - this object, 126–127**
 - this statements, 736**
 - thread-safe XML DOM documents, 540–541**
 - throttle () function, 604–605**
 - throw operator, 477–479**
 - throwing errors, 491–492**
 - timeouts, 211–213**
 - timers, 598–606**
 - function throttling in, 604–606
 - overview of, 598–600
 - repeating, 600–602
 - yielding processes in, 602–604
 - title property, 271**
 - toDateString () method, 112**
 - toExponential () method, 133**
 - toFixed () method, 132–133**
 - toISOString() to Date objects, 750**
 - toLocaleString () method, 111–112**
 - tools, JavaScript**
 - ADsafe, 771
 - AjaxDoc, 770
 - Aptana Debugger, 767
 - Caja, 771
 - Dojo Object Harness (DOH), 770
 - Dojo ShrinkSafe, 769
 - Drosera, 767
 - Firebug, 766
 - JavaScript Lint, 768
 - JsDoc Toolkit, 770
 - JSLint, 768
 - JSMIn, 768
 - JsUnit, 769
 - Microsoft Script Debugger, 765
 - Microsoft Script Editor, 766
 - Venkman, 767
 - Visual Studio.NET, 766
 - Visual Web Developer Express, 766
 - Web Inspector, 767
 - YUI Compressor, 769
 - YUI Doc, 770
 - YUI Test, 769
 - toPrecision () method, 133**
 - toString () method, 111–112**
 - toTimeString () method, 112**
 - touch events, 419–420**
 - transformations**
 - in <canvas> element, 688–691
 - complex XSLT, 539–543
 - simple XSLT, 539
 - XSLTProcessor type and, 543–544
 - transformNode () method, 539–541**
 - traversals, 342–344**
 - TreeWalker type, 347–349**
 - trim() method, 749–750**
 - true Boolean value, 30–31**
 - try-catch statement**
 - for error handling, 474–477
 - throwing errors and, 477–479
 - throwing errors vs., 479
 - two's complement, 46**
 - type attribute, 13–15**
 - type coercion, 60, 481–482**
 - type comments, 638–639**
 - type definition, 723–726**
 - type detection, 719–720**
 - TypeError, 476**
 - typeof operator**
 - as data type, 28
 - in EX4, 564
 - for primitive and reference types, 84
 - types**
 - determining, 84
 - in ECMAScript 4. See types, ECMAScript 4
 - in EX4, 547–552
 - of nodes in DOM, 263–269
 - types, ECMAScript 4**
 - assigning, 725–726
 - converting, 725–726
 - deep, 723
 - function, 724
 - parameterized, 724–725
 - union, 723
- ## U
- UI events, 383**
 - unary operators, 41–45**
 - increment/decrement, 41–43
 - plus and minus, 44–45
 - undefined data type, 28–30**
 - union type, 723**

unit testers

Dojo Object Harness (DOH), 770

JsUnit, 769

YUI Test, 769

units of measurement, 327

Universal Time Code (UTC), 109–111

unknown runtime, 495

unload event, 401

unsigned integers, 46

unsigned right shift bitwise operator, 51

URI-encoding methods, 142–143

URLs

security and, 577

in XHR requests, 569–570

usage subsets, 756–757

user-agent detection, 232–259

browser identification, 245–247

Chrome and, 238

complete script for, 255

in early browsers, 233–234

game system identification, 253–255

Gecko and, 235–237

history of, 233

Internet Explorer 3 and, 234

Internet Explorer 4-8 and, 234–235

introducing, 232

Konqueror and, 238

mobile device identification, 251–253

Netscape Communicator 4 and,
234–235

Netscape Navigator 3 and, 234

Opera and, 238–239

platform identification, 247–249

rendering engine identification,
240–244, 246

using for, 258

WebKit and, 237

Windows operating system identification,
249–251

working with, 240

UTC (Universal Time Code), 109–111

V

validation

JavaScript Lint, 768

JSLint, 768

overview of, 662–663

values

in Boolean type, 30–31

in conformance detection, 276

converting to numbers, 34–37

converting to strings, 39–40

copying, 81–82

dereferencing, 93–94

floating-point, 32–33

functions as, 125–126

iterative, 656

NaN, 33–34

in Null type, 30

in Number type, 31–32

operators manipulating. *See* operators

primitive, 79–81

range of, 33

reference, 79–80

variable declarations, 88–89

variable objects, 84–85

variable type transparency, 637–639

variable typing, ECMAScript 4, 717–720

introducing, 717

nullability in, 719

strings, numbers, and Booleans,
717–719

type detection and, 719–720

variables, 79–95

argument passing of, 82–83

block-level scopes as, 88–90

dynamic properties of, 80

in ECMAScript 4. *See* variable typing,
ECMAScript 4

execution context and scope of, 84–87

garage collection and, 90

global, 27

initializing, 30, 89

in language basics, 26–27

naming, 637

overview of, 79

scope chain augmentation and, 87–88

summary of, 94–95

type of, determining, 84

values, copying, 81–82

values in, primitive and reference,
79–80

Vector class, 740

Venkman, 513

as debugging tool, 767

virtual properties, 727–728

Visual Studio, 513

Visual Studio.NET, 766

Visual Web Developer Express, 766

W

W3C Access Control for Cross-Site Requests, 580

walkers, 347–349

watches

in Firebug, 505

in Internet Explorer debugger, 500–501

web browsers. See browsers

Web Hypertext Application Technical Working Group (WHAT-WG), 628, 672

Web Inspector, 472

as debugging tool, 767

Webkit, 237, 507

WebSocket type, 700–701

what you see is what you get (WYSIWYG) editing, 458

WHAT-WG (Web Hypertext Application Technical Working Group), 628, 672

while statement, 65

Wii, 253–255

window object, 201–215

blocking pop-up windows, 210–211

frames in, 202–205

Global scope in, 201–202

intervals in, 211–213

navigating, 207–211

opening, 207–211

popping up windows in, 208–210

position of, 205–206

relationships of, 202–205

security restrictions for, 210

size in, 206–207

system dialogs with, 213–215

timeouts in, 211–213

Windows Mobile, 253

Windows operating system identification, 249–251

wire weight, 665

with statements

avoiding, 649

in ECMAScript 4, 735

overview of, 69–70

wrapper types, primitive, 130–142

Boolean, 131–132

character method in, 134

fromCharCode () method in, 141

HTML methods in, 141–142

localeCompare () method in, 140–141

Number, 132–133

overview of, 130–131

string case method in, 137

string location methods in, 136–137

string pattern-matching methods in, 138–140

String type generally, 134

string-manipulations methods in, 135–136

write () methods, 277–278

WYSIWYG (what you see is what you get) editing, 458

X

XDR (XDomainRequest) type, 578–580

XHR (XMLHttpRequest) object, 568–578

browser differences and, 575–578

Firebug for inspecting, 766

GET requests, 573–574

HTTP headers, 571–573

overview of, 568–569

POST requests, 574–575

usage of, 569–571

XHTML (Extensible HyperText Markup Language), 17–18

XLST support in browsers, 539–546

XML, in Internet Explorer (IE), 523–527

loading XML files in, 526–527

overview of, 523–526

serializing XML in, 526

XML, in JavaScript, 515–546

in XMLSerializer type, 517–518

browser support for generally, 515

cross-browser processing, 528–529

in DOM Level 2, 515–516

in DOM Level 3 Load and Save, 518–523

in DomParser type, 516–517

ECMAScript for XML vs.. See ECMAScript for XML (EX4)

in Internet Explorer, 523–527

introducing, 515

summary of, 546

XLST support in browsers with, 539–546

XPath support in browsers for, 530–538

XML construction and manipulation, 558–560

XML DOM support, in browsers, 515–530

- cross-browser processing, 528–529
- for `DomParser` type, 516–517
- in Internet Explorer, 523–527
- for Level 2, 515–516
- for Level 3 Load and Save, 518–523
- for `XMLSerializer` type, 517–518

XML namespaces, 318–322

XML type, 547–548

XMLList type, 549–550

XMLSerializer type, 517–518

XOR bitwise operator, 49

XPath, 530–538

- cross-browser, 536–538
- DOM Level 3, 530–533
- in Internet Explorer, 535–536
- namespace support in, 534–536

XSL template object, 541

XSLT, 539–546

- cross-browser, 545–546

- in Internet Explorer, 539–543
- processor, 543–545

XSLTProcessor type, 543–545

- parameters with, 544–545
- resetting, 545
- XSLT, 543–545

XSS (cross-site scripting) attacks, 578

Y

Yahoo! Mail, 486

Yahoo! Toolbar, 210

Yahoo! User Interface (YUI) library, 181, 644

- Ext JS as extension to, 761
- as general library, 759
- YUI Compressor, 664–665, 769
- YUI Doc, 770
- YUI Test, 769

yielding processes, 602–604