

## Index

### Symbols

- (auto-decrement operator), 120
- (hyphen). *See* hyphen
- \_\_DATA\_\_ token, 368–69, 379
- \_\_END\_\_ token, 348, 368–69, 379
- , (comma). *See* comma
- ; (semicolon). *See* semicolon
- :: (scope operator), 305, 365
- ! (logical ‘not’ operator), 129
- != (not equal operator), 126
- !~ (not match operator), 150
- ? (question mark). *See* question mark
- . (dot). *See* dot
- .. (dot-dot). *See* dot-dot
- / (dot-slash), 97
- . = (concatenation operator), 142
- .login file, 87
- .profile file, 87, 88, 90, 99
- .zprofile file, 87
- ‘ (single quotes). *See* quotation characters
- “ (double quotes). *See* quotation characters
- () . *See* parentheses
- [] . *See* square brackets
- { } . *See* curly braces
- @ (at-sign). *See* array
- @\_ array, 248–49
- @ARGV array, 176–79, 202, 203
- @INC array, 301
- \* (asterisk). *See* asterisk
- \*\* (exponentiation operator), 117, 176
- \*= (multiplication operator), 120
- / (forward slash). *See* forward slash
- /= (division operator), 120
- /dev/null, 284
- \ (backslash). *See* backslash
- \1, \2, \3 (backreferences), 298
- \B (non-word boundary metacharacter), 297
- \b (word boundary metacharacter), 297
- \d (digit metacharacter), 233
- \D (non-digit metacharacter), 233
- \n (newline). *See* newline
- \r (carriage return). *See* carriage return
- \S (non-whitespace metacharacter), 233
- \s (whitespace metacharacter), 233
- \t (tab). *See* tab
- \W (non-word metacharacter), 233
- \w (word metacharacter), 233
- & (ampersand), 249
- && (logical ‘and’ operator), 129
- # (pound sign). *See* pound sign
- #! (hashbang), 100
- % (percent sign). *See* percent sign
- ` (backticks). *See* backticks
- ^ (caret). *See* caret
- ^M (Ctrl + M), 289
- + (plus sign). *See* plus sign
- ++ (auto-increment operator), 120
- += (addition operator), 120
- < (less than sign)
  - filehandle read mode, 207
  - less than operator, 126
  - redirecting standard input, 280
- <= (less than or equal to operator), 126
- <=> (spaceship operator), 183
- <> (file operator), 202–04, 206
- = (assignment operator), 110
- = (subtraction operator), 120
- == (equality operator), 125, 126
- => (hash assignment operator), 217
- =~ (binding operator), 149, 153, 158
- > (arrow operator). *See* arrow operator
- > (greater than sign)
  - filehandle write mode, 207
  - greater than operator, 126
  - in FASTA header lines, 260
  - redirecting standard output, 278
- >= (greater than or equal to operator), 126
- >> (double greater than sign)
  - appending standard output, 278
  - filehandle append mode, 207
- | (vertical bar). *See* pipe character
- || (logical ‘or’ operator), 129
- ~ (tilde), 34, 66
- \$ (dollar sign). *See* dollar sign
- \$\_ (default variable), 204–05, 296
- \$(system error variable), 210
- \$(exit status variable), 242
- \$(postmatch variable), 298
- \$(match variable), 298

- \$` (prematch variable), 298
- \$0 (program name variable), 378
- \$1, \$2, \$3 (regular expression matches), 237, 238, 298
- \$a and \$b (special package variables), 184, 223
- \$HOME (environment variable), 39, 86, 99
- \$i (loop counter variable), 188
- \$PATH (environment variable), 96, 97, 98, 124
- \$PERL5LIB (environment variable), 302
- \$SHELL (environment variable), 38
- \$USER (environment variable), 39
- 2-dimensional arrays, 309–12
- 2-dimensional data structures, 317
- abs function, 131
- absolute paths, 32, 96
- abstraction in programming, 361–67
- addition operator (+), 117
- alias command, 71
- aliases, 71–75, 80, 81
- ampersand (&), 249
- and (logical operator), 129
- angle brackets (<>), 202
- anonymous arrays, 308, 309
- anonymous hashes, 314
- answer to life, the universe, and everything, 350
- append operator (.=), 142
- append redirection operator (>>), 278
- arrays
  - @ARGV array, 176–79, 202, 203
  - accessing elements, 163–65, 167
  - adding to, and removing from, 169–71
  - combining array elements, 172
  - compared to scalar variables, 164
  - copying, 171
  - creating from string, 173
  - deleting, 171
  - determining length, 166
  - interpolation, 165
  - looping over contents, 190–93, 194
  - overview, 163–64
  - pop function, 170
  - push function, 169
  - qw function, 166, 171
  - references to, 306
  - shift function, 170, 195, 249
  - sorting, 183
  - splice function, 170, 174
  - unshift function, 170
- arrow operator (->)
  - accessing array references, 307
  - in object-oriented programming, 334
- ASCII values, 143, 183, 279
- assignment operator (=), 110
- asterisk (\*)
  - as multiplication operator, 117
  - as wildcard character (Perl), 232
  - as wildcard character (Unix), 58, 62, 66
- auto-decrement operator (--), 120, 189
- auto-increment operator (++), 120
- awk command, 291–92
- backreferences, 298
- backslash (\)
  - as reference operator, 306, 314
  - escaping special characters, 147
  - escaping Unix commands, 74
- backticks (` `)
  - in Unix, 89, 102
  - Perl operator, 241–43
- bareword error messages, 353
- beautification of code, 355–60
- binding operator (=~), 149, 153, 158
- bless keyword, 336
- blocks of code, 126, 127, 135, 188, 201
- caret (^)
  - as regular expression anchor, 229
  - negated character class metacharacter, 228
- carriage return (\r), 289, 386–87
- case of characters
  - associated ASCII values, 143
  - case-sensitivity in Unix, 25, 30
  - for filehandle names, 207
  - how to convert. *See* changing character case
  - ignoring with grep command, 270, 273
  - ignoring with matching operator, 154, 229
  - in FASTA sequence files, 261
  - when naming variables, 112, 357
- cat command, 83, 274, 278, 286
- cd command, 29–31, 32, 33, 34–35

- changing character case
  - using Perl character functions, 146
  - using transliteration (Perl), 156, 229
  - using transliteration (Unix), 279, 286
- changing directories. *See* cd command
- chmod command, 93, 94, 124
- chomp function, 212
- circular references. *See* endless loop
- classes, 333, 335–38
- clear screen, 54
- close function, 208
- cmp operator, 183, 254
- code editors, 15, 346
- columns
  - creating by splitting a string, 174, 235
  - extracting from text file, 287, 290, 291
  - in CSV and TSV files, 148, 369
  - in GFF files, 261
  - in multidimensional data structures, 306, 310, 311, 312
  - sorting data by column, 287
- combining
  - array elements to form strings, 172
  - arrays, 172
  - multiple command-line options, 42
  - standard input/output/error streams, 283
  - strings, 141
  - text files, 278
  - Unix commands, 285
- comma (,)
  - separated values. *See* CSV
  - separating function arguments, 111
  - separating list items, 160
- command not found errors, 351
- command prompts, 23, 90
- command-line completion.
  - See* tab-completion
- command-line options, 40–43, 328
- commenting-out code, 109, 346, 347
- comments
  - in Perl, 107, 109, 357, 378
  - in Unix, 82
- comparing
  - floating point numbers, 131
  - numbers, 126, 183
  - strings, 142, 183
  - Unix to Linux, 12
- compile-time errors, 346
- Comprehensive Perl Archive Network (CPAN), 340–41
- computers vs. humans, 156
- concatenation operator (.=), 142
- conditional statements, 129
  - else, 127
  - elsif, 128
  - if, 126
  - nested conditionals, 128
  - postfix notation, 130
  - trinary operator, 130
  - unless, 130
  - using floating point numbers, 131
  - while, 194–95
- configuration files in Unix, 90
- control characters in Unix
  - Ctrl + a, 53
  - Ctrl + c, 188, 264
  - Ctrl + d, 279
  - Ctrl + e, 54
  - Ctrl + l, 54
  - Ctrl + w, 54
- copying files and directories, 63–67
- cp command, 63
- CPAN (Comprehensive Perl Archive Network), 340–41
- creating
  - 2-dimensional data structures, 309
  - aliases to commands, 71
  - array references, 306
  - arrays, 164
  - arrays from strings, 173
  - directories, 47
  - empty files, 55
  - hash references, 314
  - key/value pairs for hash, 217
  - modules, 364
  - multidimensional data structures, 317
  - new text files, 77
  - objects, 333
  - random numbers, 118
  - shell scripts, 100
  - SQL databases, 374
  - strings from arrays, 172
  - Unix configuration files, 86
- CSV (comma-separated values), 173, 174, 311, 369–70

- curly braces { }
  - in anonymous hash syntax, 314
  - in hash key syntax, 216
  - to denote blocks of code, 126, 127, 188, 246
  - use in dereferencing, 307, 314
- current directory, 28, 64, 65, 96
- cut command, 287, 290
- Cygnwin, 13
- dash (-). *See* hyphen
- DATA token. *See* `__DATA`
- Data::Dumper module, 323–25
- databases
  - creating using dbmopen function, 371
  - creating using Storable package, 372
  - DBI and DBD::SQLite modules, 374
  - hash references as database records, 315
  - normalization and denormalization, 375
  - relational database management systems (RDBMS), 373
  - SQLite, 373–74
  - Structured Query Language (SQL), 373–74
- date command, 43, 89
- dbmopen function, 371
- debugging, 108, 181, 345
  - by commenting-out code, 109, 346, 348
  - common error messages, 351–54
  - compile-time vs. run-time errors, 346
  - different types of error, 345
  - general strategies for, 346
  - Perl syntax checker (-c option), 345
  - using the Perl debugger, 349
  - with print statements, 348–49
- decrement operator (--), 189
- default variable (\$\_), 204–05, 296
- defined function, 180, 221
- deleting files, 61–62
- dereferencing. *See* references
- diagnostics pragma, 352
- die function, 138, 139, 140, 178, 210
- differences
  - between dogs and people, 333
  - between functions from multiple modules, 303
  - between lexical and global variables, 136, 137
  - between scalar and list context, 167
  - between scalars, arrays, and hashes, 220
  - between text and binary files, 385
  - between text editors and word processors, 76
  - between types of line endings, 387
  - between Unix shells, 71, 86, 97
  - between various loop constructs, 196
- digit metacharacter (\d), 233
- directories
  - distinguishing from files, 43, 93
  - parent directories, 28, 30–31
  - working with directories in Unix, 47–49
- documentation
  - for Perl, 120, 377–81
  - man pages in Unix, 44–46
  - pod (Plain Old Documentation), 380–81
- dollar sign (\$)
  - as regular expression anchor, 230
  - how to print, 147
  - prefix for scalar variables, 110, 164
- DON'T PANIC!. *See* help
- dot (.)
  - as concatenation operator, 142
  - as directory location, 64, 65, 84, 96
  - at start of filenames, 83
  - in regular expressions, 230
- dot-dot (..)
  - as parent directory, 30, 65, 84
  - range operator, 193
- dot-profile file (.profile), 87, 88, 90, 99
- dot-slash (./), 97
- each function, 224
- echo command, 38–39
- editing text. *See* text editors
- else statement, 127
- elsif statement, 128
- emacs command, 77
- embedding data inside scripts, 368
- encapsulation, 334
- END token. *See* `__END`
- endless loop. *See* circular references
- environment variables, 38–39, 99, 302
- eq operator, 142
- equality
  - testing for numerical equality, 125
  - testing for numerical inequality, 126
  - testing strings for equality, 142

- equality operator (`==`), 125, 126
- error messages. *See* debugging
- escaping special characters, 147–48, 159, 231
- executable permission, 93
- executing scripts. *See* running scripts
- exists function, 221
- exit function, 139
- exit status variable (`$?`), 243
- exponentiation operator (`**`), 176
- export command, 98
- extracting
  - codons from DNA sequence, 305
  - columns from a file, 287, 291
  - data from a database file, 372, 374, 375
  - elements from an array, 170
  - embedded Perl script documentation, 381
  - first or last lines from a file, 274
  - keys and values from hashes, 221
  - matching lines from a file, 285
  - parts of a pattern match, 236, 298
  - position of a matching pattern, 297
  - substrings from a string, 144
- FASTA files, 259–61, 272
- file command, 68, 386
- file permissions, 93–95
  - changing permissions, 94
  - chmod command, 94
  - viewing current permissions, 93
- filehandles, 207, 209, 212
- files
  - appending to file, 213
  - checking file opened successfully, 210
  - file operator (`<>`), 202–04, 206
  - reading from, 202–05, 207–11
  - writing to, 211–12
- filesystem hierarchy, 26–27
- floating point numbers
  - as array indices, 168
  - precision of, 131
  - specifying with `Getopt::Long` module, 331
- for loop, 187–91, 193
- foreach loop, 192–93
- forward slash (`/`)
  - as delimiter for matching operator, 150
  - as directory delimiter, 28, 30, 56
  - as division operator, 117
  - as root level indicator, 26, 29
  - at end of directory names, 59, 64
  - how to match, 158
- frustration, 377, 385, 389
- function libraries. *See* modules
- functions
  - compared to subroutines, 247
  - documentation, 120
  - naming, 357
  - nesting inside one another, 119
  - numerical functions, 118
  - use of parentheses, 111
- garbage collection, 313, 326
- genome sequence as multidimensional structure, 322
- `Getopt::Long` module, 329–32
- `Getopt::Std` module, 329, 331
- GFF files, 259, 261–62, 286
- global matches, 153, 239
- global symbol error messages, 353
- global symbols, 134
- greater than operator (`>`), 126
- greater than or equal to operator (`>=`), 126
- greedy quantifiers, 295
- grep command, 269–73, 276
- gt operator, 142
- hash sign (`#`). *See* pound sign
- hashes, 214–24
  - adding/removing key-value pairs, 220
  - assignment operator (`=>`), 217
  - checking existence of key, 221
  - defining, 217–18
  - extracting keys and values, 221
  - looping over contents, 221–24
  - naming, 218
  - references to, 314–16
  - saving as a database file, 371
  - sorting, 223
  - syntax, 217
- head command, 274
- hello world program, 91, 107
- help, 389–92
  - built-in support tools, 391
  - online support, 392
  - Perl FAQ, 390
- hidden files in Unix, 83–84

- history command, 52
- home directories, 24, 26, 27, 28, 34–35, 39, 99
- hyphen (-)
  - as subtraction operator, 117
  - specifying character range, 229
- if statement, 126
- ignoring case. *See* case of characters
- increment operator (++), 120
- indenting code, 126, 128, 129, 135, 247, 355–56
- int function, 119
- interpreter directive, 100, 123
- interrupting running programs, 188
- invisible files. *See* hidden files in Unix
  
- join function, 172–73
  
- karma, 392
- keyboard navigation
  - of command history, 53
  - of the command-line, 53
- keys function, 221–23
- killing Unix processes, 267
  
- last keyword, 199
- lc function, 146
- lcfirst function, 146
- length function, 144
- less command, 45, 69
- less than operator (<), 126
- less than or equal to operator (<=), 126
- lib pragma, 302
- libraries. *See* modules
- line endings, 289–90, 386–88
- line feed character, 289, 387
- Linux
  - comparison with Unix, 12
  - distributions, 13
  - how to install, 13, 14
- list directory contents. *See* ls command
- lists
  - assigning values to, 160–62, 167
  - list context, 160, 165, 239
  - list context vs. scalar context, 167
  - swapping values, 162
- logical operators, 129
  
- login shells, 90
- loops, 187–201
  - \$i as loop counter, 188
  - do loop, 195
  - for loop, 187–91, 193
  - foreach loop, 192–93
  - labelled loops, 201
  - last keyword, 199
  - loop control, 197–201
  - looping over arrays, 190–93, 194
  - looping over hashes, 221–24
  - looping over range, 193–94
  - while loop, 194–95
- ls command, 24–25, 40–43, 65, 84
- lt operator, 142
  
- main namespace, 137, 303, 365
- making directories, 47–48
- man command, 44–46
- matching operator, 149–51, 159
  - alternative delimiters, 158–59, 238
  - using the default variable (\$\_), 296
- matching patterns. *See* regular expressions
- mathematical operators, 117–20
- methods, 334
- mkdir command, 47–48
- modules, 300–05
  - from CPAN. *See* CPAN
  - how to use, 301, 341–42, 364–66
  - in object-oriented programming, 335
  - installing, 341
  - namespaces, 302
  - specifying location, 301–02, 342
  - use in abstracting code, 364–66
- modulo operator (%), 117
- moving directories, 59
- moving files, 55–58
- multidimensional data structures, 318, 325
- multiplication operator (\*), 117
- mv command, 55–58
- my keyword, 133
  
- namespaces, 302, 305, 365
- naming scripts, 382
- nano command, 77–79
- negation operator (!), 129
- new constructor, 333, 336

- newline (`\n`)
  - at ends of input files, 208
  - compared to other line ending characters, 289, 387–88
  - escaping special meaning, 112, 147
  - how to remove from strings, 212
  - length of, 144
  - overview, 107
- next keyword, 197–98
- noclobber setting, 278
- non-digit metacharacter (`\D`), 233
- non-whitespace metacharacter (`\S`), 233
- non-word boundary metacharacter (`\B`), 297
- not (logical operator), 129
- not equal operator (`!=`), 126
- not match operator (`!~`), 150
- null device, 284
- numerical comparison operators, 126
  
- object-oriented programming (OOP), 333–39
  - attributes, 333–34
  - bless keyword, 336
  - class variables, 335
  - classes, 333, 335–38
  - encapsulation, 334
  - methods, 334
  - new constructor, 333, 336
  - objects vs. classes, 333
  - use in abstracting code, 366–67
- OOP. *See* Object-oriented programming
- open function, 209, 210, 244
- operator precedence, 119
- or (logical operator), 129
- other peoples' data (OPD), 385–88
  
- package statement, 300, 302
- packages. *See* namespaces
- parent directories, 28, 30–31
- parentheses (`()`)
  - as delimiter for matching operator, 159
  - as grouping metacharacters, 227, 236, 238
  - as part of function names, 111
  - assigning an empty list, 175
  - in conditional statements, 126, 130
  - indicating list context, 160, 167
  - to indicate precedence, 120
- paths, 28, 96–99
  - absolute paths, 32, 96
  - absolute vs. relative paths, 32, 49
  - modifying `$PATH`, 97–99
  - relative paths, 32
  - viewing contents of `$PATH`, 97
- pattern matching. *See* regular expressions
- percent sign (`%`)
  - as hash prefix, 216
  - as modulo operator, 117
- perl command, 108, 123
- Perl one-liners, 387
- perldoc command, 381, 389–91
- permission denied errors, 351
- permissions. *See* file permissions
- pipe character (`|`)
  - in regular expressions, 227, 236
  - in Unix pipelines, 259, 285–88
  - reading from/writing to processes, 244–45
- plus sign (`+`)
  - addition operator, 117
  - wildcard character (Perl), 232
- pod (Plain Old Documentation), 380–81
- pop function, 170
- pos function, 297–98
- pound sign (`#`)
  - as alternative delimiter for matching operator, 238
  - as comment character, 82, *See* comments
- pragmas, 114, 133
- print function, 107, 111, 118
- print working directory, 28
- printenv command, 39
- process control, 263–68
  - interrupt signal, 264
  - interrupting running programs, 188
  - kill command, 268
  - process ID (PID), 266, 267
  - ps command, 266–67
  - top command, 265–66
- profile file. *See* .profile file
- program control in Perl
  - capturing Unix output, 241–45
  - checking whether programs ran successfully, 242
  - die function, 138, 139, 140, 178, 210
  - exit function, 139
  - overview, 138

- prompts, 23, 90
- ps command, 266–67
- push function, 169
- pwd command, 28
  
- question mark (?)
  - as non-greedy quantifier in Perl, 295
  - as wildcard character (Perl), 232, 296
  - as wildcard character (Unix), 58, 62
- quotation characters
  - as delimiters for matching operator, 299
  - in hash keys, 217
  - single vs. double, 112, 142, 147
  - unquoted string error, 115
  - when providing command-line arguments, 178
  - when using numbers, 117, 121
- qw function, 166, 171
- qx operator, 242
  
- rand function, 118
- range operator (..), 193–94
- read permission. *See* file permissions
- README files, 379
- recursive copying, 67
- recursive subroutines, 325
- redirecting input, output, and error, 276–84
  - appending output to existing files, 278
  - redirecting error, 283–84
  - redirecting input, 280
  - redirecting output, 276–79
  - with pipelines, 285
- redo keyword, 198
- ref function, 307
- references, 306–26
  - 2-dimensional arrays, 309–12
  - 2-dimensional data structures, 317
  - anonymous arrays, 308, 309
  - anonymous hashes, 314
  - dereferencing arrays, 307
  - dereferencing hashes, 314
  - dereferencing scalars, 320
  - garbage collection, 313, 326
  - multidimensional data structures, 318, 325
  - passing to subroutines, 320
  - testing whether a scalar is a reference, 307
  - to arrays, 306–08
  - to hashes, 314–16
  - to scalars, 319–20
  - viewing complex data structures, 323
- regular expressions, 225–40, 295–99
  - advanced metacharacters, 233–34, 297
  - capturing matching patterns, 236–40, 298
  - character classes, 228
  - character ranges, 229
  - determining position of matches, 297
  - how to match metacharacters, 230
  - matching at start or end of strings, 229–30
  - matching rule precedence, 296
  - matching word boundaries, 297
  - specifying alternative patterns, 227
  - using quantifiers, 231–33, 295–96
  - using variables in patterns, 235
  - with grep command, 276
  - with split function, 235
  - working with default variable (\$\_), 296
- relative paths, 32
- removing directories, 49
- removing files, 61–62
- renaming directories, 59
- renaming files, 56–57
- repetition operator (x), 143
- return keyword, 253
- reverse function, 185
- revision control, 382–84
- rm command, 74
- rmdir command, 49
- root directory, 26, 27
- root level, 28, 32, 40
- running scripts, 94, 97, 108
- running Unix programs in Perl. *See* system function
- run-time errors, 346
  
- Saccharomyces Genome Database (SGD), 260, 262, 273
- sanity checks, 388
- scalar context, 167, 179
- scalar function, 167
- scalar variables. *See* variables
- scope, 134–37
- scope operator (::), 305, 365
- scripting in Unix. *See* shell scripts
- sed command, 290–91



- semicolon (;), 107, 126, 128
- setenv command, 98
- shell scripts, 92, 96, 102
- shells, 36–37
  - associated configuration files, 86
  - list of common shells, 36
- shift function, 170, 195, 249
- sort command, 287
- sorting in Perl, 183–85
  - creating custom sort functions, 253
  - reverse sort, 185
  - sort function, 183, 184–85
  - using `<=>` and `cmp`, 185
  - using `$a` and `$b`, 184, 223
- source command, 80–82, 88
- space characters in file names, 47
- spaceship operator (`<=>`), 183
- Spinal Tap, 26
- splice function, 170, 174
- split function, 173, 205, 235
- SQL. *See* databases
- SQLite, 373–74. *See* databases
- square brackets []
  - array index syntax, 164
  - as alternative delimiter for binding operator, 159
  - in anonymous array syntax, 308
  - specifying a character class, 228
- standard error, 282–84
- standard input
  - in Perl, 206
  - in Unix, 279–80
- standard output
  - in Perl, 212
  - in Unix, 276
- STDERR. *See* standard error
- STDIN. *See* standard input
- STDOUT. *See* standard output
- Storable package, 372–73
- strict pragma, 133–34, 346
- strings, 146
  - ASCII values, 143, 183
  - comparing, 142–43
  - concatenation operator, 142
  - creating from array, 172
  - extracting substrings, 144–45
  - length of, 144
  - matching. *See* matching operator
  - repetition operator, 143
  - reversing, 185
  - splitting into array, 173
  - substituting, 152–54
- Structured Query Language (SQL), 373–74
- stupidity of users, 157, 350, 385
- subdirectories, 27
- subroutines, 246–54
  - @\_ array, 248–49
  - as functions in a module, 301
  - calling, 247
  - compared to functions, 247
  - in object-oriented programming, 336
  - passing arrays or hashes, 319
  - passing data to, 247–49
  - returning data from, 253
  - syntax, 246
  - use in abstracting code, 362
- substitution operator, 152–54, 158, 238
- substr function, 144–45
- subtraction operator (-), 117
- superheroes, 217, 228, 234, 236, 237, 269, 282, 283, 295, 297
- symbol tables, 137
- symbolic links, 41
- syntax error messages, 352
- syntax highlighting, 15, 346
- system function, 244
- tab (t), 148, 355
- tab-completion, 50–52, 60, 108
- tab-separated values. *See* TSV
- tail command, 274
- terminal applications, 20–21
- text editors
  - Emacs, 77
  - nano, 77
  - vi, 76
  - vim, 76
- text files
  - editing, 76–79
  - README files, 379
  - viewing, 69
  - vs. binary files, 385
  - working with, 68
- tilde (~), 34, 66

- top command, 265–66
- touch command, 55
- tr command, 279, 280, 289
- transliteration operator, 156, 157
- trinary operator, 130
- trust no-one, 385
- truth, evaluating in Perl, 129
- TSV (tab-separated values), 148, 174, 311, 369–70
  
- uc function, 146
- ucfirst function, 146
- undef function, 180
- undefined variable. *See* variables
- uninitialized values, 116
- uniq command, 287
- uniqueness
  - of FASTA headers, 260
  - of hash keys, 216
  - of lines in a GFF file, 262
- unless statement, 130
- unshift function, 170
- usage statements, 43, 329, 378
- use keyword, 114–16, 301
- use of uninitialized variable errors, 352
  
- values function, 221
- variables
  - assigning values, 110, 111
  - declaring, 110, 134, 180
  - defined variable, 180
  - defined vs. undefined, 180–81
  - differences between scalars, arrays, and hashes, 220
  - evaluating in scalar context, 166
  - in Perl, 110–13
  - in shell scripts, 102
  - inside regular expressions, 298
  - interpolation, 112, 118, 299
  - lexical variables, 135, 137
  - naming, 112–13, 357
  - scalars, 110, 164
  - scalars vs. arrays, 164
  - scope, 134–37
  - version control. *See* revision control
  - vertical bar (|). *See* pipe character
  - vi command, 76
  - viewing text files
    - with cat command, 83, 274
    - with head and tail commands, 274
    - with less command, 69
  - vim command, 76
  - virtualization, 13, 14
  
- warnings, 114–16, 133, 181, 184, 347
- wc command, 285
- which command, 72, 99
- while loop, 194–95
- whitespace
  - \s metacharacter, 233
  - tabs vs. spaces, 148
  - use in beautifying code, 113, 217, 358
  - use in split function, 205, 235
- wildcard characters, 58, 62, 230
- word boundary metacharacter (\b), 297
- word metacharacter (\w), 233
- working directory, 28
- write permission. *See* file permissions
  
- XML (eXtensible Markup Language), 370–71