

## Index

## #

#1#, 273  
 #1=, 273  
 #n#, 145  
 #n=, 145  
 (), 9  
 (), 395  
 \*, 136  
   in COMMON LISP, 37  
 +, 27, 385  
   in COMMON LISP, 37  
 ".scm", 261  
 ".so", 261  
 <, 27  
 <=, 136  
 =, 385  
 ==, 306  
 >, 452  
 ??, 306  
 &aux, 14  
 &key, 14  
 &rest, 14

## A

a.out, 321  
 abstraction, 11, 14, 82, 149, 150, 157,  
   186, 189, 191, 209, 214, 230,  
   286, 287, 289, 293, 295, 306,  
   307, 348, 453, 468  
   closure of, 19  
   denotation of, 173, 175, 184  
   macros and, 311, 333  
   migration and, 184  
   redex and, 150  
   reflective, 307  
   value of, 19  
 accompanying function, 440  
 acons, 342  
 activation record, 87, 185  
   allocating, 189, 222  
 activation-frame, 185  
 \*active-catchers\*, 77  
 add-method!, 296  
 address, 116, 186  
 adjoin-definition!, 368  
 adjoin-free-variable!, 366  
 adjoin-global-variable!, 203  
 adjoin-temporary-variables!, 371  
 A-list, 285  
   definition, 13  
 allocate, 158  
 allocate, 132  
 allocate-Class, 441  
 ALLOCATE-DOTTED-FRAME, 217  
 ALLOCATE-FRAME, 216, 243  
 ALLOCATE-FRAME1, 243  
 allocate-immutable-pair, 143  
 allocate-list, 135  
 allocate-pair, 135  
 allocate-Poly-Field, 441  
 allocate-Polygon, 429  
 allocating  
   activation records, 189, 222  
 allocator, 422, 423  
   with initialization, 433  
   without initialization, 431  
 $\alpha$ , 152  
 $\alpha$ -conversion, 370  
 already-seen-value?, 375  
 ALTERNATIVE, 213, 230  
 Alternative, 344  
 alternative, 8, 90, 178, 209, 381  
 and, 477  
 another-fact, 64  
 apostrophe, 312  
 Application, 344  
 application, 199  
   autonomous, 283  
   functional, 7, 11  
 apply, 219, 331, 399, 401, 409, 453, 460,  
   464

apply-cont, 93  
 \*arg1\*, 228  
 \*arg2\*, 228  
 argument  
   as discriminant, 442  
 argument-cont, 93  
 Arguments, 344  
 arguments-, 396  
 arguments->C, 384  
 arguments->CPS, 406  
 arity  
   if, 7  
   varying, 196  
   verifying, 244  
 ARITY=?, 244  
 ARITY=?2, 244  
 assembly language, 230  
 assignment, 11, 111  
   addresses and, 116  
   binding and, 113  
   closed variables and, 112  
   compiling, 381  
   continuation and, 113  
   environment and, 113  
   immutable binding and, 26  
   linearizing, 228  
   local variables and, 112  
   predefined variables, 121  
   shared variables and, 112  
   value of, 13  
 assoc/de, 51  
 association list  
   definition, 13  
 atom?, 4  
 auto-interpretation, 28, 307  
 autonomous application, 283

## B

B, 163  
 backpatching, 234  
 backquote, 333, 340, 344  
 bar, 27  
 base-error-handler, 259  
 begin, xviii, 7, 9, 11, 307  
 begin-cont, 90  
 behavior, 131  
 $\beta$ -reduction  
   redex and, 150  
 better-map, 80  
 between-parentheses, 380

bezout, 103  
 Bigloo, 238, 360, 414, 415  
 bind/de, 51  
 bind-exit, 101, 249  
 binding, 113  
   assignment and, 113  
   captured, 114  
   captured by closures, 43  
   capturing, 343  
   creating an uninitialized, 60  
   deep, 23  
   dynamic, 150, 300  
   dynamic, definition, 19  
   environments and, 43  
   extent, 68  
   immutable, 26  
   implementation techniques, 25  
   initialized, 60  
   lexical, 150, 300  
   lexical, definition, 19  
   mutable, 26  
   not first class objects, 68  
   quasi-static, 300  
   scope and, 68  
   semantics, 25  
   shallow, 23  
   uninitialized, 54, 59  
 bindings->C, 383  
 block, 76, 80, 249  
   compared to catch, 79, 81  
 block-cont, 97  
 block-env, 97  
 block-lookup, 98, 100  
 Boolean  
   false, 26  
   true, 26  
   false, 9  
   in  $\lambda$ -calculus, 155  
   in C, 382  
   representing, 8  
   syntactic forms, 26  
 boolean?, xviii  
 boolean->C, 382  
 bootstrapping, 334, 336, 414, 419, 427,  
   441  
   MEROON, 335  
 bottom-cont, 95  
 bound variable  
   definition, 3  
 box, 114, 362  
   compiling, 381

- introducing, 371
  - simulating dotted pairs, 122
  - transforming variables, 115
  - Box-Creation, 362
  - boxify-mutable-variables, 363
  - Box-Read, 362
  - box-ref, 114
  - box-set!, 114
  - Box-Write, 362
  - build-application, 262
  - build-application-renaming-variables, 471
  - bury-r, 288
  - byte, 235
- ## C
- >C, 379–385
  - C, 359
  - C++, 415
  - C, Kernighan & Ritchie, 415
  - cadr, xviii, 121
  - call by name, 150, 151, 176, 180
  - call by need, 151, 176
  - call by value, 11, 150, 175, 176, 180
  - CALL0, 218
  - CALL1, 233
  - CALL1-car, 243
  - CALL1-cdr, 243
  - CALL2, 233
  - CALL3, 218, 228, 233
  - call-arguments-limit, 401
  - call/cc, 95, 195, 219, 247, 249, 402, 404, 453, 460, 464
  - call/ep, 402
  - calling
    - macros, 317
  - calling protocol, 184, 227, 228
  - call-next-method, 446
  - Caml Light, 225, 312
  - can-access?, 116
  - captured binding, 114
  - capture-the-environment, 472
  - capturing
    - bindings, 343
  - car, xviii, 27, 38, 94, 137, 195, 385, 398
  - careless-Class-fields, 428
  - careless-Class-name, 428
  - careless-Class-number, 428
  - careless-Class-superclass, 428
  - careless-Field-defining-class-number, 428
  - careless-Field-name, 428
  - case, xviii
  - catch, 74, 77, 78, 85
    - compared to block, 79, 81
    - cost of, 77
    - simulé par block, 77
    - simulated by a dynamic variable, 78
    - with unwind-protect, 85
  - catch-cont, 96
  - catch-lookup, 96
  - cc, 315, 344, 462
  - cdr, xviii
  - chap3j-interpreter, 460
  - chapter1d-scheme, 453
  - chapter1-scheme, 27
  - chapter3-interpreter, 95
  - chapter4-interpreter, 138
  - chapter61-interpreter, 195
  - chapter62-interpreter, 207
  - chapter63-interpreter, 219
  - chapter7d-interpreter, 246
  - char?, xviii
  - check-byte, 239
  - check-class-membership, 430
  - check-conflicting-name, 440
  - CHECKED-DEEP-REF, 292
  - CHECKED-GLOBAL-REF-, 466
  - CHECKED-GLOBAL-REF, 212, 240
  - CHECKED-GLOBAL-REF+, 467
  - CHECKED-GLOBAL-REF-code, 264
  - checked-r-extend\*, 290
  - check-index-range, 436
  - check-signature-compatibility, 447
  - Church's thesis, 2
  - Church-Rosser property, 150
  - circularity, 419
  - COMMON LISP, xx
    - \*in, 37
    - +in, 37
  - cl.lookup, 48
  - >Class, 420
  - Class?, 441
  - class, 87
    - Class, 427
    - ColoredPolygon, 423
    - CountedPoint, 477
    - Field, 426, 427
    - Generic, 443
    - Mono-Field, 426, 427

- Object, 426
- Point, 421
- Poly-Field, 426, 427
- Polygon, 421
  - defining, 87, 422
  - definition, 87
  - numbering, 419
  - virtual, 89
- Class-class, 427
- classe
  - ColoredPolygon, 421
- \*classes\*, 419
- Class-generate-related-names, 429
- Class-initialize!, 439
- Class-name, 441
- Class-number, 441
- \*class-number\*, 419
- CLICC, 360
- clone, 407
- CLOS, 87, 432, 442
- closure, 211
- closure, 211
  - definition, 19, 21
  - examples, 19
  - impoverished, 135
  - lexical escape environment and, 79
- Closure-Creation, 368
- closurize-main!, 369
- Cname-clash?, 374
- code walker, 340, 341
- code walking, 360
- co-definition, 120
- code-prologue, 246, 258
- collecting
  - quotations, 367, 372
  - variables, 372
- collect-temporaries!, 370, 371
- combination
  - definition, 11
- combinator, 183
  - fixed point, 63
  - K, 16
- Common Loops, 87, 442
- communication
  - hidden, 314
- comparing
  - physically, 137
- compatible-arity?, 465
- compile-and-evaluate, 289
- compile-and-run, 278
- compile->C, 372, 404, 475
- compile-file, 260, 313, 327
- compile-on-the-fly, 278
- compiler-let, 330
- COMPILE-RUN, 278
- compiling, 223
  - assignments, 381
  - boxes, 381
  - expressions, 379
  - functional applications, 382
  - into C, 359
  - macros, 332
  - quotations, 375
  - separately, 260, 426
- compositional quotation, 140
- computation
  - continuation and, 87
  - environment and, 87
  - expression and, 87
- compute-another-Cname, 374
- compute-Cname, 374
- compute-field-offset, 437
- compute-kind, 191, 203, 288, 297
- cond, xviii
- conditional, 178
- confusion
  - values with programs, 277
- cons, xviii, 27, 38, 94, 137, 183, 195, 385, 397
- CONS-ARGUMENT, 217, 233
- CONSTANT, 212, 225, 241
- Constant, 344
- CONSTANT0, 242
- CONSTANT-1, 242
- CONSTANT-code, 264
- Continuation, 405
- continuation, 89, 249
- continuation, 71
  - as functions, 81
  - assignment and, 113
  - capturing, 81
  - computation and, 87
  - constant, 207
  - definition, 73
  - extent, 78, 79, 83
  - forms for handling, 74
  - interpreter, 89
  - multiple returns, 189
  - partial, 106
  - programming by, 71
  - programming style, 102, 404-406, 410

- representing, 81, 87
  - Continuation Passing Style, 177
  - control form, 74
  - convert2arguments, 347
  - convert2Regular-Application, 405
  - count, 355
  - CountedPoint, 477
  - CountingClass, 476
  - cpp, 315, 321
  - >CPS, 405, 406
  - CPS, 177
  - cps, 177
  - cps-abstraction, 179
  - cps-application, 179
  - cps-begin, 178
  - cps-fact, 462
  - cps-if, 178
  - cpsify, 405
  - cps-quote, 178
  - cps-set!, 178
  - cps-terms, 179
  - CREATE-1ST-CLASS-ENV, 287, 288
  - create-boolean, 134
  - CREATE-CLOSURE, 230, 244
  - create-evaluator, 352
  - create-first-class-environment, 288
  - create-function, 134
  - create-immutable-pair, 143
  - create-number, 134
  - create-pair, 135
  - CREATE-PSEUDO-ENV, 297
  - create-pseudo-environment, 297
  - create-symbol, 134
  - creating
    - global variables, 279
    - variables to define them, 54
  - crypt, 118
  - currying, 319
  - Cval, 285
  - C-value?, 376
  - cycle, 144
  - cycle
    - quoting and, 144
- ## D
- d.determine!, 466
  - d.evaluate, 17
  - d.invoke, 17, 21
  - d.make-closure, 21
  - d.make-function, 17, 21
  - data, 4
    - quoting, 7
  - dd.eprogn, 50
  - dd.evaluate, 50
  - dd.evlis, 50
  - dd.make-function, 50
  - deep binding
    - definition, 23
    - implementation techniques, 25
  - DEEP-ARGUMENT-REF, 212, 240
  - DEEP-ARGUMENT-SET!, 213, 233
  - deep-fetch, 186
  - deep-update!, 186
  - defforeignprimitive, 413
  - define, xix, 54, 308
  - define-abbreviation, 316, 318, 321
  - define-abbreviation, 323, 330
  - define-alias, 333
  - define-class, 87, 319, 423–425
    - compiled, 425
    - internal state of, 423
    - interpreted, 424
  - define-CountingClass, 476
  - define-generic, 88, 444
    - syntax, 443
  - define-handy-method, 340
  - define-inline, 339
  - define-instruction, 235
  - define-instruction-set, 235
  - define-macro, 316
  - define-meroonet-macro, 336
  - define-method, 88, 447, 478
    - variant, 478
    - variation, 448
  - define-syntax, 316, 321
  - defining
    - classes, 422
    - macros by macros, 333
    - variables by creating them, 54
    - variables by modifying values, 54
  - definitial, 25, 94, 136
  - definitial-function, 38
  - definition
    - multiple, 120
  - defmacro, 321
  - defpredicate, 452
  - defprimitive, 25, 38, 94, 136, 200, 384
  - defprimitive2, 218
  - defprimitive3, 200
  - defvariable, 203
  - delay, 176

$\delta$ , 167  
 $\delta$ -rule, 151  
 denotational semantics, 147  
     definition, 149  
 desc.init, 200  
 description-extend!, 200  
 descriptor  
     definition, 200  
 determine!, 466  
 determine-method, 444  
 df.eprogn, 45  
 df.evaluate, 45, 48  
 df.evaluate-application, 45  
 df.make-function, 45  
 disassemble, 221  
 discriminant, 442  
 dispatch table, 443  
 display, xviii  
 display-cyclic-spine, 47  
 dotted pair  
     simulating by boxes, 122  
 dotted signature, 443  
 Dylan, xx, 101  
 dynamic, 169, 455  
 dynamic  
     binding, 19  
     escapes, 76  
     extent, 79, 83  
     Lisp, 19  
     Lisp, example, 20, 23  
 dynamic binding  
      $\lambda$ -calculus and, 150  
 dynamic error, 274  
 dynamic escape  
     compared to dynamic variable, 79  
 dynamic variable, 44  
     compared to dynamic escape, 79  
     protection and, 86  
 dynamic-let, 169, 252, 300, 455  
 DYNAMIC-POP, 253  
 DYNAMIC-PUSH, 253  
 DYNAMIC-PUSH-code, 265  
 DYNAMIC-REF, 253  
 DYNAMIC-REF-code, 265  
 dynamic-set!, 300, 455  
 \*dynamic-variables\*, 254  
 \*dynenv\*, 469  
 dynenv-tag, 253

## E

EcoLisp, 360  
 ef, 155  
 egal, 124  
 electronic address, xix  
 empty-begin, 10  
 endogenous mode, 316, 322  
 enrich, 290, 472  
 enumerate, 112, 473  
 enumerate-aux, 473  
 \*env\*, 207, 231  
 env.global, 25  
 env.init, 14  
 Envir, 472  
 Environment, 348  
 environment, 89, 185  
 environment  
     as abstract type, 43  
     as composite abstract type, 14  
     assignment and, 113  
     automatically extendable, 172  
     changing to simulate shallow binding, 24  
     closures and lexical escapes, 79  
     computation and, 87  
     contains bindings, 43  
     definition, 3  
     dynamic variable, 44  
     enriching, 290  
     execution, 16  
     extendable, 119  
     first class, 286  
     flat, 202  
     frozen, 117  
     function, 44  
     functional, 33  
     global, 25, 116, 117, 119  
         hyperstatic, 173  
     hyperstatic, 119  
     initial, 14, 135  
     inspecting, 302  
     lexical escape, 76  
     linked, 186  
     making it global, 206  
     mutually exclusive, 43  
     name space and, 43  
     representing, 13  
     Scheme lexical, 44  
     universal, 116  
     variables, 91

- environnement
  - global, 16
- eprogn, 9, 10
- eq?, 27, 123
- eq1, 123
- equal?, 123, 124
  - cost, 138
- equality, 122
- equalp, 123
- eqv?, 123, 125, 137
- error
  - dynamic, 274
  - static, 194, 274
- escape, 249, 250
- escape
  - definition, 71
  - dynamic, 79
  - extent, 78, 79, 81, 83
  - invoking, 251
  - lexical, 79
  - lexical, properties of, 76
  - valid, 251
- ESCAPER, 250
- escape-tag, 250
- escape-valid?, 251
- EULISP, xx
- EULISP, 101
- eval, 2, 271, 280, 332, 414
  - function, 280
  - interpreted, 282
  - properties, 3
  - special form, 277
- eval/at, 280, 472
- eval/b, 289
- EVAL/CE, 278
- eval/ce, 337
- eval-in-abbreviation-world, 328
- evaluate, 4, 7, 89, 128, 271, 275, 350
- evaluate-amnesic-if, 128
- evaluate-application, 34, 93, 131
- evaluate-application2, 35
- evaluate-arguments, 93, 105
- evaluate-begin, 90, 104, 105, 129
- evaluate-block, 97
- evaluate-catch, 96
- evaluate-eval, 275
- evaluate-ftn-lambda, 132
- evaluate-if, 90, 128
- evaluate-immutable-quote, 143
- evaluate-lambda, 92, 131, 459, 460
- evaluate-memo-quote, 141
- evaluate-nlambda, 465
- evaluate-or, 463
- evaluate-quote, 90, 140
- evaluate-return-from, 98
- evaluate-set!, 92, 130, 463
- evaluate-throw, 96
- evaluate-unwind-protect, 99
- evaluate-variable, 91, 130
- evaluation, 271
  - levels of, 351
  - simultaneous, 331
  - stack, 71
- evaluation order, 467
- evaluation rule
  - definition, 150
- Evaluator, 351
- evaluator
  - definition, 2
- eval-when, 328
- even?, 55, 66, 120, 290
- evfun-cont, 93
- evlis, 12, 451
- exception, 223
- executable, 313
- execution library, 397
- existence of a language, 28
- exogenous mode, 314, 323
- expander, 316, 318
- expand-expression, 329
- expand-store, 133
- Expansion Passing Style, 318
- EXPLICIT-CONSTANT, 241
- export, 286
- expression
  - computation and, 87
- extend, 14, 29
- extendable global environment, 119
- EXTEND-ENV, 244
- extend-env, 92
- extending
  - languages, 311
- extensionality, 153
- extent
  - continuations, 78, 79, 83
  - dynamic, 79, 83
  - escapes, 78, 79, 81, 83
  - indefinite, 81, 83
- extract!, 368
- extract-addresses, 289
- extract-things!, 368

**F**

- f, 26, 136, 195
- f.eprogn, 33
- f.evaluate, 33, 41
- f.evaluate-application, 41
- f.evlis, 33
- f.lookup, 41
- f.make-function, 34
- fact, 27, 54, 71, 82, 262, 324, 326
  - with continuation, 102
  - with call/cc, 82
  - written with prog, 71
- fact1, 322, 325
- fact2, 72, 325
- Fact-Closure, 365
- factfact, 458
- factorial, 322, 324–326
- factorial, 54, 365
- feature, 331
- \*features\*, 331
- fenv.global, 38
- fetch-byte, 237
- FEXPR, 302, 303
- fib, 27
- Field?, 441
- field
  - accessing length of, 438
  - reader, 436
  - writer, 437
- field descriptor
  - in MEROONET, 419, 420
- Field-class, 427
- Field-define-class, 427
- Field-defining-class, 427
- Field-generate-related-names, 435
- field-length, 438
- field-value, 437
- find-dynamic-value, 470
- find-expander, 318
- find-global-environment, 348
- find-symbol, 72, 75, 76, 82
  - naive style, 72
  - with block, 76
  - with call/cc, 82
  - with throw, 75
- find-variable?, 348
- FINISH, 246
- first class citizen, 32
- first class environment, 286
- first class module, 297
- first class object
  - bindings are not, 68
  - in MEROONET, 418
- fix, 63
- fix2, 457
- FIX-CLOSURE, 214, 227, 233
- fixed point
  - least, 64
- fixed point combinator, 63
- FIX-LET, 217, 226, 233
- Fix-Let, 344
- fixN, 457
- flambda?, 306
- flambda-apply, 306
- flat environment, 202
- Flat-Function, 365
- Flattened-Program, 368
- flat-variables, 445
- flet, 39
- floating-point contagion, 442
- foo, 27
- force, 176
- foreign interface, 413
- form, xviii
  - binding, 68
  - normal, 150
  - reducible, 197
  - special, 311, 312
  - special, definition, 6
- frame coalescing, 383
- free variable
  - $\lambda$ -calculus and, 150
  - definition, 3
  - dynamic scope and, 21
  - lexical scope and, 21
- Free-Environment, 365
- Free-Reference, 365
- FROM-RIGHT-CONS-ARGUMENT, 467
- FROM-RIGHT-STORE-ARGUMENT, 467
- frozen global environment, 117
- ftp, xix
- full-env, 91
- Full-Environment, 348
- \*fun\*, 228
- funcall, 36
- funcallable object, 442
- Function, 344
- function, 37, 92
  - closure, 21
- function
  - accompanying, 429, 440



- applying in  $\lambda$ -calculus, 150
  - calling protocol, 184, 227, 228
  - continuation as, 81
  - continuations, 92
  - descriptor of, 200
  - environment, 44
  - execution environment of, 16
  - generic, 87, 418, 441, 442
  - generic, defining, 88
  - generic, representing, 442
  - integrable, 26
  - nested, 363
  - nested, eliminating, 372
  - objects, 92
  - open coded, 26
  - predefined, 440
  - primitive, definition, 6
  - representing, 15
  - space, 44
  - substituting terms in  $\lambda$ -calculus, 150
  - variable arity of, 196
  - writing in  $\lambda$ -calculus, 150
  - function position
    - lambda form in, 34
    - lists in, 40
    - numbers in, 39
  - function\_, 386
  - functional
    - definition, 67
  - functional application
    - compiling, 382
    - definition, 7, 11
    - representing, 4
  - functional environment, 33
  - functional object, 442
  - Functional-Description, 384
  - functionally applicative object, 40
  - Function-Definition, 368
  - FUNCTION-GOTO, 244
  - FUNCTION-INVOKE, 229, 244
  - function-nadic, 460
  - function-with-arity, 459
  - fusing
    - temporary blocks, 383
  - Fval, 285
- ## G
- g.current, 191
  - g.current-extend!, 191
  - g.current-initialize!, 192
  - g.init, 191
  - g.init-extend!, 191
  - g.init-initialize!, 192
  - g.predef, 350
  - $\gamma$ , 170
  - garbage collection, 431
  - garbage collector, 390
  - gatekeeper, 118
  - gather-cont, 93
  - gather-temporaries!, 370
  - generate-arity, 385
  - generate-closure-structure, 386
  - generate-C-program, 372
  - generate-C-value, 376
  - generate-functions, 386
  - generate-global-environment, 373
  - generate-global-variable, 373
  - generate-header, 373
  - generate-local-temporaries, 387
  - generate-main, 387
  - generate-next-method-functions, 479
  - generate-pair, 377
  - generate-possibly-dotted-definition, 386
  - generate-quotation-alias, 376
  - generate-quotations, 375
  - generate-related-names, 477, 478
  - generate-symbol, 377
  - generate-trailer, 373
  - generate-vector-of-fix-makers, 334
  - >Generic, 420
  - Generic?, 441
  - generic function, 87, 418, 441, 442
    - default body, 445
    - defining, 88
    - dispatch table, 443
    - dotted signature, 443
    - image of, 444
    - in MEROONET, 418
    - representing, 442
  - Generic-class, 427
  - Generic-name, 441
  - \*generics\*, 420
  - gensym, 342
  - get-description, 200
  - get-dynamic-variable-index, 254
  - global environment, 25, 116, 117, 119, 206
  - global variable, 240
    - creating, 279
  - Global-Assignment, 344

global-env, 307, 308  
 global-fetch, 192  
 GLOBAL-REF, 212, 240  
 GLOBAL-REF-code, 264  
 Global-Reference, 344  
 GLOBAL-SET!, 213, 225, 233  
 global-update!, 192  
 global-value, 283, 470  
 Global-Variable, 344, 475  
 global-variable?, 191  
 Gnu Emacs Lisp, 268  
 go, 108  
 GOTO, 229, 242  
 goto, 71  
 grammar  
     Scheme, 272  
 ground floor, 351

## H

handle-location, 472  
 hash table, 125  
 header file `scheme.h`, 373  
 hidden communication, 314  
 hygiene  
     first rule, 342  
 hygiene, 341–343, 355  
 hygienic  
     definition, 61  
     variable, 61  
 hyper-fact, 69  
 hyperstatic  
     definition, 55  
     purely lexical global world, 55  
 hyperstatic global environment, 119

## I

identifier, legal, 374  
 identity, 35  
 IdScheme->IdC, 374  
**If**, 155  
 if, 7, 8  
     arity, 7  
 if-cont, 90  
 image of generic function, 444  
 immutable binding  
     assignment and, 26  
     definition, 26  
 immutable-transcode, 143  
 implementation

    of MEROONET, 419  
 import, 296  
 impoverished closure, 135  
 include file `scheme.h`, 373  
 incredible, 321  
 infinite regression, 337, 430  
 inheritance  
     fields, 422  
     methods, 422  
 inian!, 475  
 initialization-analyze!, 475  
 initialize!, 477  
 inline, 26, 121, 199, 243, 339, 384  
 insert-box!, 362, 363  
 insert-global!, 348  
 inspecting  
     environments, 302  
 install-code!, 263  
 install-macro!, 322  
 install-object-file!, 263  
 instantiation link, 419  
 instruction-decode, 237  
 instruction-size, 237  
 int->char, xviii  
 integrable function, 26  
 integrating  
     MEROONET with Scheme, 443  
     primitives, 199  
 integration, 199  
 interaction loop  
     compiling, 195  
 interface, foreign, 413  
 intermediate language, 224  
 interpreter  
     continuations, 89  
     fast, 183  
     reflective, 308  
**invoke**, 15, 89, 92, 95, 96, 211, 245, 249,  
     251, 283, 365, 454, 459, 460  
**INVOKE1**, 243  
 invoking  
     escapes, 251  
 iota, 458  
 is-a?, 430  
 IS-Lisp, xx

## J

jmp.buf, 403  
 jump, 242  
**JUMP-FALSE**, 229, 242

**K**

$\kappa$ , 152  
**kar**, 122  
 keyword, 316  
**klop**, 69  
**kons**, 122  
 Kyoto COMMON LISP, 360

**L**

$\mathcal{L}$ , 161  
**label**, 56  
**labeled-cont**, 96  
**labels**, 57  
**lambda**, xix, 7, 11, 183, 303  
**lambda form in function position**, 34  
 $\lambda$ -calculus, 147, 149  
   applied, 151  
   binding and, 150  
   combinators in, 63  
 $\lambda$ -drifting, 184  
 $\lambda$ -hoisting, 184  
**lambda-parameters-limit**, 239  
 language  
   defined, 1  
   defining, 1  
   existence of, 28  
   extending, 311  
   intermediate, 224  
   lazy, 59  
   program meaning and, 147  
   purely functional, 62  
   target, 359  
   universal, 2  
**\*last-defined-class\***, 425  
**ld**, 262  
 legal identifier, 374  
 Le-Lisp, 148  
 length of field, 438  
**let\***, xix  
**let**, xix, 47  
   creating uninitialized bindings, 60  
   purpose of, 58  
**let-abbreviation**, 318, 321  
**let/cc**, 101, 249  
**letify**, 407  
**letrec**, xix, 57, 290  
   expansion of, 58  
**letrec-syntax**, 321  
**let-syntax**, 321

lexical  
   binding, 19  
   escape environment, closures and,  
     79  
   escapes, 76  
   escapes, properties of, 76  
   Lisp, 19  
   Lisp, example, 20  
 lexical binding  
    $\lambda$ -calculus and, 150  
 lexical environment  
   in Scheme, 44  
 lexical escape  
   compared to lexical variable, 79  
 lexical index, 186  
 lexical indices, 186  
 lexical variable  
   compared to lexical escape, 79  
   space, 44  
**lift!**, 366  
**lift-procedures!**, 366  
 linearizing, 225, 226  
   assignments, 228  
 linking, 260  
 Lisp  
   dynamic, 19, 21  
   dynamic, example, 20, 23  
   lexical, 19  
   lexical, example, 20  
   literature about, 1  
   longevity, 311  
   special forms and primitive func-  
     tions, 6  
   Turing machine and, 2  
   universal language and, 2  
 Lisp<sub>1</sub>, 32  
 Lisp<sub>2</sub>, 32  
 Lisp<sub>1</sub>  
   local recursion in, 57  
 Lisp<sub>2</sub>  
   function space in, 44  
   local recursion in, 56  
   name conflicts in, 42  
 LiSP2TeX, 175  
 Lisp<sub>1</sub>  
   compared to Lisp<sub>2</sub>, 34, 40  
 Lisp<sub>2</sub>  
   compared to Lisp<sub>1</sub>, 34, 40  
**list**, xviii, 453, 467  
   with combinators, 467  
 list

as functional application, 11  
in function position, 40  
infinite, 59  
listify!, 196  
literature about Lisp, 1  
load, 268, 315, 332, 414, 469  
local variable, 239  
Local-Assignment, 344  
Local-Reference, 344  
Local-Variable, 344  
local-variable?, 186  
location, 116  
LONG-GOTO, 242  
longjmp, 402, 404  
lookup, 4, 13, 61, 89, 91, 285, 452  
loop, 312, 344

## M

m4, 316  
machine  
  Turing, 2  
  virtual, 183  
macro, 311  
  beautification, 338  
  calling, 317  
  compiling, 332  
  composable, 315  
  defined by a macro, 333  
  displacing, 320  
  exported, 334  
  expressiveness, 316  
  filtering, 340  
  hygiene and, 341, 343, 355  
  hygienic, 339  
  hyperstatic, 332  
  local, 334  
  looping, 320  
  masks, 338  
  occasional, 333  
  redefining, 331  
  scope, 333  
  shortcuts, 338  
macro expander  
  monolithic, 315  
macro expansion, 314  
  endogenous, 314  
  exogenous, 314  
  order of, 425  
macro-character, 312  
macro-eval, 322, 332  
macrolet, 319, 321  
\*macros\*, 322  
Magic-Keyword, 345  
make-allocator, 432  
make-box, 109, 114  
make-Class, 441  
make-code-segment, 246  
make-Fact-Closure, 365  
make-fix-maker, 434  
make-flambda, 306  
make-function, 11, 16, 19, 283  
make-lengther, 438  
make-macro-environment, 353, 474  
make-maker, 434  
make-named-box, 126  
make-predefined-application-generator,  
  385  
make-predicate, 430  
make-reader, 436  
make-string, xviii  
make-toplevel, 306  
make-vector, xviii  
make-writer, 438  
malloc, 390  
mangling, 374  
map, 22  
mark-global-preparation-environment,  
  348  
\*maximal-fixnum\*, 376  
\*maximal-number-of-classes\*, 419  
meaning\*, 190, 210, 216  
meaning, 175, 187, 188, 207, 276, 277, 297  
meaning, 148  
  reference implementation and, 148  
meaning\*-multiple-sequence, 188, 209,  
  214  
meaning\*-single-sequence, 188, 209, 214  
meaning-abstraction, 175, 184, 197, 214  
meaning-alternative, 188, 209, 213  
meaning-application, 200, 215  
meaning-assignment, 193, 213, 291  
meaning-bind-exit, 250  
meaning-closed-application, 197  
meaning-define, 204  
meaning-dotted\*, 198, 217  
meaning-dotted-abstraction, 196, 214  
meaning-dotted-closed-application, 198,  
  217  
meaning-dynamic-let, 253  
meaning-dynamic-reference, 253  
meaning-eval, 276, 278

meaning-export, 287  
 meaning-fix-*abstraction*, 191, 209, 214, 293, 468  
 meaning-fix-closed-application, 197  
 meaning-import, 297  
 meaning-monitor, 257  
 meaning-no-argument, 190, 210, 216  
 meaning-no-dotted-argument, 198, 217  
 meaning-primitive-application, 201, 218  
 meaning-quotation, 188, 212  
 meaning-redefine, 468  
 meaning-reference, 193, 208, 212, 291, 299  
 meaning-regular-application, 189, 210, 216  
 meaning-sequence, 188, 209, 214  
 meaning-some-arguments, 190, 210, 216  
 meaning-some-dotted-arguments, 198, 217  
 memo, memo-ize, 320  
 memo-delay, 176  
 memo-function, 141  
     quoting and, 141  
 memory, 132, 183  
     decreasing consumption of, 184  
 MEROON, 417  
 meroon-apply, 338  
 MEROONET, 317, 417  
     field descriptors in, 419  
     first class objects in, 418  
     generic functions in, 418  
     implementation of, 419  
     multiple inheritance in, 418  
     object representation in, 419  
     self-description in, 418  
 meroon-if, 339  
 message  
     sending, 133, 441  
 message-passing style, 114  
 meta-fact, 63  
 metamethod, 361  
 method, 87  
     defining, 88  
     propagating, 445  
 M-expression, 7  
 migration, 184  
 \*minimal-fixnum\*, 376  
 min-max, 111  
 min-max1, 462  
 min-max2, 462  
 mode *autoquote*, 13  
 modifying

    values to define variables, 54  
 monitor, 256  
 Mono-Field-class, 427  
 multimethod, 418, 442  
 multiple dispatch, 417  
 multiple inheritance  
     in MEROONET, 418  
 multiple values, 103  
 multiple worlds, 313  
 mutable binding  
     definition, 26

## N

naive-endogeneous-macroexpander, 322  
 name, 112, 113  
 name  
     call by, 150  
 name conflict  
     in Lisp<sub>2</sub>, 42  
     name spaces and, 43  
     solutions of, 43  
 name space  
     environments and, 43  
 naming conventions, 89  
 NARY-CLOSURE, 214, 230  
 nesting  
     functions, 363  
 new1-evaluate-set!, 463  
 new2-evaluate-set!, 463  
 new-assoc/de, 455  
 newline, xviii  
 new-location, 133  
 new-renamed-variable, 371  
 new-Variable, 405  
 next-method?, 448  
 NfixN, 457  
 NfixN2, 457  
 NIL, 9, 395  
 nil, 26, 136, 195  
 No-Argument, 344  
 No-Free, 365  
 no-more-argument-cont, 105  
 no-more-arguments, 93  
 NON-CONT-ERR, 257, 259  
 normal form  
     definition, 150  
 normal order, 150  
 ν, 152  
 NULL, 382  
 null-env, 91

number?, xviii  
 number  
   in function position, 39  
 number->class, 419  
 number-of, 347

## O

OakLisp, xx, 442  
 Object?, 422  
   better version, 476  
 Object, 88  
 object  
   functional, 442  
   functionally applicative, 40  
 object representation  
   in MEROONET, 419  
 Object-class, 427  
 object->class, 422  
 objectification, 344, 360  
 objectify, 345  
 objectify-alternative, 346  
 objectify-application, 346  
 objectify-assignment, 349  
 objectify-free-global-reference, 348,  
   475  
 objectify-function, 348  
 objectify-quotation, 346  
 objectify-sequence, 346  
 objectify-symbol, 348  
 objectify-variables-list, 348  
 ObjVlisp, 418  
 odd?, 55, 66, 120, 290  
 odd-and-even, 66  
 $\omega$ , 150  
 OO-lifting, 365  
 open coded function, 26  
 operator  
   *J*, 81  
   special, definition, 6  
 order  
   evaluation, 467  
   initialization not specified in Scheme,  
     59  
   initializing variables and, 58  
   macro expansion and, 425  
   not important in evaluation, 62  
   right to left, 467  
 order, normal, 150  
 other-box-ref, 114  
 other-box-set!, 114

other-make-box, 114  
 other-make-named-box, 126

## P

package, 336  
 PACK-FRAME!, 230  
 pair?, 137  
 pair?., xviii  
 pair-eq?, 463  
 parse-fields, 440  
 parse-variable-specifications, 444  
 passwd, 118  
 pc-independent code, 231  
 PC-Scheme, 225  
 Perl, 20, 316  
 physical modifier, 122  
 $\pi$ , 152  
 Planner, 441  
 P-list, 285  
 pointer, 359  
 Poly-Field-class, 427  
 polymorphism, 158  
 POP-ARG1, 244  
 POP-ARG2, 244  
 pop-dynamic-binding, 253, 469, 470  
 POP-ESCAPER, 250  
 pop-exception-handler, 258  
 POP-FRAME!, 243  
 POP-FRAME!0, 243  
 POP-FUNCTION, 229, 244  
 POP-HANDLER, 257  
 popping, 226  
 pp, 373  
 PREDEFINED, 212, 241  
 PREDEFINED0, 241  
 Predefined-Application, 344  
 predefined-fetch, 192  
 Predefined-Reference, 344  
 Predefined-Variable, 344  
 predicate, 422  
 premethod, 447  
 preparation, 312  
 prepare, 314, 316, 471  
 PRESERVE-ENV, 229, 244  
 preserve-environment, 245, 469  
 primitive, 94  
 primitive  
   integrating, 199  
 primitive function  
   definition, 6

primitives, 179  
 procedure?, xviii  
 procedure->definition, 293, 294  
 procedure->environment, 292, 295  
 process-closed-application, 346  
 process-nary-closed-application, 347  
 prog, 71  
 progn, 9  
     role in block, 76  
 Program, 344  
 program?, 274  
 program  
     definition, 147  
     meaning of, 147  
     pretreater, 204  
     pretreating, 203  
     representing, 4  
 program counter, 228  
 programming by continuations, 71  
 promise  
     example, 353  
 propagating  
     methods, 445  
 properties, 455  
 property list, 125  
 protection, 84, 86  
 protect-return-cont, 99  
 pseudo-activation-frame, 297  
 pseudo-activation-frame, 297  
 Pseudo-Variable, 405  
 push-dynamic-binding, 253, 469, 470  
 PUSH-ESCAPER, 250  
 push-exception-handler, 258  
 PUSH-HANDLER, 257  
 pushing, 226  
 PUSH-VALUE, 229, 244

## Q

qar, 463  
 qdr, 463  
 qons, 463  
 quotation?, 274  
 quotation  
     collecting, 367, 372  
     compiling, 375  
 quotation-fetch, 241  
 Quotation-Variable, 368  
 quote, xix, 7, 140, 277, 312  
 quote78, 143  
 quote79, 143

quote80, 143  
 quote81, 143  
 quote82, 143  
 quoting  
     compositional, 140  
     continuations and, 90  
     cycles and, 144  
     definition, 8  
     implicit and explicit, 8  
     memo-functions and, 141  
     semantics of, 140  
     transformations and, 141

## R

r.global, 136  
 r.init, 94, 129, 195, 288  
 random-permutation, 466  
 read, xviii, 139, 272, 312  
 reader  
     for fields, 436  
 read-file, 260  
 recommended reading, xx, 29, 70, 110,  
     181, 222, 268, 309, 357, 415,  
     449  
 recursion, 53  
     local, 56, 57  
     mutual, 55  
     simple, 54  
     tail, 104  
     without assignments, 62  
 redefining  
     macros, 331  
 redex  
     definition, 150  
 reducible form, 197  
 Reference, 344  
 reference implementation, 148  
 reference->C, 380  
 referential transparency, 22  
 reflection, 271, 418  
 ReflectiveClass, 478  
 REFLECTIVE-FIX-CLOSURE, 293  
 reflective-lambda, 295  
 register-class, 439  
 register-CountingClass, 476  
 register-generic, 445  
 register-method, 447  
 regression, infinite, 430  
 Regular-Application, 344  
 REGULAR-CALL, 216, 227, 229

reified-continuation, 460  
 reified-environment, 287  
 relocate-constants!, 264  
 relocate-dynamics!, 265  
 relocate-globals!, 264  
 Renamed-Local-Variable, 371  
 renaming  
   variables, 370  
 renaming-variables-counter, 371  
 repeat, 312  
 repeat1, 473  
 representing  
   Booleans, 8  
   continuations, 87  
   continuations as functions, 81  
   data, 7  
   environments, 13  
   functional applications, 4  
   functions, 15  
   generic functions, 442  
   programs, 4, 7  
   special forms, 6  
   values, 133  
   variables, 4  
 rerooting, 25  
 RESTORE-ENV, 229, 244  
 restore-environment, 245, 469  
 restore-stack, 247  
 resume, 89, 90, 92, 93, 95–99, 104, 105  
 retrieve-named-field, 436  
 RETURN, 231, 244  
 return, 72, 386  
 return-from, 76, 80, 249  
 return-from-cont, 98  
 re-usability, 22  
 r-extend\*, 186, 288, 348  
 r-extend, 348  
 run, 229, 237  
 run-application, 266  
 run-clause, 235  
 run-machine, 246, 259  
 RunTime-Primitive, 350

## S

s.global, 136  
 s.init, 133  
 s.lookup, 24, 452  
 s.make-function, 24, 452  
 s.update!, 24, 452  
 save-stack, 247  
 COMMON LISP  
   lexical escapes in, 76  
 EPS, 318  
 GNU EMACS LISP, 20  
 ILOG TALK, 360  
 LLM3, 148  
 PL, 148  
 PSL, 148  
 PVM, 148  
 VDM, 148  
 scan-pair, 377  
 scan-quotations, 375  
 scan-symbol, 377  
 ->Scheme, 474  
 Scheme, xx  
   grammar of, 272  
   initialization order not specified, 59  
   lexical environment in, 44  
   semantics of, 151  
   uninitialized bindings not allowed,  
     60  
 scheme.h, 390  
 scheme.h include file, 373  
 Scheme->C, 360  
 Scheme->C-names-mapping, 374  
 schemelib.c, 395  
 SCM, 392  
 SCM, 414  
 SCM\_DefinePredefinedFunctionVariable,  
   395  
 SCM\_DefineGlobalVariable, 373  
 SCM\_invoke\_continuation, 408  
 SCM\_2bool, 394  
 SCM\_allocate\_box, 381  
 SCM\_Car, 394  
 SCM\_Cdr, 394  
 SCM\_CheckedGlobal, 380, 395  
 SCM\_close, 385, 397  
 SCM\_CLOSURE\_TAG, 397  
 SCM\_cons, 385  
 SCM\_Content, 381, 396  
 SCM\_DeclareFunction, 386  
 SCM\_DeclareLocalDottedVariable, 386  
 SCM\_DeclareLocalVariable, 386  
 SCM\_Define, 393  
 SCM\_Define... , 379  
 SCM\_DefineClosure, 386, 396  
 SCM\_DefinePair, 379  
 SCM\_DefineString, 379  
 SCM\_DefineSymbol, 379  
 SCM\_error, 398



- SCM\_false, 379, 382
- SCM\_Fixnum2int, 391
- SCM\_FixnumP, 391
- SCM\_header, 392
- SCM\_Int2fixnum, 379, 391
- SCM\_invoke, 382, 396, 399, 403
- SCM\_nil, 379
- SCM\_nil\_object, 395
- SCM\_object, 391
- SCM\_Plus, 385
- SCM\_print, 387
- SCM\_STACK\_HIGHER, 403
- SCM\_tag, 392
- SCM\_true, 379
- SCM\_undefined, 395
- SCM\_unwrapped\_object, 392
- SCM\_unwrapped\_pair, 393
- SCM\_Wrap, 379, 394
- SCMq-, 409
- SCMref, 393
- scope
  - bindings and, 68
  - definition, 20
  - lexical, 68
  - macros and, 333
  - shadowing and, 21
  - textual, 68
- search-dynenv-index, 253, 469
- search-exception-handlers, 258
- selector, 423
  - careless-, 428
- self-, 396
- self-description
  - in MEROONET, 418
- semantics, 147
  - algebraic, 149
  - axiomatic, 149
  - definition, 148
  - denotational, definition, 149
  - meaning and, 148
  - natural, 149
  - of quoting, 140
  - operational, 148
  - Scheme, 151
- send, 441, 446
- sending
  - messages, 133, 441
- SEQUENCE, 214, 226, 233
- Sequence, 344
- sequence, 9, 90, 129, 154, 178, 209, 214, 382
- set!, xix, 7, 11
- set
  - as prefix, 438
  - as suffix, 438
- set of free variables, 365
- set-car!, xviii, 124
- set-cdr!, xviii, 27, 124, 137, 398
- set-Class-name!, 441
- set-Class-subclass-numbers!, 441
- set!-cont, 92
- SET-DEEP-ARGUMENT!, 240
- set-difference, 204
- (setf symbol-function), 285
- (setf symbol-value), 285
- set-field-value!, 438
- SET-GLOBAL!, 240
- SET-GLOBAL!-code, 264
- set-global-value!, 283
- setjmp, 402
- set-kdr!, 122
- setq, 11
- SET-SHALLOW-ARGUMENT!, 228, 240
- SET-SHALLOW-ARGUMENT!2, 240
- set-variable-value!, 301
- set-winner!, 112
- sg.current, 192
- sg.current.names, 467
- sg.init, 192
- sg.predef, 350
- sh, 313, 413
- shadowable-fetch, 299
- shadow-extend\*, 298
- shadowing
  - scope, 21
- SHADOW-REF, 299
- shallow binding
  - definition, 23
  - implementation techniques, 25
- SHALLOW-ARGUMENT-REF, 212, 237, 239, 240
- SHALLOW-ARGUMENT-REF0, 240
- SHALLOW-ARGUMENT-REF1, 240
- SHALLOW-ARGUMENT-REF2, 240
- SHALLOW-ARGUMENT-REF3, 240
- SHALLOW-ARGUMENT-SET!, 213, 228
- \*shared-memo-quotations\*, 141
- shell, 313
- SHORT-GOTO, 242
- SHORT-JUMP-FALSE, 242
- SHORT-NUMBER, 242
- side effect, 111, 121

- $\sigma$ , 152
  - signal-exception**, 258
  - signature, dotted, 443
  - silent
    - variable, 22
  - simulating
    - dotted pairs, 122
    - shallow binding, 24
  - simultaneous-eval-macroexpander**, 331
  - single-threaded, 169
  - SIOD, 414
  - size, 423
  - size\_, 396
  - size-clause, 235
  - Smalltalk, 272, 420, 441
  - some-facts**, 322
  - space
    - dynamic variable, 44
    - function, 44
    - lexical variables, 44
    - of macros, 327
  - special form, 311, 312
    - catch, 74
    - function, 21
    - if, 8
    - quote, 7
    - throw, 74
    - definition, 6
    - representing, 6
  - special operator
    - definition, 6
  - special variable, 22
  - special-begin, 350
  - special-define-abbreviation, 353
  - special-eval-in-abbreviation-world, 353
  - special-extend, 48
  - \*special-form-keywords\***, 350
  - special-if, 350
  - special-lambda, 350
  - special-let-abbreviation, 354
  - special-quote, 350
  - special-set!, 350
  - special-with-aliases, 354
  - Sqil, 238, 360
  - sr.init, 195
  - sr-extend\*, 185
  - sr-extend, 350
  - \*stack\***, 226
  - stack, 226
    - C, 403
    - evaluation, 71
    - popping, 226
    - pushing, 226
  - \*stack-index\***, 226
  - stack-pop, 226
  - stack-push, 226
  - stammer, 306
  - stand-alone-producer, 204, 467
  - stand-alone-producer7d, 246
  - standard header file scheme.h, 373
  - \*starting-offset\***, 422
  - static error, 194, 274
  - static-wrong, 194
  - STORE-ARGUMENT**, 216, 226, 227, 233
  - string?, xviii
  - string, xviii
  - string-ref, xviii
  - string-set!, xviii
  - string->symbol, xviii, 284
  - substituting, 150
  - supermethod, 447
  - symbol?, xviii
  - symbol, 125
    - variables and, 4
  - symbol table, 213, 467
  - symbol-concatenate, 429
  - symbol-function, 285
  - symbol-value, 285
  - symbol->variable, 5
  - syntax
    - data and, 272
    - programs and, 272
    - tree, 372
  - system, 413
- ## T
- T, xx
  - t, 26, 136, 195
  - tagbody, 108
  - Talk, xx
  - target language, 359
  - TEΛOΣ, 87
  - T<sub>EX</sub>, 20
  - the-empty-list, 133
  - the-environment, 303, 307, 472
  - the-false-value, 9
  - the-non-initialized-marker, 61
  - thesis, Church's, 2
  - threaded code, 221
  - throw, 74, 77, 78, 462
  - throw-cont, 96

throwing-cont, 96  
 thunk, 48, 51, 52, 107, 176, 212, 228, 245,  
     454, 461  
 THUNK-CLOSURE, 468  
 toplevel, 306  
 transcode, 139  
 transcode-back, 139, 276  
 transforming  
     quotations and, 141  
 transparency, referential, 22  
 tree  
     syntactic, 372  
 tree-equal, 123  
 tree-shaker, 326  
 TR-FIX-LET, 217, 233  
 TR-REGULAR-CALL, 216, 233  
 Turing machine, 2  
 type, 392  
     of fix, 67  
 typecase, 341  
 type-testing, 199

## U

#<UFO>, 14, 60  
 unchangeability, 123  
 undefined, 472  
 #<uninitialized>, 60, 431  
 uninitialized allocations, 431  
 unique world, 313  
 universal global environment, 116  
 universal language, 2  
 UNLINK-ENV, 244  
 unordered, 466  
 until, 311, 312  
 unwind, 98, 99  
 unwind-cont, 99  
 unwind-protect, 252  
 unwind-protect-cont, 99  
 upate, 129  
 update!, 13, 89, 92, 285  
 update\*, 130  
 update, 129  
 update-generics, 445  
 update-walk!, 361

## V

va\_list, 401  
 \*val\*, 225  
 value, 89

value  
     binding and environments, 43  
     binding and name spaces, 43  
     call by, 11, 150  
     modifying and defining variables, 54  
     multiple, 103  
     of an assignment, 13  
     on A-list of variables, 13  
     representing, 133  
     variables and, 4

Variable, 344

variable  
     address, 116  
     assignment and shared, 112  
     assignments and closed, 112  
     assignments and local, 112  
     bound, definition, 3  
     boxing, 115  
     closed and assignments, 112  
     collecting, 372  
     creating global, 279  
     defining by creating, 54  
     defining by modifying values, 54  
     discriminating, 443  
     dynamic, 44, 79, 86  
     environment, 91  
     floating, 297  
     free, 150  
     free and dynamic scope, 21  
     free and lexical scope, 21  
     free, definition, 3  
     global, 203, 240, 373  
     lexical, 79  
     local, 239  
     local and assignments, 112  
     mutable, 373  
     predefined, 373  
     predefined and assignments, 121  
     renaming, 370  
     representing, 4  
     set of free, 365  
     shared, 113  
     shared and assignments, 112  
     silent, 22  
     special, 22  
     symbols and, 4  
     uninitialized, 467  
     values and, 4  
     values on A-list, 13  
     visible, 68

variable->C, 380

**variable-defined?**, 301, 309, 472  
**variable-env**, 91  
**variable->Scheme**, 474  
**variables-list?**, 274, 309  
**variable-value**, 301, 472  
**variable-value-lookup**, 299  
**vector?**, xviii  
**vector**, xviii, 183  
**vector-copy!**, 247  
**vector-ref**, xviii  
**vector-set!**, xviii  
**verifying**  
     arity, 244  
**virtual machine**, 148  
**virtual reality**, 54  
**Vlisp**, 249  
**vowel1<=**, 142  
**vowel2<=**, 142  
**vowel3<=**, 143  
**vowel<=**, 142

## W

**W**, 63  
**WCL**, 360  
**when**, 307, 334  
**while**, 311, 312, 320  
**&whole**, 320  
**winner**, 112  
**with-quotations-extracted**, 319  
**With-Temp-Function-Definition**, 370  
**world**  
     multiple, 313  
     unique, 313  
**writer**  
     for fields, 437  
**write-result-file**, 260  
**wrong**, 6, 41, 154, 194

## X

**xscheme**, 268