

Index

SYMBOLS AND NUMERICS

& (ampersand)

OpenOffice.org Writer, 292–294
Visual Basic .NET, 505

' (apostrophe), 205–209

* (asterisk)

cardinality operators, 62–64
findstr utility, 310–311
OpenOffice.org Writer, 285–286
PowerGREP, 333–335
specificity, 258
VBScript, 474
Word, 256–258, 423–424

@ (at sign), 258–260

\ (backslash). See also listings under letter following backslash

debugging, 251
described, 102–103

\> (backslash, greater-than sign), 166–167

^ (caret)

\$ (dollar sign) metacharacter, 153–157
findstr utility, 315
first line of file, examining only, 146–148
line or string, matching immediately after beginning, 144–146
literal character, matching, 133–135
metacharacters, 133–135
MySQL, 404–406
negated character classes, 376–379
part numbers, 153–155
positional metacharacters, 144–146
within square brackets [], 133–135
VBScript, 475–478
Visual Basic .NET, 505

) (closed parens), 505

, (comma)

four-digit number, adding, 216–220
names, reversing order and adding, 467

{ } (curly braces)

{0,m}, 67–69
{n}, 66
{n,}, 70–71
{n,m}, 67, 69–70, 285–286
Word, 260–262

\$ (dollar sign)

with caret (^) metacharacter, 153–157
described, 149–150
findstr utility, 315
in multiline mode, 150–152
MySQL, 404–406
part numbers, matching, 153–155
positional metacharacters, 149–157
PowerGREP, 339–340
VBScript, 475–478

/ (forward-slash), 433–436

// (forward slash, double), 568

> (greater-than symbol), 166–167

- (hyphen), 228–230

< (less-than symbol), 164–166

(number sign)

Access date/time data, 425
Word database management system, 424

?# (open parens, question mark, number sign), 505

() (parentheses)

alternation, 177–185
back references, 190–193
C#, 519
capturing, 185–188, 499–502
characters, choice between two, 108
GroupCollection and Group classes, Visual Basic .NET, 497–499
grouping, 171–177, 482–483
JavaScript and JScript, 445–448
literal, matching, 175
named groups, 187–188
nested, numbering, 186–187
non-capturing, 188–189
numbering groups, 185–186
PowerGREP, 335–339

() (parentheses) (continued)

() (parentheses) (continued)

- VBScript, 482–483
- Visual Basic.NET, 499–502
- (?) (parentheses-enclosed question mark), 196–197
- % (percent sign)
 - MySQL relational database, 397–399
 - SQL Server 2000, 366–372
- . (period)
 - described, 75–78
 - escaping, 102
 - inventory, matching, 79–80
 - PowerGREP, 332–333
- | (pipe symbol), 690–691
- + (plus sign)
 - cardinality operators, 64–66
 - in OpenOffice.org Writer, 285–286
 - PowerGREP, 333–335
 - VBScript, 474
 - Word, 256, 258–260
- ? (question mark)
 - OpenOffice.org Writer, 285–286
 - PowerGREP, 333–335
 - VBScript, 474
 - Word, 256, 258, 422–423
- (?) (question mark enclosed in parentheses), 196–197
- [] (square brackets)
 - literal, 113–114
 - metacharacter, 133–135
- _ (underscore)
 - MySQL relational database, 397–399
 - SQL Server 2000, 372–373
- _(underscore), matching characters other than, 82–83, 332–333
- 0 (zero), 518
- 0 through 9 (zero through nine), 83–88
- 1 (one), 518

A

/a **command-line switch**, 318–319

abbreviations

- data problems, 246
- sensitivity and specificity, 234

Access (Microsoft) database management system

- asterisk (*)metacharacter, 423–424
- character classes, 426–428
- date/time data, 425
- described, 413
- hard-wired query, 414–419
- interface, 413–414
- metacharacters listed, 422
- numeric digit, 424
- parameter query, 419–421
- question mark (?)metacharacter, 422–423
- after characters. *See* positional metacharacters
- alert boxes, displaying matches in separate, 441–443, 447–448

[:a1num:]

- OpenOffice.org Writer, 140–141
- PHP, 583–584

alphabet, ASCII

- described, 81–82
- W3C XML Schema, 615

alphabet, non-ASCII

- described, 82–83
- PowerGREP, 332–333

alphabetic character, matching

- described, 75–78
- escaping, 102
- inventory, matching, 79–80
- PowerGREP, 332–333

alphabetic order, reverse, 128–129

alphabetic ranges, character class, 115–117

alternation

- described, 177–178
- multiple options, 180–182
- OpenOffice.org Writer, 289–292
- Perl, 690–691
- PowerGREP, 339
- two literals, 178–179
- unexpected behavior, 182–185
- W3C XML Schema, 615

ampersand (&)

- OpenOffice.org Writer, 292–294
- Visual Basic .NET, 505

analytical approach

- appropriateness, 35
- data source and contents, considering, 33–34
- described, 31–32
- documenting, 35–38
- expressing in English, 32–33
- options, considering, 34
- sensitivity and specificity, 34–35

anchor

- described, 143
- end of line or file, immediately before, 149–157
- first line of file, examining only, 146–148
- IP address, 161–163
- line or string, immediately after beginning
 - (^ metacharacter), 144–146
- MySQL, 404–406
- PCRE, 586
- Perl, 686–687
- PowerGREP, 339–340
- sensitivity and specificity, 231
- VBScript, 475–478, 476, 478
- W3C XML Schema, 613–614
- word boundaries, 164–169

apostrophe ('), 205–209

appropriateness, 35

argument, returning previous, 638

array, PCRE, 576–578

ASCII alphabet, matching

- described, 81–82
- W3C XML Schema, 615

ASCII alphabet, matching characters other than

- described, 82–83
- PowerGREP, 332–333

assignment statement, 438

asterisk (*)

cardinality operators, 62–64
 findstr utility, 310–311
 OpenOffice.org Writer, 285–286
 PowerGREP, 333–335
 specificity, 258
 VBScript, 474
 Word, 256, 257, 258, 423–424

at sign (@), 258–260

atomic zero-width assertions. See positional metacharacters

B

\B metacharacter, 168

\b metacharacter, 112–113

\b quantifiers with character classes, 112–113

/b switch, findstr utility, 315

back references

described, 5, 195
 detecting, 190–193
 OpenOffice.org Writer, 292–294
 parentheses, 190–193
 Perl, 689–690
 PowerGREP, 335–339
 Visual C#.NET, 545–547
 Word, 265, 275–278

background color, 318–319

backslash (\). See also listings under letter following backslash

debugging, 251
 described, 102–103

backslash, greater-than sign (\>), 166–167**base 16 numbers, 119–120****Basic Latin character block, 652**

before characters. See positional metacharacters

beginning boundary, word, 164–166

beginning-of-line position, findstr utility, 315–316

between characters. See positional metacharacters

blank lines, matching, 155–157**blocks, character**

Java, 652–653
 Unicode, 608–612

Boolean value

C# string argument, 520–521
 exec() method of RegExp object, 441–443
 Execute() method, VBScript, 470–471
 JavaScript and JScript, 441–443
 positional metacharacters, VBScript, 478
 Test() method, VBScript RegExp object, 464–465
 VBScript Global property, 458–462

boundary, word

beginning, identifying (<), 164–166
 end, identifying (>), 166–167
 positions, findstr utility, 313–315
 PowerGREP, 340–342
 uppercase letter, beginning or end, 168
 VBScript, 479

browser forms validation. See JavaScript; JScript

button click function, 534–535**C****C#**

character sequences, replacing (Replace() method), 526–528
 classes listed, 517
 CompileToAssembly() method, 519
 console application example, 512–516
 described, 511
 GetGroupNames() method, 519
 GetGroupNumbers() method, 519
 GroupNameFromNumber() method, 519
 GroupNumberFromName() method, 519
 groups in a match (GroupCollection object), 536–538
 groups in collection (Group object), 536–538
 inline comments (IgnorePatternWhitespace option), 539–541
 IsMatch() method, 520–521
 Match class, 532–533
 Match() method, 521
 Matches() method, 522–526
 NextMatch() method, 533–536
 Options property of Regex class, 518
 overload for Replace() method, 532
 overload for Split() method, 532
 Regex class methods listed, 518–519
 Regex class properties listed, 517
 RegexOptions class listed, 539
 regular expressions support described, 512
 RightToLeft property of Regex class, 518
 static methods of Regex class, 531–532
 string, splitting Split() method, 528–531
 tools, 30

callback function, PCRE, 580**Canadian postal code**

pattern matching, 85–87
 problem definition, 85
 uppercase alphabetic characters, matching only, 87–88

CaptureCollection and Capture classes, Visual Basic .NET, 499–502

captured groups, Perl, 687–689**cardinality operators**

asterisk (*) quantifier, 62–64
 plus sign (+) quantifier, 64–66

caret (^)

dollar sign (\$) metacharacter, 153–157
 findstr utility, 315
 first line of file, examining only, 146–148
 line or string, matching immediately after beginning, 144–146
 literal character, matching, 133–135
 metacharacters, 133–135
 MySQL, 404–406
 negated character classes, 376–379
 part numbers, 153–155
 positional metacharacters, 144–146
 within square brackets [], 133–135
 VBScript, 475–478
 Visual Basic .NET, 505

case sensitivity

described, 6
 matching, 15

case sensitivity (continued)

- metacharacters, 16
- strings, splitting, 564–566

case-insensitive matching

- Java, 629
- modifiers, 104
- Perl, 669–674
- PHP, 559–564
- RegexOptions enumeration, 502–505
- strings, splitting, 566–567
- VBScript, 462–464
- Word, 262–265

character blocks

- Java, 652–653
- Unicode, 608–612

character classes

- Access, 426–428
- choice between two characters, 108–111
- collections, widely used, 105
- described, 105–108
- findstr utility, 311–313, 320
- HTML heading elements, finding, 132–133
- Java, 647–651
- metacharacters within, 133–136
- MySQL, 406–408
- negated, matching, 136–139
- OpenOffice.org Writer, 286–289
- PCRE, 587–589
- Perl, 692–696
- POSIX, 139–141, 582–585
- quantifiers, using with, 111–115
- SQL Server 2000, 373–376
- Unicode, 605–606
- VBScript, 478
- Word, 265, 268

character classes, range

- alphabetic, 115–117
- date separators, differing, 129–132
- described, 114–115
- digit, 117–119
- hexadecimal numbers, 119–120
- IP addresses, 120–127
- Java, 647
- negated, 378
- reverse, 128–129
- Word examples, 268

character sequences

- C#, 526–528
- different, 54–56
- followed by other sequence of characters, 199–202
- not followed by another sequence of characters, 202–203
- not preceded by another sequence of characters, 213–214
- preceded by another sequence of characters, 209–213
- replacing Star with Moon in example, 237–240
- in string, matching all, PCRE, 574–576

characters

- differ among contexts, 13–15
- documentation, 37
- grouping, parentheses, 172–173
- Java, 635–638, 642–644
- pattern class, Java, 632
- positions versus, 74–75

- preceding, 258–260
- tab, matching (`\t` metacharacter), 98–99

characters, position relative to. See positional metacharacters

classes. See also character classes

- C#, 517
- Visual Basic .NET, 490

client-side replace functions, 455

client-side validation, forms data. See JavaScript; JScript

closed parens (), 505

collections, 105

color values, 119–120

column, beginning, 404–406

comma (,)

- four-digit number, adding, 216–220
- names, reversing order and adding, 467

command-line switches

- /a, 318–319
- /v, 316–318

comments

- described, 243
- pattern class, Java, 630–632
- Visual Basic .NET, 505–507

compile() method, Java, 633

CompileToAssembly() method, C#, 519

concatenation character, 505

console application example, C#, 512–516

CONTAINS predicate, SQL Server 2000, 386–390

contents, analyzing, 33–34

contexts, characters in different, 13–15

counting

- groups, 639
- matches, 538

curly braces ({})

- {0, m}, 67–69
- {n}, 66
- {n, }, 70–71
- {n, m}, 67, 69–70, 285–286
- Word, 260–262

D

\D metacharacter

- alternative, less succinct, 90–92
- described, 83, 89–90

\d metacharacter

- alternative, less succinct, 90–92
- Java, 645–647
- PowerGREP, 332–333
- W3C XML Schema, 614–615

data

- debugging, 246–247
- sensitivity and specificity, 233–236
- source and contents, considering, 33–34
- types in W3C XML Schema, 599–601

data validation. See JavaScript; JScript

database program. See MySQL relational database

date

- Access # metacharacter, 425
- formats, changing, 6
- PHP splitting, 566

- search-and-replace examples, 273–275
 - separators, 129–132
 - DATE columns, MySQL database, 399**
 - debugging**
 - backslashes, 251
 - data problems, 246–247
 - described, 241
 - interactions and, 251
 - test cases, creating, 247–248
 - whitespace, 248–251
 - decimal numbers, Unicode, 606–607**
 - delimiters, Perl, 675–676**
 - derivation, 602–603**
 - digit**
 - OpenOffice.org Writer, 302–304
 - ranges, character class, 117–119
 - directory listings, manipulating**
 - described, 7–8
 - VBScript, 455
 - Document Type Definitions (DTDs), 593–598**
 - documenting**
 - characters, 37
 - comments, adding to code, 243
 - described, 241–242
 - in English, 32–33
 - expected outcome, 36–37
 - extended mode, 243–245
 - inline, Visual Basic.NET, 506–507
 - JavaScript and JScript, 452
 - PHP, 589–590
 - problem definition, 242–243
 - undesired text, 37
 - Visual Basic .NET (Microsoft), 505–507
 - when to use, 35–36
 - whitespace, 37–38
 - documents**
 - positive lookahead, 203–205
 - SQL Server 2000 filters, 391
 - dollar currency, matching, 158–161**
 - dollar sign (\$)**
 - with caret (^) metacharacter, 153–157
 - described, 149–150
 - `findstr` utility, 315
 - in multiline mode, 150–152
 - MySQL, 404–406
 - part numbers, matching, 153–155
 - positional metacharacters, 149–157
 - PowerGREP, 339–340
 - VBScript, 475–478
 - dot. See . (period)**
 - DOTALL mode, Java, 632**
 - double character matching, 47–49**
 - doubled references, finding and removing. See back references**
 - described, 5, 195
 - detecting, 190–193
 - OpenOffice.org Writer, 292–294
 - parentheses, 190–193
 - Perl, 689–690
 - PowerGREP, 335–339
 - Visual C#.NET, 545–547
 - Word, 265, 275–278
 - downloading**
 - MySQL relational database, 393–394
 - XML editors, 593
 - DTDs (Document Type Definitions), 593–598**
- ## E
- /e switch, findstr utility, 315**
 - echo statement, 576**
 - editors, XML, 592**
 - email addresses, 224–228**
 - end boundary, word, 166–167**
 - end-of-line position**
 - described, 149–150, 315–316
 - `findstr` utility, 315
 - in multiline mode, 150–152
 - MySQL, 404–406
 - part numbers, matching, 153–155
 - PowerGREP, 339–340
 - VBScript, 475–478
 - end-of-string position. See \$ (dollar sign)**
 - English alphabet characters, matching**
 - described, 81–82
 - W3C XML Schema, 615
 - English, documenting in, 32–33**
 - enumeration**
 - Visual Basic .NET, 502–505
 - W3C XML Schema, 602–603
 - errors, finding**
 - backslashes, 251
 - data problems, 246–247
 - described, 241
 - interactions and, 251
 - test cases, creating, 247–248
 - whitespace, 248–251
 - escaping characters/sequences**
 - backslash (\), finding, 102–103
 - dollar amounts, finding, 158–161
 - Java, 653–654
 - pattern delimiters, PCRE, 570
 - PCRE, 579
 - period (.) metacharacter, 102
 - Perl, 701–702
 - W3C XML Schema, 616
 - Word wildcards, 359
 - Excel (Microsoft) wildcards**
 - in data forms, 360–362
 - described, 28–29, 351
 - escaping, 359
 - in filters, 362–363
 - Find interface, 351–355
 - listed, 355–358
 - excluding characters, 133–135**
 - Execute() method, VBScript, 467–471**
 - expected outcome, documenting, 36–37**
 - extended mode, 243–245**
 - eXtensible HyperText Markup Language (XHTML)**
 - color values, matching hexadecimal number ranges, 119–120
 - optional whitespaces, matching, 96–98

eXtensible Markup Language (XML)

eXtensible Markup Language (XML)

- instance document, creating, 592, 594, 595–598
 - optional whitespaces, matching, 96–98
 - Web forms validation, W3C specification, 429
 - whitespace and non-whitespace metacharacters, 92–93
- extreme sensitivity, awful specificity, 222–223**

F

- /f switch, 322–323**
- false
- C# string argument, 520–521
- exec() method of RegExp object, 441–443
- Execute() method, VBScript, 470–471
- Global property, VBScript, 458–462
- JavaScript and JScript, 441–443
- positional metacharacters, VBScript, 478
- Test() method, VBScript RegExp object, 464–465

fields, Java pattern class, 629

file access, VBScript, 455

File Finder tab, PowerGREP, 329–330

filename searches

- non-wildcard, 322–323
- wildcard, 319–322

filters, Word wildcards, 362–363

Find All button, OpenOffice.org Writer, 281

Find interface, Word wildcards, 351–355

findstr utility

- beginning- and end-of-line positions, 315–316
- character classes, 311–313
- command-line switches, 316–319
- described, 22–23, 305–306
- filelist example, 322–323
- literal text, 306–308
- metacharacters, 308–309
- multiple file examples, 321–323
- quantifiers, 310–311
- single file examples, 319–321
- word-boundary positions, 313–315

Firefox (Mozilla)

- forward-slash syntax, RegExp object instance, 436–437
- JavaScript enabling, 430
- regular expressions support, 430

first character, position before

- dollar sign (\$) metacharacter, 153–157
- findstr utility, 315
- first line of file, examining only, 146–148
- line or string, matching immediately after beginning, 144–146
- literal character, matching, 133–135
- metacharacters, 133–135
- MySQL, 404–406
- negated character classes, 376–379
- part numbers, 153–155
- positional metacharacters, 144–146
- within square brackets [], 133–135
- VBScript, 475–478
- Visual Basic .NET, 505

first character, position of, 644

first line of file, examining only, 146–148

first matching character sequence, 644

first name, swapping with last, 467

flags() method, Java pattern class, 633

folder, finding with PowerGREP, 329–330

for loop, if statement with nested, 576

foreach loop, 538

foreign languages

- character sensitivity and specificity, 234–235
- right to left matching, Visual Basic.NET, 507
- RightToLeft property of Regex class, 518

forms validation. *See also* JavaScript; JScript

- described, 5
- Word wildcards, 360–362
- XML, 429

forward-slash (/), 433–436. *See also* listings under letter following forward slash

forward slash, double (//), 568

four-digit number, comma separating, 216–220

frequently run queries, 414–419

full-text search, SQL Server 2000

- CONTAINS predicate, 386–390
- described, 379
- index, enabling and creating, 380–385

G

/g switch, 322–323

global matching

- JavaScript and JScript, 441–443, 445–448
- modifiers, 103
- strings, replacing, 679–681
- VBScript, 458–462, 470–471

greater-than symbol (>), 166–167

greedy matching, Microsoft Word, 265, 268

grouping parentheses

- characters, 172–173
- described, 171–172
- quantifiers and, 173–175
- U.S. telephone numbers, 175–177
- VBScript, 482–483

groups

- C#, 519, 536–538
- captured, 499–502, 687–689
- in collection (Group object), 536–538
- counting, 639
- number, getting in C#, 519
- PCRE, 572–574
- Perl, 687–689
- PowerGREP, 335–339
- preceding, 258–260
- Visual Basic.NET, 497–499, 499–502
- Visual C#.NET, 544–545

H

hard-wired query, 414–419

hexadecimal colors, 318–319

HTML (HyperText Markup Language)

- color values, matching hexadecimal number ranges, 119–120
- heading elements, finding, 132–133
- IP address style, amending, 233

optional whitespaces, matching, 96–98
PowerGREP horizontal rule elements, 343–346

HTTP (HyperText Transfer Protocol), 321–322

hyperlinks, 6–7

hyphen (-), 228–230

I

if statement with nested for loop, 576

image columns, SQL Server 2000, 391

implementation, differences among, 12–13

index, SQL Server 2000 full-text search, 380–385

inline comments, C#, 539–541

input box, VBScript, 461

installing

Java, 620

MySQL relational database, 393–394

Perl, 659–662

PHP 549–553

instances, JavaScript and JScript patterns, 432–433

interactions, debugging, 251

interface. *See* user interface

Internet Explorer (Microsoft)

forward-slash syntax, RegExp object instance, 433–436

JavaScript enabling, 430–432

length property, VBScript strings, 472–473

positional metacharacters, VBScript, 476

properties, RegExp object, 439–441

Internet protocols, 320–321

inventory, matching, 79–80

IP address

HTML document style, 233

positional metacharacters, using, 161–163

ranges, character class, 120–127

IsMatch() method, C#, 520–521

J

Java

character sequence, replacing, 635–638, 642–644

described, 619

first character, position in most recent match (start() method), 644

first matching character sequence, 644

groups, counting (groupCount() method), 639

last character, position of (end() method), 638

Matcher class methods, listed, 634–635

matches() method, 621

metacharacters listed, 645

methods, pattern class, 633–634

modes, pattern class, 629–632

obtaining and installing, 620

pattern class described, 620–621

patterns, returning, 642

positive and negative character classes, combining, 137–139

regular expressions, role of, 232–233

simple examples, 621–629

state information, resetting, 644

string class methods, 654–658

strings of previous argument, returning (group() method), 638

substring, test string (find() method), 638

syntax error (PatternSyntaxException class), 644

test strings, 639–642

tools, 30

Java metacharacters

character classes, 647–651

escaped, 653–654

listed, 645

POSIX character classes, 651–652

single numeric digit, 645–647

Unicode character classes and character blocks, 652–653

JavaScript

attributes of RegExp object, 438

described, 29, 429

documenting, 452

forward-slash syntax for RegExp object instance, 433–436

global property, exec() method of RegExp object, 441–443

metacharacters, 451

nonglobal property, exec() method of RegExp object, 444–445

parentheses and global matching, exec() method of RegExp object, 445–448

patterns with instances of RegExp object, 432–433

position of last match, RegExp object, 438

RegExp() constructor, 436–437

regular expressions, role of, 232–233

source text, holding, RegExp object, 438–440

SSN validation example, 452–454

string matching pattern, test() method of RegExp object, 441

String object, 448–451

JScript

attributes of RegExp object, 438

described, 29, 429

documenting, 452

forward-slash syntax for RegExp object instance, 433–436

global property, exec() method of RegExp object, 441–443

metacharacters, 451

nonglobal property, exec() method of RegExp object, 444–445

parentheses and global matching, exec() method of RegExp object, 445–448

patterns with instances of RegExp object, 432–433

position of last match, RegExp object, 438

RegExp() constructor, 436–437

source text, holding, RegExp object, 438–440

SSN validation example, 452–454

string matching pattern, test() method of RegExp object, 441

String object, 448–451

JScript.NET, 430

K

Komodo Regular Expressions Toolkit, 28

L

languages, 17

last character, position after. *See also* \$ (dollar sign)

- with caret (^) metacharacter, 153–157
- described, 149–150

languages, 17

- findstr utility, 315
- in multiline mode, 150–152
- MySQL, 404–406
- part numbers, matching, 153–155
- positional metacharacters, 149–157
- PowerGREP, 339–340
- VBScript, 475–478

last character, position of, 638

last match position, 438

last name

- selecting specified, 109–111
- swapping with first, 467

lazy matching, Microsoft Word, 265–268

length property, VBScript string matches, 472–473

less-than symbol (<), 164–166

LIKE keyword, SQL Server 2000, 366

line boundary, 144

line or string, immediately after beginning

- dollar sign (\$) metacharacter, 153–157
- findstr utility, 315
- first line of file, examining only, 146–148
- line or string, matching immediately after beginning, 144–146
- literal character, matching, 133–135
- metacharacters, 133–135
- MySQL, 404–406
- negated character classes, 376–379
- part numbers, 153–155
- positional metacharacters, 144–146
- within square brackets [], 133–135
- VBScript, 475–478
- Visual Basic .NET, 505

lines

- blank, matching, 155–157
- entirely of desired characters, finding, 153–155

linking URLs, 6–7

listings, manipulating

- described, 7–8
- VBScript, 455

literal characters

- alternation, 178–179
- caret (^), 133–135
- MySQL relational database, 400
- parentheses, 175
- period (.), 80–81
- square brackets ([]), 113–114
- two, alternation, 178–179

literal text, 306–308

lookahead

- apostrophe, inserting, 205–209
- commas, adding to large numbers, 216–220
- described, 195–196, 197–198
- negative, 202–203
- OpenOffice.org Writer, 294
- Perl, 696–698

- positions, matching between characters, 214–216

positive, 199–202

PowerGREP 342–343

question mark (?), 196–197

sensitivity and specificity, 232

VBScript, 479–481

Visual Basic .NET (Microsoft), 510

Word, 265

lookbehind

commas, adding to large numbers, 216–220

described, 195–196

negative, 213–214

OpenOffice.org Writer, 294

Perl, 698–699

positions, matching between characters, 214–216

positive, 209–213

PowerGREP 342–343

sensitivity and specificity, 232

Visual Basic .NET (Microsoft), 510

Word, 265

lowercase character, matching

described, 75–78

escaping, 102

inventory, matching, 79–80

PowerGREP 332–333

M

m / / operator, 667–669

Match class, C#, 532–533

Match() method

C#, 521

Regex object, 492–493

Matcher Java class methods, listed, 633–635

Matches collection, VBScript, 471–473

Matches() method

C#, 522–526

Regex object, 493–495

matches() method, Java

described, 621

pattern class, 633

matching modes, Perl, 700–702

matching time example, 346–349

metacharacters. *See also* positional metacharacters;

specific metacharacters

alphabetic character or numeric digit, matching any, 75–78

case sensitivity, 16

characters versus positions, 74–75

described, 49–54, 73

digits and nondigits, 83–92

findstr utility, 308–309

Java, 645

JavaScript and JScript, 451

MySQL relational database, 396–397

OpenOffice.org Writer, 284

PCRE, 585–586

Perl, 684–685

PHP 581–582

PowerGREP 331–332

sensitivity and specificity, 230–233

SQL Server 2000, 365–366

Unicode, 607–608

VBScript, 473–474
 Visual Basic .NET, 508–509
 Visual C#.NET, 542–543
 W3C XML Schema, 612–613
 whitespace and non-whitespace, 92–103
 Word, 256, 422

methods

Java pattern class, 633–634
 Regex object, 491–492, 495–497

Microsoft Access database management system

asterisk (*) metacharacter, 423–424
 character classes, 426–428
 date/time data, 425
 described, 413
 hard-wired query, 414–419
 interface, 413–414
 metacharacters listed, 422
 numeric digit, 424
 parameter query, 419–421
 question mark (?) metacharacter, 422–423

Microsoft Excel wildcards

in data forms, 360–362
 described, 28–29, 351
 escaping, 359
 in filters, 362–363
 Find interface, 351–355
 listed, 355–358

Microsoft Internet Explorer

forward-slash syntax, RegExp object instance, 433–436
 JavaScript enabling, 430–432
 length property, VBScript strings, 472–473
 positional metacharacters, VBScript, 476
 properties, RegExp object, 439–441

Microsoft .NET Framework, 430. See also C#; Visual Basic .NET (Microsoft)**Microsoft Visual Basic .NET**

CaptureCollection and Capture classes, 499–502
 classes of System.Text.RegularExpressions, 490
 described, 485
 GroupCollection and Group classes, 497–499
 inline documentation, IgnorePatternWhitespace option, 505–507
 lookahead and lookbehind, 510
 metacharacters, 508–509
 namespace of System.Text.RegularExpressions, 486–489
 Regex object, 490–497
 RegexOptions enumeration, 502–505
 right to left matching, 507
 simple example, 486–489
 tools, 29

Microsoft Visual C# .NET, 545–547

back references, 545–547
 metacharacters, listed, 542–543
 named groups, 544–545

Microsoft Windows findstr utility. See findstr utility**Microsoft Windows Installer, 550–553****Microsoft Word**

back references, 265, 275–278
 character class examples, including ranges, 268
 character classes, 265

dates, manipulating, 273–275
 lazy matching versus greedy matching, 265–268
 lookahead and lookbehind, 265
 macro, VBA, 278–280
 metacharacters, 256
 modes, 262–265
 name structure, changing with back references, 270–272
 quantifier metacharacters, 256–260
 user interface, 253–254
 whole word searches, 269–270
 wildcards, 23–27, 254–255

misspellings, 247**modes**

OpenOffice.org Writer, 286
 sensitivity and specificity, 232
 Word, 262–265

modifiers

case-insensitive search, 104
 described, 103
 global search, 103
 PCRE, 570–571

Mozilla Firefox

forward-slash syntax, RegExp object instance, 436–437
 JavaScript enabling, 430
 regular expressions support, 430

multiline matching

described, 150–152
 Java, 632

RegexOptions enumeration, 505

multiple alternation options, 180–182**multiple matches in string, PCRE, 579–580****multiple optional characters, 59–61****multiple solutions for single problem, 16****MySQL relational database**

character classes and negated character classes, 406–408
 creating database, 394–396
 described, 393
 downloading and installing, 393–394
 literal underscore (_) and percent (%) , 400
 metacharacters, listed, 396–397
 quantifiers, 408–409
 REGEXP keyword, 401–404
 SSN example, 410–411
 tools, 30
 underscore(_) and percent (%) metacharacters, 397–399

N

\n metacharacter, 99–101

{n} syntax, 66

{n, } syntax, 70–71

{n, m} syntax

described, 66, 67, 69–70

Word quantifier, 258, 260–262

named groups

capturing parentheses, 187–188
 Visual C#.NET, 544–545

names

data problems, 246–247
 reversing, 519
 reversing order and adding comma, 467

names (continued)

- sensitivity and specificity, 235–236
 - structure, changing with back references, 270–272
 - swapping, VBScript, 482–483
- negated character classes**
- described, 136
 - Java, 647
 - MySQL, 406–408
 - PHP, 587–589
 - positive, combined with, 137–139
 - SQL Server 2000, 376–379
 - Word, 427–428
- negative lookahead**
- described, 202–203
 - Perl, 697
 - VBScript, 479–481
 - Visual Basic .NET, 510
- negative lookbehind, 213–214, 510**
- nested, numbering, 186–187**
- .NET Framework (Microsoft), 430. See also C#; Visual Basic .NET (Microsoft)**
- newline character, 78, 93–95, 99–101**
- `NextMatch()` **C# method, 533–536**
- non-capturing parentheses, 188–189**
- nondigits, mixed with digits, 83, 89–90**
- non-English character matching**
- described, 75–78
 - escaping, 102
 - inventory, matching, 79–80
 - PowerGREP, 332–333
- non-English characters. See . (period)**
- nonglobal property, JavaScript and JScript, 444–445**
- nongrouping parentheses, VBScript, 482–483**
- non-matching lines, displaying, 316–318**
- number sign (#)**
- Access date/time data, 425
 - Word database management system, 424
- numbered groups**
- C#, 519
 - creating, 185–186
- numeric digits. See also . (period)**
- `\d` metacharacter, 83–88
 - number sign (#) metacharacter, Access, 424
 - OpenOffice.org Writer, 302–304
 - POSIX character classes, 302–304
 - PowerGREP, 332–333
 - W3C XML Schema, 614–615
 - 0 through 9, 83–88
- numeric order, reverse, 128–129**
- O**
- obtaining**
- Java, 620
 - MySQL relational database, 393–394
 - Perl, 659–662
 - XML editors, 593
- one (1), 518**
- one character matching**
- described, 42–47
 - MySQL relational database, 397–399
 - Word, 258, 422–423

- online chat search-and-replace example, 297–301**
- online searching, 8**
- open parens, question mark (?), 505**
- OpenOffice.org Writer**
- `[:alnum:]`, using quantifiers with, 140–141
 - alternation, 289–292
 - back references, limited (& metacharacter), 292–294
 - character classes, 286–289
 - described, 27
 - Find All button, 281
 - lookahead and lookbehind, 294
 - metacharacters, listed, 284
 - modes, 286
 - online chat search-and-replace example, 297–301
 - POSIX character classes, 301–304
 - quantifiers, 285–286
 - search example, 294–296
 - support for regular expressions, 281
 - user interface, 282–283
- operators, Perl, 667**
- optional characters**
- described, 56–58
 - multiple, 59–61
- Options property, 518**
- order**
- reverse alphabetic or numeric, 128–129
 - reversing in VBScript string, 465–467
- overload, C#, 532**

P

- parameter query, 419–421**
- parentheses ()**
- alternation, 177–185
 - back references, 190–193
 - C#, 519
 - capturing, 185–188, 499–502
 - characters, choice between two, 108
 - `GroupCollection` and `Group` classes, Visual Basic .NET, 497–499
 - grouping, 171–177, 482–483
 - JavaScript and JScript, 445–448
 - literal, matching, 175
 - named groups, 187–188
 - nested, numbering, 186–187
 - non-capturing, 188–189
 - numbering groups, 185–186
 - PowerGREP, 335–339
 - VBScript, 482–483
 - Visual Basic.NET, 499–502
- parentheses-enclosed question mark (?), 196–197**
- parts numbers**
- digit ranges, character class matching, 117–119
 - lookahead example, 198–199
 - matching with caret (^) and dollar sign (\$) metacharacters, 153–155
 - variably structured, matching, 78–80
- pattern class, Java**
- case-insensitive matching, U.S. ASCII characters, 629
 - characters, 632
 - comments, 630–632
 - described, 620–621

- DOTALL mode, 632
- MULTILINE flag, 632
- non-U.S. characters, case-insensitive flag, 632
- properties (fields), 629
- Unix multiline text, 632
- pattern matching**
 - assigning to variable with `RegExp` object, 438
 - Canadian postal code, 85–87
 - DATE type columns, MySQL database, 399
 - Java, 642
 - JavaScript and JScript, 432–433
 - PCRE delimiters, 568–570
- pattern property**
 - string value, holding in VBScript, 456–458
 - VBScript, 456–458
- PCRE (Perl Compatible Regular Expressions)**
 - array, matching, 576–578
 - callback to another function, 580
 - character classes, 587–589
 - character sequences in string, matching all, 574–576
 - escaping pattern delimiters, 570
 - escaping strings, 579
 - groups, matching pattern against test string, 572–574
 - listed, 568
 - metacharacters, listed, 585–586
 - modifiers, matching, 570–571
 - multiple matches in string, 579–580
 - pattern delimiters, 568–570
 - positional metacharacters, 586
 - splitting test string, 580–581
 - string, matching pattern against test, 571–572
- percent sign (%)**
 - MySQL relational database, 397–399
 - SQL Server 2000, 366–372
- period (.)**
 - described, 75–78
 - escaping, 102
 - inventory, matching, 79–80
 - PowerGREP, 332–333
- Perl**
 - alternation (`|`), 690–691
 - captured groups, 687–689
 - case insensitive string, 669–674
 - character classes, 692–696
 - delimiters, 675–676
 - described, 659
 - doubled words, detecting with back references, 689–690
 - lookahead, 696–698
 - lookbehind, 698–699
 - matching modes, 700–702
 - metacharacters, listed, 684–685
 - obtaining and installing, 659–662
 - operators, listed, 667
 - positional metacharacters, 686–687
 - quantifiers, 685–686
 - simple program, 663–666
 - string (`m//` operator), 667–669
 - strings, replacing (`s///` operator), 678–684
 - tools, 30
 - variable, matching against, 674–675
 - variable substitution, 676–678
- Perl Compatible Regular Expressions. See PCRE**
- PHP (PHP Hypertext Processor)**
 - case insensitive matching, `eregi()` function, 559–564
 - described, 549
 - documenting, 589–590
 - installing, 549–553
 - metacharacters, listed, 581–582
 - PCRE, listed, 568
 - POSIX regular expressions, 553–556
 - regular expression pattern, creating (`sql_regcase()` function), 567
 - strings, splitting (`split()` function), 564–567
 - three arguments, `ereg()` function, 556–559
 - tools, 30
- pipe symbol (|), 690–691**
- plus sign (+)**
 - cardinality operators, 64–66
 - in OpenOffice.org Writer, 285–286
 - PowerGREP, 333–335
 - VBScript, 474
 - Word, 256, 258–260
- position**
 - characters versus, 74–75
 - first in most recent match (`start()` method), 644
 - last, 438, 638
- positional metacharacters**
 - described, 143
 - end of line or file, immediately before end, 149–157
 - first line of file, examining only, 146–148
 - IP address, 161–163
 - line or string, immediately after beginning (`^` metacharacter), 144–146
 - MySQL, 404–406
 - PCRE, 586
 - Perl, 686–687
 - PowerGREP, 339–340
 - sensitivity and specificity, 231
 - VBScript, 475–478, 476, 478
 - W3C XML Schema, 613–614
 - word boundaries, 164–169
- positive character classes**
 - negated, combined with, 137–139
 - Word, 426–427
- positive lookahead**
 - later in same sentence, 200–202
 - Perl, 697
 - same document, 203–205
 - Star Training example, 199–200
 - VBScript, 479–481
 - Visual Basic .NET, 510
- positive lookbehind, 209–213, 510**
- POSIX character classes**
 - `[:alnum]`, 139–141, 583–584
 - described, 139
 - Java metacharacters, 651–652
 - OpenOffice.org Writer, 301–304
 - PHP, 553–556, 582–585
- PowerGREP**
 - alternation, 339
 - described, 28, 325
 - File Finder tab, 329–330
 - HTML horizontal rule elements, 343–346
 - interface, 325–326

PowerGREP (continued)

- lookahead and lookbehind, 342–343
- matching time example, 346–349
- metacharacters, listed, 331–332
- numeric digits and alphabetic characters, 332–333
- other tabs, 331
- positional metacharacters, 339–340
- quantifiers, 333–335
- Replace tab, 328–329
- simple find example, 326–328
- syntax coloring, 330–331
- word-boundary metacharacters, 340–342

preceding character or group, 258–260

problem definition, 242–243

proper names. See names

properties

- Internet Explorer, 439–441
- Java pattern class, 629
- Regex object, 491

Q

quantifiers

- [:alnum:], using with, 140–141
- character classes, using with, 111–115
- findstr utility, 310–311
- grouping, parentheses, 173–175
- MySQL, 408–409
- OpenOffice.org Writer, 285–286
- Perl, 685–686
- PowerGREP, 333–335
- VBScript, 474
- Word, 256–260

question mark (?)

- OpenOffice.org Writer, 285–286
- PowerGREP, 333–335
- VBScript, 474
- Word, 256, 258, 422–423

question mark enclosed in parentheses (?), 196–197

R

ranges, character class

- alphabetic, 115–117
- date separators, differing, 129–132
- described, 114–115
- digit, 117–119
- hexadecimal numbers, 119–120
- IP addresses, 120–127
- Java, 647
- negated, 378
- reverse, 128–129
- Word examples, 268

read-only property, VBScript, 471

Regex object

- C#, 517–519
- described, 490
- Match() method, 492–493
- Matches() method, 493–495
- Match.Success property and Match.NextMatch method, 495–497

- methods, listed, 491–492
- properties, 491

Regex **static methods**, 531

RegexOptions **enumeration**

- C#, 539
- case-insensitive matching, 502–505
- described, 502
- multiline matching, 505

RegexTester, Perl, 703–705

regular expression tools

- C#, 30
- described, 21–22
- findstr, 22–23
- Java, 30
- JavaScript and JScript, 29
- Komodo Regular Expressions Toolkit, 28
- MySQL, 30
- Perl, 30
- PHP, 30
- PowerGrep, 28
- SQL Server 2000, 31
- StarOffice Writer/OpenOffice.org Writer, 27
- VBScript, 29
- Visual Basic.NET, 29
- W3C XML Schema, 31, 592–593
- Word wildcards, 23–27, 28–29

regular expressions

- case sensitivity, 15–16
- characters differ among contexts, 13–15
- choosing, 17
- date formats, changing, 6
- described, 2–5
- directory listings, 7–8
- doubled words, finding, 5
- evolution, 16
- implementation, differences among, 12–13
- incorrect case, finding, 6
- lack of standards body, 12
- languages supporting, 17
- multiple solutions for single problem, 16
- online searching, 8
- syntax, 8
- text, replacing in quantity, 17–19
- URLs, linking, 6–7
- Web form input, checking, 5
- whitespace, meaning and, 9–11
- word processor search and replace, 7

replacing

- all strings, Java, 658
- back references, 275–278
- C# overload, 532
- client-side functions, 455
- dates, manipulating, 273–275
- first string, Java, 656–657
- Java character sequence, 635–638, 642–644
- name structure, changing with back references, 270–272
- PHP, 561–564
- PowerGREP, 328–329, 336–339
- strings, (s/// operator), 678–684
- text in quantity, 17–19
- VBScript, 465–467

resetting Java state information, 644
restriction, W3C XML Schema, 602–603
results of regular expression, testing, 38–39
reverse numeric order, 128–129
reverse ranges, character class, 128–129
reversing string order, VBScript, 465–467
right to left character matching
 C#, 518
 Visual Basic.NET, 507

S

\s metacharacter, 615
s/// operator, Perl, 678–684
sales data for particular period, 414–419
Scalable Vector Graphics (SVG), 119–120
scripting. See JavaScript; JScript
search and replace text, Java, 619
search example
 OpenOffice.org Writer, 294–296
 words in proximity, 294–296
search-and-replace examples
 back references, 275–278
 dates, manipulating, 273–275
 name structure, changing with back references, 270–272
security settings
 JavaScript and JScript forms validation, 430, 431–432
 Word macro, 280
SELECT statement, 401–404
sensitivity
 data and, 233–236
 described, 34–35, 221–222
 email addresses example, 224–228
 extreme sensitivity, awful specificity, 222–223
 hyphens, replacing, 228–230
 metacharacters, 230–233
 Star Training example, 236–240
 trade-off, 230
sentence, words occurring in same, 200–202
sequences, character
 C#, 526–528
 different, 54–56
 followed by other sequence of characters, 199–202
 not followed by another sequence of characters, 202–203
 not preceded by another sequence of characters, 213–214
 preceded by another sequence of characters, 209–213
 replacing Star with Moon in example, 237–240
 in string, matching all, PCRE, 574–576
server-side validation, forms data, 429
sets, character, 478
single character matching
 described, 42–47
 MySQL relational database, 397–399
 Word, 258, 422–423
single file examples, 319–321
single numeric digit. See also \d metacharacter
 Java, 645–647
 PowerGREP 332–333
 W3C XML Schema, 614–615
Social Security number. See SSN
source text, JavaScript and JScript, 438–440
space character
 metacharacter matching, 93–95
 PHP, 584–585
specificity
 asterisk (*) metacharacter in Word, 258
 data and, 233–236
 described, 34–35, 221–222
 email addresses example, 224–228
 extreme sensitivity, awful specificity, 222–223
 hyphens, replacing, 228–230
 maximizing, 236
 metacharacters, 230–233
 Star Training example, 236–240
 trade-off, 230
splitting
 C# overload, 532
 C# string, 528–531
 Java pattern class, 634
 Java strings, 658, 682–684
 PCRE test string, 580–581
 PHP strings, 566
SQL Server 2000
 character classes, 373–376
 described, 365
 document filters on image columns, 391
 LIKE keyword, 366
 metacharacters, 365–366
 negated character classes, 376–379
 one character, exactly (_ metacharacter), 372–373
 tools, 31
 zero or more characters (% metacharacter), 366–372
SQL Server 2000 full-text search
 CONTAINS predicate, 386–390
 described, 379
 index, enabling and creating, 380–385
square brackets ([])
 literal, 113–114
 metacharacter, 133–135
SSN (Social Security number)
 C# example, 541
 inline document, Visual Basic.NET, 506–507
 JavaScript and JScript validation example, 452–454
 MySQL example, 410–411
 x matching modifier, documenting in PHP, 590
standards body, absence of, 12
Star Training Company examples
 back references, 237–240, 275–278
 PCRE, 580
 positive lookahead, 199–200
StarOffice Writer
 described, 27
 support for regular expressions, 281
state information, Java, 644
static methods, C# Regex class, 531–532
string boundary, 144
string class methods, Java
 all, replacing, 658
 first, replacing, 656–657
 splitting, 658

string matching

- C# argument, 520–521
- Execute() method, VBScript, 467–471
- Java, 654–656
- JavaScript and JScript, 441, 448–451
- PCRE, 571–572
- Perl operator (m/ /), 667–669
- previous argument, returning (group() method), 638
- Test() method, VBScript RegExp object, 464–465
- Visual Basic .NET example, 486–489

string pattern. See also findstr utility

- beginning- and end-of-line positions, 315–316
- character classes, 311–313
- command-line switches, 316–319
- described, 22–23, 305–306
- filelist example, 322–323
- literal text, 306–308
- metacharacters, 308–309
- multiple file examples, 321–323
- quantifiers, 310–311
- single file examples, 319–321
- word-boundary positions, 313–315

string value, VBScript pattern property, 456–458

strings, escaping, 579

strings, replacing

- Java, 678–684
- Perl, 678–684
- VBScript, 465–467

strings, splitting

- case-insensitive, 566–567
- case-sensitive, 564–566
- described, 528–531

StylusStudio, 592, 594. See also W3C XML Schema

substring, Java, 638

subtraction, Java character classes, 650–651

surname. See last name

SVG (Scalable Vector Graphics), 119–120

switches, command-line

- /a, 318–319
- /v, 316–318

syntax

- Java error, 644
- JavaScript and JScript, 433–436
- {n}, 66
- {n, }, 70–71
- {n, m}, 333–335
- PowerGREP coloring, 330–331
- problems with, 8
- Word quantifier, 258, 260–262

System.Text.RegularExpressions namespace. See C#

T

\t metacharacter, 98–99

tab character

- metacharacter matching, 93–95, 98–99
- PowerGREP, 331

Test() method, VBScript, 464–465

test string

- C#, 522–526
- Java, 638, 639–642

testing

- debugging cases, creating, 247–248
- results of regular expression, 38–39

text

- colors, 318–319
- JavaScript and JScript source, 438–440
- replacing in quantity, 17–19
- replacing, PHP, 561–564

text validation. See Java

time

- Access, 425
- PowerGREP example, 346–349

tools

- C#, 30
- described, 21–22
- findstr, 22–23
- Java, 30
- JavaScript and JScript, 29
- Komodo Regular Expressions Toolkit, 28
- MySQL, 30
- Perl, 30
- PHP, 30
- PowerGrep, 28
- SQL Server 2000, 31
- StarOffice Writer/OpenOffice.org Writer, 27
- VBScript, 29
- Visual Basic.NET, 29
- W3C XML Schema, 31, 592–593
- Word wildcards, 23–27, 28–29

triple character matching, 49

true

- C# string argument, 520–521
- exec() method of RegExp object, 441–443
- Execute() method, VBScript, 470–471
- JavaScript and JScript, 441–443
- positional metacharacters, VBScript, 478
- Test() method, VBScript RegExp object, 464–465
- VBScript Global property, 458–462

two literals, alternation, 178–179

U

underscore (_)

- matching characters excluding, 82–83, 332–333
- MySQL relational database, 397–399
- SQL Server 2000, 372–373

undesired text, documenting, 37

Unicode

- character blocks, 608–612
- character classes, 605–606
- decimal numbers, 606–607
- described, 604–605
- Java metacharacters, 652–653
- JavaScript/JScript string, 448–451
- metacharacters, mixing, 607–608
- numeric references, 130
- W3C XML Schema, 604–614

Uniform Resource Locators (URLs)

- HTTP, finding across multiple files, 321–322
- linking, 6–7

union, Java character classes, 649–650

Unix multiline text, 632**uppercase characters, matching**

- beginning or end boundary with (b), 168
- Canadian postal code, 87–88

URLs (Uniform Resource Locators)

- HTTP, finding across multiple files, 321–322
- linking, 6–7

U.S. telephone numbers, 175–177**U.S. zip codes, 645–647****user input**

- C#, 516
- doubled words, finding and removing in Visual C#.NET, 546–547
- validating lines comprised entirely of desired characters, 153–155
- VBScript, 461

user interface

- OpenOffice.org Writer, 282–283
- PowerGREP, 325–326
- Word, 253–254, 413–414

user-set query, 419–421**V****/v command-line switch, 316–318****validation. See also JavaScript; JScript**

- described, 5
- lines comprised entirely of desired characters, 153–155
- Web form input, 5
- Word wildcards, 360–362
- XML, 429

variable

- assignment statement, 438
- Perl, 674–678

VBA (Visual Basic for Applications), 278–280**VB.NET, 232–233****VBScript**

- character classes, 478
- described, 455
- Execute() method, Matches collection, 471–473
- Execute() method, RegExp object, 467–471
- Global property, RegExp object, 458–462
- grouping and nongrouping parentheses, 482–483
- IgnoreCase property, RegExp object, 462–464
- lookahead, 479–481
- metacharacters, listed, 473–474
- pattern property, RegExp object, 456–458
- quantifiers, 474
- RegExp object, described, 455–456
- Replace() method, RegExp object, 465–467
- Test() method, RegExp object, 464–465
- tools, described, 29
- word boundaries, 479

VBScript interpreter, 455**Visual Basic for Applications (VBA), 278–280****Visual Basic .NET (Microsoft)**

- CaptureCollection and Capture classes, 499–502
- classes of System.Text.RegularExpressions, 490
- described, 485
- GroupCollection and Group classes, 497–499

- inline documentation, IgnorePatternWhitespace option, 505–507

lookahead and lookbehind, 510

metacharacters, 508–509

namespace of System.Text.RegularExpressions, 486–489

Regex object, 490–497

RegexOptions enumeration, 502–505

right to left matching, 507

simple example, 486–489

tools, 29

Visual C# .NET (Microsoft)

back references, 545–547

metacharacters, listed, 542–543

named groups, 544–545

Visual Studio 2003. See C#; Visual Basic .NET (Microsoft)**W****\w metacharacter**

described, 82–83

PowerGREP, 332–333

\w metacharacter

described, 81–82

W3C XML Schema, 615

Web forms validation, 429. See also JavaScript; JScript**Web pages**

client-side replace functions, 455

color values, matching hexadecimal number ranges, 119–120

heading elements, finding, 132–133

IP address style, amending, 233

optional whitespaces, matching, 96–98

PowerGREP horizontal rule elements, 343–346

Web servers, 120–127**Web-based applications. See PHP (PHP Hypertext Processor)****while loop**

C# matching, 536

Java, 649

whitespace

any single, matching, 93–95

debugging, 248–251

documentation, 37–38

escaped characters, 102–103

JavaScript and JScript, 452

meaning and, 9–11

newline character, matching (n metacharacter), 99–101

non-whitespace, matching any, 98

optional, matching, 96–98

tab, matching, 98–99

VBScript, 483

in XML, 92–93, 615

wildcards

Access, primary use in, 413

filenames, searching, 319–322

Word, 254–255, 262–265

wildcards, Excel (Microsoft)

in data forms, 360–362

described, 28–29, 351

wildcards, Excel (Microsoft) (continued)

- escaping, 359
- in filters, 362–363
- Find interface, 351–355
- listed, 355–358

Windows (Microsoft) findstr utility. See findstr utility

Windows (Microsoft) Installer, 550–553

Windows Script Host (WSH), 455

word

- identifying, 164
- metacharacters, 144
- proximal, searching for, 294–296
- two, occurrence in same sentence, 200–202
- W3C XML Schema, 615

word boundary

- beginning, identifying (<), 164–166
- end, identifying (>), 166–167
- positions, findstr utility, 313–315
- PowerGREP, 340–342
- uppercase letter, beginning or end, 168
- VBScript, 479

Word (Microsoft)

- back references, 265, 275–278
- character class examples, including ranges, 268
- character classes, 265
- dates, manipulating, 273–275
- lazy matching versus greedy matching, 265–268
- lookahead and lookbehind, 265
- macro, VBA, 278–280
- metacharacters, 256
- modes, 262–265
- name structure, changing with back references, 270–272
- quantifier metacharacters, 256–260
- user interface, 253–254
- whole word searches, 269–270
- wildcards, 23–27, 254–255

word processor search and replace, 7

WSH (Windows Script Host), 455

W3C XForms specification, 429

W3C XML Schema

- alternation, 615
- constraints, 598
- datatypes, 599–601
- derivation by restriction, 602–603
- described, 591–592
- DTDs versus, 593–598
- editors, trial downloads, 593
- escaping metacharacters, 616
- metacharacters listed, 612–613
- numeric digits, 614–615

- positional metacharacters, 613–614
- tools, 31, 592–593
- Unicode, 604–614
- whitespace characters (*s* metacharacter), 615
- word characters (*w* metacharacter), 615

X

x modifier

- Perl, 700–701
- PHP, 589–590

XForms specification, 429

XHTML (eXtensible HyperText Markup Language)

- color values, matching hexadecimal number ranges, 119–120
- optional whitespaces, matching, 96–98

XML (eXtensible Markup Language)

- instance document, creating, 592, 594, 595–598
- optional whitespaces, matching, 96–98
- Web forms validation, W3C specification, 429
- whitespace and non-whitespace metacharacters, 92–93

XML Schema

- alternation, 615
- constraints, 598
- datatypes, 599–601
- derivation by restriction, 602–603
- described, 591–592
- DTDs versus, 593–598
- editors, trial downloads, 593
- escaping metacharacters, 616
- metacharacters listed, 612–613
- numeric digits, 614–615
- positional metacharacters, 613–614
- tools, 31, 592–593
- Unicode, 604–614
- whitespace characters (*s* metacharacter), 615
- word characters (*w* metacharacter), 615

XMLSpy, 592, 594, 595–598. See also W3C XML Schema

XMLWriter. See W3C XML Schema

Z

zero (0), 260–262, 518

zero or more characters

- MySQL relational database, 397–399
- SQL Server 2000, 366–372
- Word, 257–258, 423–424

zero through nine (0 through 9), 83–88

zero width lookahead assertion. See lookahead

zip codes, 645–647