Index

!, negation operator 63
%, remainder operator 52, 166
&&, short-circuit AND operator 63
&, address-of operator 24, 32, 94, 158, 167
&, implicit 158, 167
&^, AND-NOT operator 53
&^, bit-clear operator 53
' quote character 56
*, indirection operator 24, 32
++, increment statement 5, 37, 94
+, string concatenation operator 5, 65
+, unary operator 53
+=, -=, etc., assignment operator 5
-, unary operator 53
--, decrement statement 5, 37
... argument 139, 142
... array length 82
... parameter 91, 142, 143, 172
... path 292, 299
/*...*/ comment 5, 25
// comment 5, 25
:= short variable declaration 5, 31, 49
<<, left shift operator 54
==, comparison operator 40, 63
>>, right shift operator 54
^, bitwise complement operator 53
^, exclusive OR operator 53
_, blank identifier 7, 38, 95, 120, 126, 287
` backquote character 66
| in template 113
|, bitwise OR operator 166, 167
||, short-circuit OR operator 63
Abstract Syntax Notation One (ASN.1) 107
abstract type 24, 171
abstraction, premature 216, 316, 317
ad hoc polymorphism 211
address of local variable 32, 36
address of struct literal 103
addressable expression 159, 341
addressable value 32
address-of operator & 24, 32, 94, 158, 167
aggregate type 81, 99
Alef programming language xiii
algorithm
  breadth-first search 139, 239
  depth-first search 136
  Fibonacci 37, 218
  GCD 37
  insertion sort 101
  Lissajous 15
  slice rotation 86
  topological sort 136
aliasing, pointer 33
alignment 354
allocation
  heap 36
  memory 36, 71, 89, 169, 209, 322
  stack 36
anchor element, HTML 122
AND operator &&, short-circuit 63
AND-NOT operator &^ 53
animation, GIF 13
anonymous
  function 22, 135, 236
  function, defer 146
  function, recursive 137
  struct field 104, 105, 106, 162
API
  encoding 213, 340
  error 127, 152
  package 284, 296, 311, 333, 352
  runtime 324
  SQL 211
  system call 196
  template 115
  token-based decoder 213, 215, 347
APL programming language xiii
append built-in function 88, 90, 91
appendInt example 88
argument
  ... 139, 142
  command-line 4, 18, 33, 43, 179, 180, 290, 313
  function 119
  pointer 33, 83
  slice 86
arithmetic expression evaluator 197
array
  comparison 83
  length, ... 82
  literal 82, 84
  type 81
  underlying 84, 88, 91, 187
  zero value 82
ASCII 56, 64, 66, 67, 305
ASN.1 (Abstract Syntax Notation One) 107
assembly line, cake 234
assertion
  function 316
  interface type 208, 210

  - test 306
  - type 205, 211
- assignability 38, 175
- assignability, interface 175
- assignment
  - implicit 38
  - multiple-value 37
  - operator +=, -=, etc. 5
  - operators 36, 52
  - statement 5, 7, 36, 52, 94, 173
  - tuple 31, 37
- associativity, operator 52
- atomic operation 264
- attack, HTML injection 115
- attack, SQL injection 211
- autoescape example 117

- back-door, package 315
- back-off, exponential 130
- backquote character, ` 66
- bank example package 258, 261, 263
- bare return 126
- basename example 72
- behavior, undefined 260
- Benchmark function 302, 321
- bidirectional to unidirectional channel conversion 231
- binary
  - operators, table of 52
  - semaphore 262
  - tree 102
- bit vector 165
- bit-clear operator &^ 53
- bit-set data type 77
- bitwise
  - complement operator ^ 53
  - operators, table of 53
  - OR operator | 166, 167
- black-box test 310
- blank identifier _ 7, 38, 95, 120, 126, 287
- blank import 287
- block
  - file 46
  - lexical 46, 120, 135, 141, 212
  - local 46
  - package 46
  - universe 46
- blocking profile 324
- Blog, Go xvi, 326
- boiling example 29
- bool type 63
- boolean
  - constant, false 63
  - constant, true 63
  - zero value 30
- breadthFirst function 139
- breadth-first search algorithm 139, 239
- break statement 24, 46
- break statement, labeled 249
- brittle test 317
- broadcast 251, 254, 276
- Brooks, Fred xiv
- btoi function 64
- buffered channel 226, 231
- bufio package 9
- bufio.NewReader function 98
- bufio.NewScanner function 9
- (*bufio.Reader).ReadRune method 98
- bufio.Scanner type 9
- (*bufio.Scanner).Err method 97
- (*bufio.Scanner).Scan method 9
- (*bufio.Scanner).Split method 99
- bufio.ScanWords function 99
- +build comments 296
- build constraints 296
- build tags 296
- building packages 293
- built-in function
  - append 88, 90, 91
  - cap 84, 232
  - close 226, 228, 251
  - complex 61
  - copy 89
  - delete 94
  - imag 61
  - len 4, 54, 64, 65, 81, 84, 233
  - make 9, 18, 88, 94, 225
  - new 34
  - panic 148, 149
  - real 61
  - recover 152
- built-in interface, error 196
- built-in type, error 11, 128, 149, 196
- byte slice to string conversion 73
- byte type 52
- ByteCounter example 173
- bytes package 71, 73
- bytes.Buffer type 74, 169, 172, 185
- (*bytes.Buffer).Grow method 169
- (*bytes.Buffer).WriteByte method 74
- (*bytes.Buffer).WriteRune method 74
- (*bytes.Buffer).WriteString method 74
- bytes.Equal function 86
- bzip C code 362
- bzip example package 363
- bzipper example 365

- C++ programming language xiv, xv, 361
- C programming language xii, xv, 1, 6, 52, 260, 361
- cache, concurrent non-blocking 272
- cache, non-blocking 275
- cake assembly line 234
- call
  - by reference 83
  - by value 83, 120, 158
  - interface method 182
  - ok value from function 128
- calling C from Go 361
- camel case 28
- cancellation 251, 252
- cancellation of HTTP request 253
- cap built-in function 84, 232
- capacity, channel 226, 232, 233
- capacity, slice 88, 89
- capturing iteration variable 140
- capturing loop variable 141, 236, 240
- case in type switch 212
- case, select 245
- Celsius type 39
- CelsiusFlag function 181
- cf example 43
- cgo tool 361, 362
- <-ch, channel receive 18, 225, 232
- ch<-, channel send 18, 225, 232
- chaining, method 114
- chan type 225
- channel
  - buffered 226, 231
  - capacity 226, 232, 233
  - close 228, 251
  - closing a 225
  - communication 225, 245
  - comparison 225
  - conversion, bidirectional to unidirectional 231
  - draining a 229, 252
  - make 18, 225
  - nil 246, 249
  - polling 246
  - range over 229
  - receive <-ch 18, 225, 232
  - receive, non-blocking 246
  - receive, ok value from 229
  - send ch<- 18, 225, 232
  - synchronous 226
  - type 18
  - type <-chan T, receive-only 230
  - type chan<- T, send-only 230
  - type, unidirectional 230, 231
  - unbuffered 226
  - zero value 225, 246
- character conversion 71
- character test 71
- charcount example 98
- chat example 254

chat server 253
CheckQuota function 312, 313
client, email 312
client, SMTP 312
clock example 220, 222
clock server, concurrent 219
close built-in function 226, 228, 251
close, channel 228, 251
closer goroutine 238, 250
closing a channel 225
closure, lexical 136
cmplx.Sqrt function 61
code
  format 3, 6, 9, 48
  point, Unicode 67
  production 301
ColoredPoint example 161
comma example 73
command, testing a 308
command-line argument 4, 18, 33, 43, 179, 180, 290, 313
comment
  /*...*/ 5, 25
  // 5, 25
  doc 42, 296
  // Output 326
comments, +build 296
communicating sequential processes (CSP) xiii, 217
communication, channel 225, 245
comparability 9, 38, 40, 53, 86, 93, 97, 104
comparison
  array 83
  channel 225
  function 133
  interface 184
  map 96
  operator == 40, 63
  operators 40, 93
  operators, table of 53
  slice 87
  string 65
  struct 104
compilation, separate 284
complement operator ^, bitwise 53
complex built-in function 61
complex type 61
composite literal 14
composite type xv, 14, 81
composition, parallel 224
composition, type xv, 107, 162, 189
compress/bzip2 package 361
compression 361
conceptual integrity xiv
concrete type 24, 171, 211, 214
concurrency 17, 217, 257
  excessive 241, 242
  safe 275
  safety 256, 257, 272, 365
  with shared variables 257
concurrent
  clock server 219
  directory traversal 247
  echo server 222
  non-blocking cache 272
  web crawler 239
confinement, serial 262
confinement, variable 261
consistency, sequential 268, 269
const declaration 14, 75
constant
  false boolean 63
  generator, iota xiii, 77
  time.Minute 76
  time.Second 164
  true boolean 63
  types, untyped 78
constants, precision of 78
constraints, build 296
contention, lock 267, 272
context switch 280
continue statement 24, 46
continue statement, labeled 249
contracts, interfaces as 171
control flow 46
conversion
  bidirectional to unidirectional channel 231
  byte slice to string 73
  character 71
  implicit 79
  narrowing 40, 55
  numeric 79
  operation 40, 55, 64, 71, 78, 79, 173, 187, 194, 208, 231, 353, 358
  rune slice to string 71
  rune to string 71
  string 71
  string to byte slice 40, 73
  string to rune slice 71, 88
  unsafe.Pointer 356
copy built-in function 89
countdown example 244, 245, 246
counting semaphore 241
coverage, statement 318, 320
coverage, test 318
coverage_test example 319
CPU profile 324
crawl example 240, 242, 243
crawler, concurrent web 239
crawler, web 119
critical section 263, 270, 275
cross-compilation 295
cryptography 55, 83, 121, 325
crypto/sha256 package 83
customSort example 190
cyclic data structure 337
cyclic test dependency 314

data
  race 259, 267, 275
  structure, cyclic 337
  structure, recursive 101, 102, 107
  type, bit-set 77
database driver, MySQL 284
database/sql package 211, 288
daysAgo function 114
deadbeef 55, 80
deadlock 233, 240, 265
declaration
  const 14, 75
  func 3, 29, 119
  import 3, 28, 42, 284, 285, 362
  method 40, 155
  package 2, 28, 41, 285
  package-level 28
  scope 45, 137
  shadowing 46, 49, 206, 212
  short variable 5, 7, 30, 31
  statement, short variable 7
  struct 99
  type 39
  var 5, 30
declarations, order of 48
decode example, S-expression 347
decoder API, token-based 213, 215, 347
decoding, S-expression 344
decoding, XML 213
decrement statement -- 5, 37
dedup example 97
deep equivalence 87, 317, 358
default case in select 246
default case in switch 23
default case in type switch 212
defer anonymous function 146
defer example 150, 151
defer statement 144, 150, 264
deferred function call 144
delete built-in function 94
depth-first search algorithm 136
dereference, implicit 159
diagram
  helloworld substring 69
  pipeline 228
  slice capacity growth 90
  slice of months 84
  string sharing 65
  struct hole 355
  thumbnail sequence 238
digital artifact example 178
Dijkstra, Edsger 318
Dilbert 100
directed acyclic graph 136, 284
directory traversal, concurrent 247

discriminated union 211, 213, 214
Display function 333
display example 333
display function 334
displaying methods of a type 351
Distance function 156
doc comment 42, 296
doc.go doc comment file 42, 296
documentation, package 296
domain name, import path 284
dot . in template 113
downloading packages 292
Dr. Strangelove 336
draining a channel 229, 252
du example 247, 249, 250
dup example 9, 11, 12
duplicate suppression 276
dynamic dispatch 183
dynamic type, interface 181

echo example 5, 7, 34, 309
echo test 309
echo server, concurrent 222
echo_test.go 310
effective tests, writing 316, 317
email client 312
embarrassingly parallel 235
embedded struct field 161
embedding, interface 174
embedding, struct 104, 161
Employee struct 100
empty
  interface type 176
  select statement 245
  string 5, 7, 30
  struct 102
encapsulation 168, 284
encoding API 213, 340
encoding, S-expression 338
encoding/json package 107
encoding/xml package 107, 213
end of file (EOF) 131
enum 77
environment variable
  GOARCH 292, 295
  GOMAXPROCS 281, 321
  GOOS 292, 295
  GOPATH xvi, 291, 295
  GOROOT 292
equal function 87, 96
equality, pointer 32
equivalence, deep 87, 317, 358
error built-in interface 196
error built-in type 11, 128, 149, 196
error API 127, 152
error.Error method 196
errorf function 143
error-handling strategies 128, 152, 310, 316
errors package 196
errors.New function 196
escape
  hexadecimal 66
  HTML 116
  octal 66
  sequence 10
  sequences, table of 66
  Unicode 68, 107
  URL 111
escaping variables 36
eval example 198
event multiplexing 244
events 227, 244
Example function 302, 326
example
  autoescape 117
  basename 72
  boiling 29
  ByteCounter 173
  bzipper 365
  cf 43
  charcount 98
  chat 254
  clock 220, 222
  ColoredPoint 161
  comma 73
  countdown 244, 245, 246
  coverage_test 319
  crawl 240, 242, 243
  customSort 190
  dedup 97
  defer 150, 151
  digital artifact 178
  display 333
  du 247, 249, 250
  dup 9, 11, 12
  echo 5, 7, 34, 309
  eval 198
  fetch 16, 148
  fetchall 18
  findlinks 122, 125, 139
  ftoc 29
  github 110, 111
  graph 99
  helloworld 1, 2
  http 192, 194, 195
  intset 166
  issues 112
  issueshtml 115
  issuesreport 114
  jpeg 287
  lissajous 14, 22, 35
  mandelbrot 62
  memo 275, 276, 277, 278, 279
  methods 351
  movie 108, 110
  netcat 221, 223, 227
  netflag 78
  nonempty 92
  outline 123, 133
  package, bank 258, 261, 263
  package, bzip 363
  package, format 332
  package, geometry 156
  package, http 192
  package, links 138
  package, memo 273
  package, params 348
  package, storage 312, 313
  package, tempconv 42
  package, thumbnail 235
  palindrome 303, 305, 308
  params 348
  Parse 152
  pipeline 228, 230, 231
  playlist 187
  rev 86
  reverb 223, 224
  server 19, 21
  sexpr 340
  S-expression decode 347
  sha256 83
  sleep 179
  spinner 218
  squares 135
  sum 142
  surface 59, 203
  tempconv 39, 180, 289
  temperature conversion 29
  tempflag 181
  test of word 303
  thumbnail 236, 237, 238
  title 153
  topoSort 136
  trace 146
  treesort 102
  urlvalues 160
  wait 130
  word 303, 305, 308
  xmlselect 215
  appendInt 88
exception 128, 149
excessive concurrency 241, 242
exclusion, mutual 262, 267
exclusive lock 263, 266, 270
exclusive OR operator ^ 53
exponential back-off 130
export of struct field 101, 106, 109, 110, 168
export_test.go file 315
Expr.Check method 202
expression
  addressable 159, 341
  evaluator 197
  method 164
  receive 225
Expr.Eval method 199

extending a slice 86
Extensible Markup Language (XML) 107
external test package 285, 314

Fahrenheit type 39
failure message, test 306
fallthrough statement 23, 212
false boolean constant 63
fetch example 16, 148
fetchall example 18
fib function 37, 218
Fibonacci algorithm 37, 218
field
  anonymous struct 104, 105, 106, 162
  embedded struct 161
  export of struct 101, 106, 109, 110, 168
  order, struct 101, 355
  selector 156
  struct 15, 99
  tag, omitempty 109
  tag, struct 109, 348
figure
  Lissajous 13
  Mandelbrot 63
  3-D surface 58, 203
File Transfer Protocol (FTP) 222
file
  block 46
  export_test.go 315
  name, Microsoft Windows 72
  name, POSIX 72
  _test.go 285, 302, 303
findlinks example 122, 125, 139
fixed-size stack 124
flag package 33, 179
flag
  go tool -bench 321
  go tool -benchmem 322
  go tool -covermode 319
  go tool -coverprofile 319
  go tool -cpuprofile 324
  go tool -nodecount 325
  go tool -text 325
  go tool -web 326
  godoc -analysis 176
  go list -f 315
  go -race 271
  go test -race 274
  go test -run 305
  go test -v 304
flag.Args function 34
flag.Bool function 34
flag.Duration function 179
flag.Parse function 34
flag.String function 34
flag.Value interface 179, 180
floating-point
  number 56
  precision 56, 57, 63, 78
  truncation 40, 55
fmt package 2
fmt.Errorf function 129, 196
fmt.Fprintf function 172
fmt.Printf function 10
fmt.Println function 2
fmt.Scanf function 75
fmt.Sscanf function 180
fmt.Stringer interface 180, 210
for scope 47
for statement 6
forEachNode function 133
foreign-function interface (FFI) 361
format, code 3, 6, 9, 48
format example package 332
formatAtom function 332
framework, web 193
ftoc example 29
func declaration 3, 29, 119
function
  anonymous 22, 135, 236
  append built-in 88, 90, 91
  argument 119
  assertion 316
  Benchmark 302, 321
  body, missing 121
  breadthFirst 139
  btoi 64
  bufio.NewReader 98
  bufio.NewScanner 9
  bufio.ScanWords 99
  bytes.Equal 86
  call, deferred 144
  call, ok value from 128
  cap built-in 84, 232
  CelsiusFlag 181
  CheckQuota 312, 313
  close built-in 226, 228, 251
  cmplx.Sqrt 61
  comparison 133
  complex built-in 61
  copy built-in 89
  daysAgo 114
  delete built-in 94
  Display 333
  display 334
  Distance 156
  equal 87, 96
  errorf 143
  errors.New 196
  Example 302, 326
  fib 37, 218
  flag.Args 34
  flag.Bool 34
  flag.Duration 179
  flag.Parse 34
  flag.String 34
  fmt.Errorf 129, 196
  fmt.Fprintf 172
  fmt.Printf 10
  fmt.Println 2
  fmt.Scanf 75
  fmt.Sscanf 180
  forEachNode 133
  formatAtom 332
  gcd 37
  handler 19, 21, 152, 191, 194, 195, 348
  html.Parse 121, 125
  http.DefaultServeMux 195
  http.Error 193
  http.Get 16, 18
  http.Handle 195
  http.HandleFunc 19, 22, 195
  http.ListenAndServe 19, 191
  http.NewRequest 253
  http.ServeMux 193
  hypot 120
  imag built-in 61
  image.Decode 288
  image.RegisterFormat 288
  incr 33
  init 44, 49
  intsToString 74
  io.Copy 17, 18
  ioutil.ReadAll 16, 272
  ioutil.ReadDir 247
  ioutil.ReadFile 12, 145
  io.WriteString 209
  itob 64
  json.Marshal 108
  json.MarshalIndent 108
  json.NewDecoder 111
  json.NewEncoder 111
  json.Unmarshal 110, 114
  len built-in 4, 54, 64, 65, 81, 84, 233
  links.Extract 138
  literal 22, 135, 227
  log.Fatalf 49, 130
  main 2, 310
  make built-in 9, 18, 88, 94, 225
  math.Hypot 156
  math.Inf 57
  math.IsInf 57
  math.IsNaN 57
  math.NaN 57
  multi-valued 11, 30, 37, 96, 125, 126
  mustCopy 221
  net.Dial 220
  net.Listen 220
  new built-in 34
  nil 132
  os.Close 11

os.Exit 16, 34, 48
os.Getwd 48
os.IsExist 207
os.IsNotExist 207
os.IsPermission 207
os.Open 11
os.Stat 247
panic built-in 148, 149
parameter 119
params.Unpack 349
png.Encode 62
PopCount 45
real built-in 61
recover built-in 152
recursive anonymous 137
reflect.TypeOf 330
reflect.ValueOf 331, 337
reflect.Zero 345
regexp.Compile 149
regexp.MustCompile 149
result list 119
runtime.Stack 151
SearchIssues 111
sexpr.Marshal 340
sexpr.readList 347
sexpr.Unmarshal 347
signature 120
sort.Float64s 191
sort.Ints 191
sort.IntsAreSorted 191
sort.Reverse 189
sort.Strings 95, 137, 191
Sprint 330
sqlQuote 211, 212
strconv.Atoi 22, 75
strconv.FormatInt 75
strconv.Itoa 75
strconv.ParseInt 75
strconv.ParseUint 75
strings.Contains 69
strings.HasPrefix 69
strings.HasSuffix 69
strings.Index 289
strings.Join 7, 12
strings.Map 133
strings.NewReader 289
strings.NewReplacer 289
strings.Split 12
strings.ToLower 72
strings.ToUpper 72
template.Must 114
template.New 114
Test 302
time.After 245
time.AfterFunc 164
time.Now 220
time.Parse 220
time.Since 114
time.Tick 244, 246
title 144, 145
type 119, 120
unicode.IsDigit 71
unicode.IsLetter 71
unicode.IsLower 71
unicode.IsSpace 93
unicode.IsUpper 71
unsafe.AlignOf 355
unsafe.Offsetof 355
unsafe.Sizeof 354
url.QueryEscape 111
utf8.DecodeRuneInString 69
utf8.RuneCountInString 69
value 132
variadic 142, 172
visit 122
WaitForServer 130
walkDir 247
zero value 132

garbage collection xi, xiii, 7, 35, 230, 353, 357
garbage collector, moving 357
GCD algorithm 37
gcd function 37
geometry example package 156
geometry.Point.Distance method 156
getter method 169
GIF animation 13
GitHub issue tracker 110
github example 110, 111
Go
  Playground xvi, 326
  Blog xvi, 326
  issue 110, 112, 358
go tool 2, 42, 44, 290
go tool -bench flag 321
go tool -benchmem flag 322
go tool -covermode flag 319
go tool -coverprofile flag 319
go tool -cpuprofile flag 324
go tool -nodecount flag 325
go tool pprof 325
go tool -text flag 325
go tool -web flag 326
go tool cover 318, 319
go doc tool 25
go statement 18, 218
GOARCH environment variable 292, 295
go build 2, 286, 293, 294
go doc 296
godoc -analysis flag 176
godoc tool xvi, 25, 297, 326
go env 292
gofmt tool 3, 4, 44, 286
go get xvi, 2, 292, 293
go help 290
goimports tool 3, 44, 286
go install 295
golang.org/x/net/html package 122
golint tool 292
go list 298, 315
go list -f flag 315
GOMAXPROCS environment variable 281, 321
GOOS environment variable 292, 295
GOPATH environment variable xvi, 291, 295
gopl.io repository xvi
go -race flag 271
GOROOT environment variable 292
goroutine 18, 217, 233, 235
  closer 238, 250
  identity 282
  leak 233, 236, 246
  monitor 261, 277
  multiplexing 281
  vs. OS thread 280
go run 2, 294
go test 301, 302, 304
go test -race flag 274
go test -run flag 305
go test -v flag 304
goto statement 24
graph example 99
GraphViz 326
Griesemer, Robert xi
growth, stack 124, 280, 358
guarding mutex 263

half-open interval 4
handler function 19, 21, 152, 191, 194, 195, 348
"happens before" relation 226, 257, 261, 277
"has a" relationship 162
hash table 9, 93
Haskell programming language xiv
heap
  allocation 36
  profile 324
  variable 36
helloworld example 1, 2
helloworld substring diagram 69
hexadecimal escape 66
hexadecimal literal 55
hidden pointer 357
Hoare, Tony xiii
hole, struct 354
HTML
  anchor element 122
  escape 116
  injection attack 115
  metacharacter 116
  parser 121

html.Parse function 121, 125
html/template package 113, 115
HTTP
  GET request 21, 127, 272, 348
  POST request 348
  request, cancellation of 253
  request multiplexer 193
http example 192, 194, 195
http example package 192
(*http.Client).Do method 253
http.DefaultClient variable 253
http.DefaultServeMux function 195
http.Error function 193
http.Get function 16, 18
http.Handle function 195
http.HandleFunc function 19, 22, 195
http.Handler interface 191, 193
http.HandlerFunc type 194, 203
http.ListenAndServe function 19, 191
http.NewRequest function 253
http.Request type 21, 253
(*http.Request).ParseForm method 22, 348
http.ResponseWriter type 19, 22, 191, 193
http.ServeMux function 193
hypot function 120

identifier _, blank 7, 38, 95, 120, 126, 287
identifier, qualified 41, 43
identity, goroutine 282
IEEE 754 standard 56, 57
if, initialization statement in 22, 206
if-else scope 47
if-else statement 9, 22, 47
imag built-in function 61
image manipulation 121
image package 62, 287
image/color package 14
image.Decode function 288
image/png package 288
image.RegisterFormat function 288
imaginary literal 61
immutability 261
immutability, string 65, 73
implementation with slice, stack 92, 215
implicit
  & 158, 167
  assignment 38
  conversion 79
  dereference 159
import declaration 3, 28, 42, 284, 285, 362
import
  blank 287
  path 284
  path domain name 284
  renaming 286
incr function 33
increment statement ++ 5, 37, 94
index operation, string 64
indirection operator * 24, 32
infinite loop 6, 120, 228
information hiding 168, 284
init function 44, 49
initialization
  lazy 268
  package 44
  statement in if 22, 206
  statement in switch 24
initializer list 30
injection attack, HTML 115
injection attack, SQL 211
in-place slice techniques 91
insertion sort algorithm 101
int type 52
integer
  literal 55
  overflow 53, 113
  signed 52, 54
  unsigned 52, 54
integration test 314
interface
  assignability 175
  comparison 184
  dynamic type 181
  embedding 174
  error built-in 196
  flag.Value 179, 180
  fmt.Stringer 180, 210
  http.Handler 191, 193
  io.Closer 174
  io.Reader 174
  io.Writer 15, 22, 172, 174, 186, 208, 209, 309
  JSON 110
  method call 182
  nil 182
  pitfall 184
  ReadWriteCloser 174
  ReadWriter 174
  satisfaction 171, 175
  sort.Interface 186
  type 171, 174
interface{} type 143, 176, 331
interface
  type assertion 208, 210
  type, empty 176
  value 181
  with nil pointer 184
  zero value 182
interfaces as contracts 171
internal package 298
intset example 166
intsToString function 74
invariants 159, 169, 170, 265, 284, 311, 352
io package 174
io.Closer interface 174
io.Copy function 17, 18
io.Discard stream 22
io.Discard variable 18
io.EOF variable 132
io/ioutil package 16, 145
io.Reader interface 174
iota constant generator xiii, 77
ioutil.ReadAll function 16, 272
ioutil.ReadDir function 247
ioutil.ReadFile function 12, 145
io.Writer interface 15, 22, 172, 174, 186, 208, 209, 309
io.WriteString function 209
“is a” relationship 162, 175
issue, Go 110, 112, 358
issue tracker, GitHub 110
issues example 112
issueshtml example 115
issuesreport example 114
iteration order, map 95
iteration variable, capturing 140
itob function 64

Java programming language xv
JavaScript Object Notation (JSON) 107, 338
JavaScript programming language xv, 107
jpeg example 287
JSON
  interface 110
  interface, Open Movie Database 113
  interface, xkcd 113
  marshaling 108
  unmarshaling 110
json.Decoder type 111
json.Encoder type 111
json.Marshal function 108
json.MarshalIndent function 108
json.NewDecoder function 111
json.NewEncoder function 111
json.Unmarshal function 110, 114

keyword, type 212
keywords, table of 27
Knuth, Donald 323

label scope 46
label, statement 46
labeled

  `break` statement 249
  `continue` statement 249
  statement 46
layout, memory 354, 355
lazy initialization 268
leak, goroutine 233, 236, 246
left shift operator `<<` 54
`len` built-in function 4, 54, 64, 65, 81, 84, 233
lexical block 46, 120, 135, 141, 212
lexical closure 136
lifetime, variable 35, 46, 135
`links` example package 138
`links.Extract` function 138
Lisp programming language 338
Lissajous algorithm 15
Lissajous figure 13
`lissajous` example 14, 22, 35
list, initializer 30
literal
  array 82, 84
  composite 14
  function 22, 135, 227
  hexadecimal 55
  imaginary 61
  integer 55
  map 94
  octal 55
  raw string 66
  `rune` 56
  slice 38, 86
  string 65
  struct 15, 102, 106
local
  block 46
  variable 29, 141
  variable, address of 32, 36
  variable scope 135
locating packages 291
lock
  contention 267, 272
  exclusive 263, 266, 270
  mutex 102, 263, 264, 324
  non-reentrant 265
  readers 266
  shared 266
  writer 266
`log` package 49, 130, 170
`log.Fatalf` function 49, 130
lookup `m[key]`, map 94
lookup, `ok` value from map 96
loop
  infinite 6, 120, 228
  `range` 6, 9
  variable, capturing 141, 236, 240
  variable scope 141, 236
  while 6

`main` function 2, 310
`main`, package 2, 285, 310
`make` built-in function 9, 18, 88, 94, 225
`make` channel 18, 225
`make` map 9, 18, 94
`make` slice 88, 322
Mandelbrot figure 63
Mandelbrot set 61
`mandelbrot` example 62
map
  as set 96, 202
  comparison 96
  element, nonexistent 94, 95
  iteration order 95
  literal 94
  lookup `m[key]` 94
  lookup, `ok` value from 96
  `make` 9, 18, 94
  `nil` 95
  `range` over 94
  type 9, 93
  with slice key 97
  zero value 95
marshaling JSON 108
`math` package 14, 56
`math/big` package 63
`math/cmplx` package 61
`math.Hypot` function 156
`math.Inf` function 57
`math.IsInf` function 57
`math.IsNaN` function 57
`math.NaN` function 57
`math/rand` package 285, 308
`memo` example 275, 276, 277, 278, 279
`memo` example package 273
memoization 272
memory allocation 36, 71, 89, 169, 209, 322
memory layout 354, 355
metacharacter, HTML 116
method
  `(*bufio.Reader).ReadRune` 98
  `(*bufio.Scanner).Err` 97
  `(*bufio.Scanner).Scan` 9
  `(*bufio.Scanner).Split` 99
  `(*bytes.Buffer).Grow` 169
  `(*bytes.Buffer).WriteByte` 74
  `(*bytes.Buffer).WriteRune` 74
  `(*bytes.Buffer).WriteString` 74
  call, interface 182
  chaining 114
  declaration 40, 155
  `error.Error` 196
  `Expr.Check` 202
  expression 164
  `Expr.Eval` 199
  `geometry.Point.Distance` 156
  getter 169
  `(*http.Client).Do` 253
  `(*http.Request).ParseForm` 22, 348
  name 156
  `net.Conn.Close` 220
  `net.Listener.Accept` 220
  `(*os.File).Write` 183
  `path.Distance` 157
  promotion 161
  receiver name 157
  receiver parameter 156
  receiver type 157
  `reflect.Type.Field` 348
  `reflect.Value.Addr` 342
  `reflect.Value.CanAddr` 342
  `reflect.Value.Interface` 331, 342
  `reflect.Value.Kind` 332
  selector 156
  setter 169
  `String` 40, 166, 329
  `(*sync.Mutex).Lock` 21, 146, 263
  `(*sync.Mutex).Unlock` 21, 146, 263
  `(*sync.Once).Do` 270
  `(*sync.RWMutex).RLock` 266
  `(*sync.RWMutex).RUnlock` 266
  `(*sync.WaitGroup).Add` 238
  `(*sync.WaitGroup).Done` 238
  `template.Funcs` 114
  `template.Parse` 114
  `(*testing.T).Errorf` 200, 304, 306
  `(*testing.T).Fatal` 306
  `time.Time.Format` 220
  value 164
  `(*xml.Decoder).Token` 213
`methods` example 351
methods of a type, displaying 351
Microsoft Windows file name 72
missing function body 121
`m[key]`, map lookup 94
mobile platforms 121
Modula-2 programming language xiii
modularity 283
monitor 264, 275
monitor goroutine 261, 277
`movie` example 108, 110
moving garbage collector 357
multimap 160, 193
multiple-value assignment 37
multiplexer, HTTP request 193
multiplexing, event 244
multiplexing, goroutine 281
multithreading, shared-memory 217, 257
multi-valued function 11, 30, 37, 96,

  125, 126
`mustCopy` function 221
mutex 145, 163, 256, 269
  guarding 263
  lock 102, 263, 264, 324
  read/write 266, 267
mutual exclusion 262, 267
MySQL database driver 284

name
  method 156
  method receiver 157
  package 28, 43
  parameter 120
  space 41, 156, 283
named
  result 120, 126
  result zero value 120, 127
  type 24, 39, 40, 105, 157
naming convention 28, 169, 174, 289
naming, package 289
NaN (not a number) 57, 93
narrowing conversion 40, 55
negation operator `!` 63
`net` package 219
`netcat` example 221, 223, 227
`net.Conn` type 220
`net.Conn.Close` method 220
`net.Dial` function 220
`netflag` example 78
`net/http` package 16, 191
`net.Listen` function 220
`net.Listener` type 220
`net.Listener.Accept` method 220
`net/smtp` package 312
`net/url` package 160
networking 121, 219
`new` built-in function 34
`new`, redefining 35
nil
  channel 246, 249
  function 132
  interface 182
  map 95
  pointer 32
  pointer, interface with 184
  receiver 159, 185
  slice 87
non-blocking
  cache 275
  cache, concurrent 272
  channel receive 246
  `select` 246
`nonempty` example 92
nonexistent map element 94, 95
non-reentrant lock 265
non-standard package 121
number, floating-point 56
number zero value 5, 30
numeric
  conversion 79
  precision 55, 78
  type 51

Oberon programming language xiii
object 156
object-oriented programming (OOP) 155, 168
octal escape 66
octal literal 55
`ok` value 37
`ok` value from channel receive 229
`ok` value from function call 128
`ok` value from map lookup 96
`ok` value from type assertion 206
`omitempty` field tag 109
Open Movie Database JSON interface 113
operation, atomic 264
operation, conversion 40, 55, 64, 71, 78, 79, 173, 187, 194, 208, 231, 353, 358
operator
  `+=`, `-=`, etc., assignment 5
  `&`, address-of 24, 32, 94, 158, 167
  `&^`, AND-NOT 53
  `&^`, bit-clear 53
  `^`, bitwise complement 53
  `|`, bitwise OR 166, 167
  `==`, comparison 40, 63
  `^`, exclusive OR 53
  `*`, indirection 24, 32
  `<<`, left shift 54
  `!`, negation 63
  `%`, remainder 52, 166
  `>>`, right shift 54
  `&&`, short-circuit AND 63
  `||`, short-circuit OR 63
  `+`, string concatenation 5, 65
  `-`, unary 53
  `+`, unary 53
  associativity 52
  precedence 52, 63
  `s[i:j]`, slice 84, 86
  `s[i:j]`, substring 65, 86
operators
  assignment 36, 52
  comparison 40, 93
  table of binary 52
  table of bitwise 53
  table of comparison 53
optimization 264, 321, 323
optimization, premature 324
OR operator `||`, short-circuit 63
order of declarations 48
order, struct field 101, 355
organization, workspace 291
OS thread vs. goroutine 280
`os` package 4, 206
`os.Args` variable 4
`os.Close` function 11
`os.Exit` function 16, 34, 48
`*os.File` type 11, 13, 172, 175, 185, 336
`os.FileInfo` type 247
`(*os.File).Write` method 183
`os.Getwd` function 48
`os.IsExist` function 207
`os.IsNotExist` function 207
`os.IsPermission` function 207
`os.LinkError` type 207
`os.Open` function 11
`os.PathError` type 207
`os.Stat` function 247
`outline` example 123, 133
`// Output` comment 326
overflow, integer 53, 113
overflow, stack 124

`package` declaration 2, 28, 41, 285
package
  API 284, 296, 311, 333, 352
  back-door 315
  `bank` example 258, 261, 263
  block 46
  `bufio` 9
  `bytes` 71, 73
  `bzip` example 363
  `compress/bzip2` 361
  `crypto/sha256` 83
  `database/sql` 211, 288
  documentation 296
  `encoding/json` 107
  `encoding/xml` 107, 213
  `errors` 196
  external test 285, 314
  `flag` 33, 179
  `fmt` 2
  `format` example 332
  `geometry` example 156
  `golang.org/x/net/html` 122
  `html/template` 113, 115
  `http` example 192
  `image` 62, 287
  `image/color` 14
  `image/png` 288
  initialization 44
  `internal` 298
  `io` 174
  `io/ioutil` 16, 145
  `links` example 138
  `log` 49, 130, 170
  `main` 2, 285, 310
  `math` 14, 56
  `math/big` 63
  `math/cmplx` 61

  math/rand 285, 308
  memo example 273
  name 28, 43
  naming 289
  net 219
  net/http 16, 191
  net/smtp 312
  net/url 160
  non-standard 121
  os 4, 206
  params example 348
  path 72
  path/filepath 72
  reflect 330
  regexp 149
  runtime 151
  sort 95, 186, 189
  storage example 312, 313
  strconv 22, 71, 75
  strings 7, 71, 72, 289
  sync 237, 263
  syscall 196, 208
  tempconv example 42
  testing 285, 302
  text/scanner 344
  text/tabwriter 188
  text/template 113, 300
  thumbnail example 235
  time 18, 77, 183
  unicode 71
  unicode/utf8 69
  unsafe 354
package-level declaration 28
packages
  building 293
  downloading 292
  locating 291
  querying 298
palindrome 191
palindrome example 303, 305, 308
panic 64, 152, 253
panic built-in function 148, 149
paradoxical race 267
parallel composition 224
parallel, embarrassingly 235
parallelism 217
parameter
  ... 91, 142, 143, 172
  function 119
  method receiver 156
  name 120
  passing 120
  unused 120
params example 348
params example package 348
params.Unpack function 349
parentheses 4, 6, 9, 52, 63, 119, 146, 158, 285, 335, 345
Parse example 152
parser, HTML 121
Pascal programming language xiii
path, ... 292, 299
path package 72
path.Distance method 157
path/filepath package 72
Pike, Rob xi, xiii, 67, 107
pipeline example 228, 230, 231
pipeline 227
pipeline diagram 228
pitfall, interface 184
pitfall, scope 140
platforms, mobile 121
Playground, Go xvi, 326
playlist example 187
png.Encode function 62
pointer 24, 32, 34
  aliasing 33
  argument 33, 83
  equality 32
  hidden 357
  nil 32
  receiver 158, 167
  to struct 100, 103
  zero value 32
polling channel 246
polymorphism, ad hoc 211
polymorphism, subtype 211
PopCount function 45
Portable Network Graphics (PNG) 62
POSIX file name 72
POSIX standard xi, 55, 72, 197
precedence, operator 52, 63
precision
  floating-point 56, 57, 63, 78
  numeric 55, 78
  of constants 78
predeclared names, table of 28
premature abstraction 216, 316, 317
premature optimization 324
Printf %% 10
Printf verbs, table of 10
Printf %b 10, 54, 75
Printf %c 10, 56
Printf %d 10, 55
Printf %e 10, 57
Printf %f 10, 57
Printf %g 10, 57
Printf %[n] 56
Printf %o 10, 55
Printf %q 10, 56, 97
Printf %s 10
Printf %*s 134
Printf %T 10, 80, 83, 184, 331
Printf %t 10, 83
Printf %#v 106, 207
Printf %v 10, 11
Printf % x 71
Printf %#x 56
Printf %x 10, 55, 83
production code 301
profile
  blocking 324
  CPU 324
  heap 324
profiling 324
programming language
  Alef xiii
  APL xiii
  C++ xiv, xv, 361
  C xii, xv, 1, 6, 52, 260, 361
  Haskell xiv
  Java xv
  JavaScript xv, 107
  Lisp 338
  Modula-2 xiii
  Oberon xiii
  Pascal xiii
  Python xv, 193
  Ruby xv, 193
  Scheme xiii
  Squeak, Newsqueak xiii
promotion, method 161
protocol buffers 107
Python programming language xv, 193

qualified identifier 41, 43
querying packages 298
quote character, ' 56

race
  condition 21, 257, 258, 259
  detector 271, 274
  paradoxical 267
randomized testing 307
range loop 6, 9
range over channel 229
range over map 94
range over string 69, 88
{{range}} template action 113
raw string literal 66
reachability 36
read, stale 268
readers lock 266
read/write mutex 266, 267
ReadWriteCloser interface 174
ReadWriter interface 174
real built-in function 61
receive
  <-ch, channel 18, 225, 232
  expression 225
  non-blocking channel 246
  ok value from channel 229
receive-only channel type <-chan T 230
receiver

  name, method 157
  nil 159, 185
  parameter, method 156
  pointer 158, 167
  type, method 157
`recover` built-in function 152
recursion 121, 124, 247, 333, 339, 345, 359
recursive
  anonymous function 137
  data structure 101, 102, 107
  type 48
redefining `new` 35
reference
  call by 83
  identity 87
  type 9, 12, 93, 120
`reflect` package 330
reflection 329, 352, 359
`reflect.StructTag` type 348
`reflect.Type` type 330
`reflect.Type.Field` method 348
`reflect.TypeOf` function 330
`reflect.Value` type 331, 342
`reflect.Value` zero value 332
`reflect.Value.Addr` method 342
`reflect.Value.CanAddr` method 342
`reflect.Value.Interface` method 331, 342
`reflect.Value.Kind` method 332
`reflect.ValueOf` function 331, 337
`reflect.Zero` function 345
`regexp` package 149
`regexp.Compile` function 149
`regexp.MustCompile` function 149
regular expression 66, 149, 305, 321
relation, “happens before” 226, 257, 261, 277
relationship, “has a” 162
relationship, “is a” 162, 175
remainder operator `%` 52, 166
renaming import 286
rendezvous 234
replacement character �, Unicode 70, 98
repository, `gopl.io` xvi
request
  HTTP GET 21, 127, 272, 348
  HTTP POST 348
  multiplexer, HTTP 193
result list, function 119
result, named 120, 126
return, bare 126
`return` statement 29, 120, 125
`rev` example 86
`reverb` example 223, 224
right shift operator `>>` 54
Ruby programming language xv, 193
`rune` literal 56
`rune` type 52, 67
rune slice to string conversion 71
rune to string conversion 71
`runtime` package 151
runtime API 324
runtime scheduler 281
`runtime.Stack` function 151

satisfaction, interface 171, 175
Scalable Vector Graphics (SVG) 58
scheduler, runtime 281
Scheme programming language xiii
scope
  declaration 45, 137
  `for` 47
  `if-else` 47
  label 46
  local variable 135
  loop variable 141, 236
  pitfall 140
  short variable declaration 22, 48
  `switch` 47
search algorithm, breadth-first 139, 239
search algorithm, depth-first 136
`SearchIssues` function 111
`select case` 245
`select`, `default` case in 246
`select`, non-blocking 246
`select` statement 244, 245
`select{}` statement 245
selective recovery 152
selector, field 156
selector, method 156
semaphore, binary 262
semaphore, counting 241
semicolon 3, 6
send `ch<-`, channel 18, 225, 232
send statement 225
send-only channel type `chan<- T` 230
separate compilation 284
sequence diagram, `thumbnail` 238
sequential consistency 268, 269
serial confinement 262
`server` example 19, 21
server
  chat 253
  concurrent clock 219
  concurrent echo 222
set, map as 96, 202
setter method 169
`sexpr` example 340
S-expression
  decode example 347
  decoding 344
  encoding 338
`sexpr.Marshal` function 340
`sexpr.readList` function 347
`sexpr.Unmarshal` function 347
SHA256 message digest 83
`sha256` example 83
shadowing declaration 46, 49, 206, 212
shared
  lock 266
  variables 257
  variables, concurrency with 257
shared-memory multithreading 217, 257
shift operator `<<`, left 54
shift operator `>>`, right 54
short
  variable declaration 5, 7, 30, 31
  variable declaration scope 22, 48
  variable declaration statement 7
short-circuit
  AND operator `&&` 63
  evaluation 63
  OR operator `||` 63
signature, function 120
signed integer 52, 54
`s[i:j]`, slice operator 84, 86
`s[i:j]`, substring operator 65, 86
simple statement 6, 22
`Sizeof` table 354
`sleep` example 179
slice 4
  argument 86
  capacity 88, 89
  capacity growth diagram 90
  comparison 87
  extending a 86
  key, map with 97
  literal 38, 86
  `make` 88, 322
  nil 87
  of months diagram 84
  operator `s[i:j]` 84, 86
  rotation algorithm 86
  techniques, in-place 91
  type 84
  used as stack 123
  zero length 87
  zero value 74, 87
SMTP client 312
socket
  TCP 219
  UDP 219
  Unix domain 219
sort algorithm, topological 136
`sort` package 95, 186, 189
`sort.Float64s` function 191
`sort.Interface` interface 186
`sort.Ints` function 191
`sort.IntsAreSorted` function 191

sort.IntSlice type 191
sort.Reverse function 189
sort.Strings function 95, 137, 191
spinner example 218
Sprint function 330
SQL API 211
SQL injection attack 211
sqlQuote function 211, 212
squares example 135
Squeak, Newsqueak programming language xiii
stack
  allocation 36
  fixed-size 124
  growth 124, 280, 358
  implementation with slice 92, 215
  overflow 124
  slice used as 123
  trace 149, 253
  variable 36
  variable-size 124
stale read 268
standard
  IEEE 754 56, 57
  POSIX xi, 55, 72, 197
  Unicode 2, 27, 52, 66, 67, 69, 97
statement
  --, decrement 5, 37
  ++, increment 5, 37, 94
  assignment 5, 7, 36, 52, 94, 173
  break 24, 46
  continue 24, 46
  coverage 318, 320
  defer 144, 150, 264
  fallthrough 23, 212
  for 6
  go 18, 218
  goto 24
  if-else 9, 22, 47
  label 46
  labeled 46
  return 29, 120, 125
  select{} 245
  select 244, 245
  send 225
  short variable declaration 7
  simple 6, 22
  switch 23, 47
  tagless switch 24
  type switch 210, 212, 214, 329
  unreachable 120
storage example package 312, 313
Strangelove, Dr. 336
strategies, error-handling 128, 152, 310, 316
strconv package 22, 71, 75
strconv.Atoi function 22, 75
strconv.FormatInt function 75
strconv.Itoa function 75
strconv.ParseInt function 75
strconv.ParseUint function 75
stream, io.Discard 22
String method 40, 166, 329
string
  concatenation operator + 5, 65
  conversion 71
  immutability 65, 73
  index operation 64
  literal 65
  literal, raw 66
  range over 69, 88
  sharing diagram 65
  test 71
  to byte slice conversion 40, 73
  to rune slice conversion 71, 88
  zero value 5, 7, 30
  comparison 65
strings package 7, 71, 72, 289
strings.Contains function 69
strings.HasPrefix function 69
strings.HasSuffix function 69
strings.Index function 289
strings.Join function 7, 12
strings.Map function 133
strings.NewReader function 289
strings.NewReplacer function 289
strings.Reader type 289
strings.Replacer type 289
strings.Split function 12
strings.ToLower function 72
strings.ToUpper function 72
struct declaration 99
struct
  comparison 104
  embedding 104, 161
  Employee 100
  empty 102
  field 15, 99
  field, anonymous 104, 105, 106, 162
  field, embedded 161
  field, export of 101, 106, 109, 110, 168
  field order 101, 355
  field tag 109, 348
  hole 354
  hole diagram 355
  literal 15, 102, 106
  literal, address of 103
  pointer to 100, 103
  type 15, 24, 99
struct{} type 227, 241, 250
struct type, unnamed 163
struct zero value 102
substitutability 193
substring operator s[i:j] 65, 86
subtype polymorphism 211
sum example 142
surface example 59, 203
surface figure, 3-D 58, 203
SVG 58
SWIG 361
Swiss army knife 290
switch, default case in 23
switch, initialization statement in 24
switch scope 47
switch statement 23, 47
switch statement, tagless 24
switch statement, type 210, 212, 214, 329
switch, context 280
sync package 237, 263
synchronous channel 226
sync.Mutex type 263, 269
(*sync.Mutex).Lock method 21, 146, 263
(*sync.Mutex).Unlock method 21, 146, 263
sync.Once type 270
(*sync.Once).Do method 270
sync.RWMutex type 266, 270
(*sync.RWMutex).RLock method 266
(*sync.RWMutex).RUnlock method 266
sync.WaitGroup type 237, 250, 274
(*sync.WaitGroup).Add method 238
(*sync.WaitGroup).Done method 238
syscall package 196, 208
syscall.Errno type 196, 197
system call API 196

table of
  binary operators 52
  bitwise operators 53
  comparison operators 53
  escape sequences 66
  keywords 27
  predeclared names 28
  Printf verbs 10
  UTF-8 encodings 67
table, Sizeof 354
table-driven testing 200, 306, 319
tag, struct field 109, 348
tagless switch statement 24
tags, build 296
TCP socket 219
techniques, in-place slice 91
tempconv example 39, 180, 289
tempconv example package 42
temperature conversion example 29
tempflag example 181
template API 115
template

  | in 113
  action, `{{range}}` 113
  dot `.` in 113
`template.Funcs` method 114
`template.HTML` type 116
`template.Must` function 114
`template.New` function 114
`template.Parse` method 114
`Test` function 302
test
  black-box 310
  brittle 317
  character 71
  coverage 318
  dependency, cyclic 314
  `echo` 309
  failure message 306
  integration 314
  of `word` example 303
  package, external 285, 314
  string 71
  white-box 311
  assertion 306
`_test.go` file 285, 302, 303
`testing` package 285, 302
testing
  a command 308
  randomized 307
  table-driven 200, 306, 319
`testing.B` type 321
`testing.T` type 302
`(*testing.T).Errorf` method 200, 304, 306
`(*testing.T).Fatal` method 306
tests, writing effective 316, 317
`text/scanner` package 344
`text/tabwriter` package 188
`text/template` package 113, 300
Thompson, Ken xi, 67
thread 218, 280
thread-local storage 282
3-D surface figure 58, 203
`thumbnail` example 236, 237, 238
`thumbnail` example package 235
`thumbnail` sequence diagram 238
`time` package 18, 77, 183
`time.After` function 245
`time.AfterFunc` function 164
`time.Duration` type 76, 179
`time.Minute` constant 76
`time.Now` function 220
`time.Parse` function 220
`time.Second` constant 164
`time.Since` function 114
`time.Tick` function 244, 246
`time.Time` type 114
`time.Time.Format` method 220
`title` example 153
`title` function 144, 145
token-based decoder API 213, 215, 347
token-based XML decoding 213
tool
  `cgo` 361, 362
  `go` 2, 42, 44, 290
  `go doc` 25
  `godoc` xvi, 25, 297, 326
  `gofmt` 3, 4, 44, 286
  `goimports` 3, 44, 286
  `golint` 292
topological sort algorithm 136
`topoSort` example 136
`trace` example 146
trace, stack 149, 253
tree, binary 102
`treesort` example 102
`true` boolean constant 63
truncation, floating-point 40, 55
tuple assignment 31, 37
type declaration 39
`type` keyword 212
type
  abstract 24, 171
  aggregate 81, 99
  array 81
  assertion 205, 211
  assertion, interface 208, 210
  assertion, ok value from 206
  `bool` 63
  `bufio.Scanner` 9
  `byte` 52
  `bytes.Buffer` 74, 169, 172, 185
  `Celsius` 39
  `chan` 225
  channel 18
  `<-chan T`, receive-only channel 230
  `chan<- T`, send-only channel 230
  `complex` 61
  composite xv, 14, 81
  composition xv, 107, 162, 189
  concrete 24, 171, 211, 214
  displaying methods of a 351
  empty interface 176
  `error` built-in 11, 128, 149, 196
  `Fahrenheit` 39
  function 119, 120
  `http.HandlerFunc` 194, 203
  `http.Request` 21, 253
  `http.ResponseWriter` 19, 22, 191, 193
  `int` 52
  `interface{}` 143, 176, 331
  interface 171, 174
  interface dynamic 181
  `json.Decoder` 111
  `json.Encoder` 111
  map 9, 93
  method receiver 157
  mismatch 55
  named 24, 39, 40, 105, 157
  `net.Conn` 220
  `net.Listener` 220
  numeric 51
  `*os.File` 11, 13, 172, 175, 185, 336
  `os.FileInfo` 247
  `os.LinkError` 207
  `os.PathError` 207
  recursive 48
  reference 9, 12, 93, 120
  `reflect.StructTag` 348
  `reflect.Type` 330
  `reflect.Value` 331, 342
  `rune` 52, 67
  slice 84
  `sort.IntSlice` 191
  `strings.Reader` 289
  `strings.Replacer` 289
  `struct{}` 227, 241, 250
  struct 15, 24, 99
  `switch`, `case` in 212
  `switch`, `default` case in 212
  `switch` statement 210, 212, 214, 329
  `sync.Mutex` 263, 269
  `sync.Once` 270
  `sync.RWMutex` 266, 270
  `sync.WaitGroup` 237, 250, 274
  `syscall.Errno` 196, 197
  `template.HTML` 116
  `testing.B` 321
  `testing.T` 302
  `time.Duration` 76, 179
  `time.Time` 114
  `uint` 52
  `uintptr` 52, 354, 357
  underlying 39
  unidirectional channel 230, 231
  unnamed struct 163
  `unsafe.Pointer` 356
  `url.URL` 193
types, untyped constant 78

UDP socket 219
`uint` type 52
`uintptr` type 52, 354, 357
unary operator `+` 53
unary operator `-` 53
unbuffered channel 226
undefined behavior 260
underlying array 84, 88, 91, 187
underlying type 39
Unicode
  code point 67
  escape 68, 107
  replacement character � 70, 98

standard 2, 27, 52, 66, 67, 69, 97
unicode package 71
unicode.IsDigit function 71
unicode.IsLetter function 71
unicode.IsLower function 71
unicode.IsSpace function 93
unicode.IsUpper function 71
unicode/utf8 package 69
unidirectional channel type 230, 231
union, discriminated 211, 213, 214
universe block 46
Unix domain socket 219
unmarshaling JSON 110
unnamed struct type 163
unnamed variable 34, 88
unreachable statement 120
unsafe package 354
unsafe.AlignOf function 355
unsafe.Offsetof function 355
unsafe.Pointer conversion 356
unsafe.Pointer type 356
unsafe.Pointer zero value 356
unsafe.Sizeof function 354
unsigned integer 52, 54
untyped constant types 78
unused parameter 120
URL 123
URL escape 111
url.QueryEscape function 111
url.URL type 193
urlvalues example 160
UTF-8 66, 67, 98
UTF-8 encodings, table of 67
utf8.DecodeRuneInString function 69
utf8.RuneCountInString function 69
utf8.UTFMax value 98

value
  addressable 32
  call by 83, 120, 158
  function 132
  interface 181
  method 164
  utf8.UTFMax 98
var declaration 5, 30
variable
  confinement 261
  heap 36
  http.DefaultClient 253
  io.Discard 18
  io.EOF 132
  lifetime 35, 46, 135
  local 29, 141
  os.Args 4
  stack 36
  unnamed 34, 88
variables, escaping 36
variables, shared 257
variable-size stack 124
variadic function 142, 172
vector, bit 165
vendoring 293
visibility 28, 29, 41, 168, 297
visit function 122

wait example 130
WaitForServer function 130
walkDir function 247
web
  crawler 119
  crawler, concurrent 239
  framework 193
while loop 6
white-box test 311
Wilkes, Maurice 301
Wirth, Niklaus xiii
word example 303, 305, 308
word example, test of 303
workspace organization 291
writer lock 266
writing effective tests 316, 317

xkcd JSON interface 113
XML decoding 213
XML (Extensible Markup Language) 107
(*xml.Decoder).Token method 213
xmlselect example 215

zero length slice 87
zero value
  array 82
  boolean 30
  channel 225, 246
  function 132
  interface 182
  map 95
  named result 120, 127
  number 5, 30
  pointer 32
  reflect.Value 332
  slice 74, 87
  string 5, 7, 30
  struct 102
  unsafe.Pointer 356