Index

accumulators, 101

booleans, 7

built-in functions
- *: multiplication function, 19
- +: addition function, 19
- -: subtraction function, 19
- /: division function, 19
- <: less than, 43
- <=: less than or equal to, 43
- =: equal to, 43
- >: greater than, 43
- =>: greater than or equal to, 43
- abs, 145
- append, 147
- apply, 176
- boolean?: type checker, 41
- cons: create a cons cell, 131, 133
- eq?, 44
- equal?, 205
- eval: evaluation function, 21
- even?: is its input even?, 145
- even?even?, 145
- first: access first element of a list, 131
- format, 217
- fourth, 133
- integer?, 21, 111
- length, 138
- list, 135, 162
- list-ref, 139
- list?: type checker, 41
- load: for loading files, 62
- make-vector, 200
- member, 140
- not: flip boolean value, 86
- null?: type checker, 41
- number?: type checker, 41
- printf, 22
- procedure?: type checker, 41
- quotient: integer division, 21, 30
- random: pseudo-random number gen., 118
- remainder: integer division, 21, 30
- remove, 147
- rest: access the rest of a list, 131
- reverse, 153
- second, 133
- sort, 165
- string-append, 217
- string?: type checker, 41
- symbol?: type checker, 41
- third, 133
- vector, 199
- vector-length, 201
- vector-ref, 200
- vector-set!, 206
- void, 22, 142
- void?: type checker, 41, 60

character sequences, 7, 8
comments, 63
computational model, 2, 15, 30
cons cell, 165, see also pair
contracts, 20
- as comments in a Scheme program, 64
data, 7
- non-primitive, 10
- primitive, 7
data structures, 165, 209
- field-based, 209
- fields, 165
- implemented using vectors, 209, 210
- index-based, 209
- recursively defined, 209
data type, 7
- booleans, 8
- functions, 19
- non-empty lists, 25
- null, 9
- numbers, 7
- strings, 10
- symbols, 9
- void, 9, 22, 142
debug, 3
deep lists, see also hierarchical lists, nested lists
Default Rule for evaluating non-empty lists, 26, 29
- applying functions to inputs, 27, 30
- generating errors, 29
- on lists of lists, 27
denotation
single arrow notation (→), 8, 16
destructive functions, 3
DrScheme, 15
   interactions window, 15, 16
   prompt, 16
   the Run button, 63
elements
   of non-empty lists, 25
empty list, 9
environments, 12
escape sequences, 59
evaluation, 11, 15
   double arrow notation (⇒), 12, 16
   of booleans, 11
   of functions, 21
   of non-empty lists, 26
   of numbers, 11
   of special forms, 33
   of strings, 11
   of symbols, 12
   of the empty list, 11
   of the void datum, 11
evaluation function, 11
   as a built-in function, 21
expressions, 7
   primitive data, 7
   primitive data expressions, 7
function-call expression, 75
functions, 2, 11, 47
   built-in, 9, 19
   defining vs. applying, 47
   factorial, 76
   function application using the Default Rule, 27
   recursive, 75
   tail-recursive, 99
Global Environment, 13, 16
   and built-in functions, 19
hierarchical lists, see also deep lists, nested lists
identity function, 11
insertion-sort algorithm, 156
keyword symbols, 33, see also special forms
   else, 83
   redefining, 36
Lambda Calculus, 2
lambda function
   applying to inputs, 52
   lazy evaluation, 67
   of the and special form, 88, 94
   of the cond special form, 84
   of the if special form, 68
   of the or special form, 94
local environment, 52, 115
local variable, 115
local vs. global environments, 53, 118
nested lists, see also hierarchical lists, deep lists
   nested if expressions, 81
   equivalence in terms of cond, 82
non-destructive functions, 3
non-empty lists, 25
   syntax and semantics, 25
non-strict truth values
   with and, 88, 89
   with if, 69
   with not, 86
   with or, 89
numbers, 7
pair, 165, see also cons cell
predicates, 41
   defined using boolean operators, 91
   type-checker predicates, 41
program, 7
recursion, 75
   accumulators, 101
   base case, 75
   recursive case, 76
   tail recursion, 97
   wrapper functions, 109
sample data structures
   date, 213
dice-dice, 221
sample functions
   add-squares, 138
   approx-e-acc, 108
   approx-e-wr, 110
   approx-pi-acc, 106
   approx-pi-wr, 110
   bad-reverse, 154
cafe-open?, 91
cafe-open?-alt, 92
civil-to-mil-hours, 90
compare-sorting-algs, 164
cone-volume, 50
continue!, 223
continue-wr!, 226
convert-to-boolean, 72
date-day, 211
Introduction to Computer Science via Scheme  © 2018 Luke Hunsberger  Spring 2018

date-month, 211
date-year, 211
date?, 210
dd->string, 221
discriminant, 55
f, 55
facty, 96, 100, 129
facty-acc, 103, 129
facty-iter, 185
facty-v1, 76
facty-v2, 95
facty-verbose, 80
facty-wr, 109
fetch-nth-element, 138
fetch-random-element, 201
flip-coin, 119
forth, 133
how-big, 68
in-class, 90
incr?, 141
init-veck, 206
insert, 156
insert-acc, 158
insertion-sort, 156
is-elt-of-alt?, 141
is-elt-of?, 139
isort-acc, 160
konk, 154
lengthy, 138, 172
letter-grade, 81
letter-grade-v2, 82
list-down-to-zero-acc, 151
list-down-to-zero-wr, 152
list-one, 134
list-two, 134, 162
make-date, 210
map, 146
map*, 173
mappy, 145
merge, 160, 162
merge-acc, 163
merge-sort, 160, 161
mult-all, 135
mult-all-acc, 149
mult-all-wr, 150
my-and, 88
my-apply, 176
my-dotimes, 193
my-multy, 175
my-multy-multy, 175
my-multy-multy-v2, 176
my-not, 87
my-while, 186
new-game, 221
num-leaf-items*, 172
one-elt-list?, 133
print-date, 214
print-func-vals, 100
print-in-reverse, 204
print-n-dashes, 98, 187
print-vector, 203
print-vector-from, 202
print-vector-wr, 202
quick-list?, 171
rect-area, 50
remover, 146
reset!, 223
reset-wr!, 226
reversey, 153
roll!, 223
roll-wr!, 226
save!, 223
save-wr!, 226
sekund, 133
set-date-month!, 211
set-date-year!, 211
set-date-year!, 211
show-game, 221
sphere-volume, 52
split, 160
split-acc, 162
square, 49, 54, 57
sum-all*, 170
sum-iter, 185
sum-squares, 97
sum-squares-acc, 105
sum-to-n, 80, 101
sum-to-n-acc, 102, 189
tester, 62
and quote, 62
thurd, 133
toss-die, 119
transfer-all, 152, 159
vector-equal?, 205
vector-swap!, 207
verbose-func, 61
well-formed-list?, 168
semantics, 1
side effect
  vs. output value, 9
side effect, of the printf function, 22
side effects, 2, 56
  of define special form, 34
  of printf, 30
special forms, 33
  and, 87
cond, 81
define, 34
   semantics of, 34
   side effect of, 34
   syntax of, 34
define-struct, 213
   semantics of, 213
   syntax of, 213
dolist, 194
dotimes, 192
if, 67
lambda, 48
   semantics of, 51
   syntax of, 50
let, 115
   in terms of lambda, 123
   semantics of, 116
   syntax of, 115
let*, 115, 124
   equivalence to nested lets, 126
   semantics of, 125
   syntax of, 124
letrec, 115, 127
   semantics of, 127
   syntax of, 127
or, 89
quote, 37
   alternate syntax ('), 38
   semantics of, 37
   syntax of, 37
time, 163
when, 72
while, 183
syntax, 1

truth values, 7

#<undefined>, 127

variables, 12

vectors
   immutable, 198
   mutable, 199, 200
   the pound syntax, 198
   the #(...) syntax, 198

void datum, 9, 22, 34, 77