

# Index

- abstract types, 71, 74, 212
- algebraic data types, 104
- alignment, formatting with `printf`, 156
- animation
  - creating with mixins, 256
  - displaying animated shapes, 258
- anonymous functions, 24, 34
- Arg module, 296
- arguments
  - argument types, 285
  - default arguments, 286
  - defining custom types, 286
  - inference of, 46
  - labeled arguments, 17, 42
  - optional arguments, 45, 286
  - sequences of, 288
- Array module
  - Array.blit, 138
  - Array.set, 138
- array-like data, 138
- arrays
  - imperative programming and, 24
  - memory representation of, 428
  - pointers and, 411
- assert directive, 126
- association lists, 263
- AST (abstract syntax-tree), 362, 424, 449, 465
- Async library
  - benefits of, 297
  - DuckDuckGo searching example, 309
  - echo server example, 303
  - exception handling in, 313
  - finalizers, 445
  - ivars, 301
  - system threads and, 321
  - timeouts and cancellations, 318
  - try\_with, 313
- Async\_rpc, 186
- ATDgen Library
  - annotations in, 355
  - basics of, 354
  - compiling specifications to OCaml, 355
  - example of, 357
  - installation of, 354
- autocompletion, 294
- backtraces, 130, 478
  - capturing the most recent, 130
  - control recording of, 131
  - raising exceptions without one, 131
- Base standard library, 3, 9
  - finding with `ocamlfind`, 68
  - imperative dictionaries in, 134
- Base.Poly, 63, 270
- bash autocompletion, 294
- benign effects
  - laziness, 145
  - memoization, 146
- best-fit allocation, 440
- bigarrays, 139, 433
- Bigstring module, 433
- binary methods, 247
- binary numbers, formatting with `printf`, 156
- bind function, 122, 299
- bindings
  - top-level, 31
- BLAS mathematical library, 433
- blocking IO, 298
- blocks (of memory), 425
- boxing (of values), 425
- bytecode compiler
  - compiling and linking code, 474
  - instruction set for, 473
  - tools used, 472
  - values stored by, 473
- C object files, 475
- C99 scalar types, 410
- Camlimages library, 228
- Camomile unicode parser, 370
- cancellations in Async, 318
- catch-all cases, 113
- classes
  - basic syntax for, 237
  - benefits of, 228
  - binary methods for, 247
  - class parameters and polymorphism, 238
  - class types, 244
  - inheritance in, 243

- initializersfor, 254
- multipleinheritance in, 254
- object types as interfaces, 239
- open recursion in, 245
- private methods for, 246
- virtualclasses, 251
- client queries, 310
- Cmddlinerlibrary, 296
- cmi files, 461, 476
- cmt files, 465
- cmti files, 465
- cmx files, 476
- cohttp library, 309
- combinators, 177
  - functionalcombinators, 280, 349
  - in Yojson library, 349
- combinators, implementing with GADTs, 180
- Command library, 280
- command-line parsing
  - alternatives to Command library, 296
  - argument types, 285
  - autocompletion with bash, 294
  - basic approach to, 280
  - Command library, 280
  - labeled flags and, 289
  - running async commands, 309
  - subcommand grouping, 291
- Command.basic, 281
- Command.group, 291
- commas vs. semicolons, 18
- compaction, 442
- Comparable module, 208, 267
- comparator
  - Comparator module, 266
  - comparator witness, 264
- compilationprocess
  - compiler source code, 448
  - diagram of, 448
  - fast native code, 476
  - file extensions, 486
  - parsing source code, 449
  - phases of, 424
  - portable bytecode, 472
  - static type checking, 454
  - typed syntax tree, 465
  - untyped lambda form, 468
- compilationunits, 448, 461
- compile-time static checking, 346, 425, 456
- compiler
  - enabling and disabling warnings, 88
- compilers
  - bytecode vs. native code, 448
- completion handlers, 296
- concrete types, 73
- concurrent programming, 297, 313
- Container.Make, 208
- context-free grammar, 364
- contravariance, 231
- copying collection, 436
- Core standard library, 9
- covariance, 231
- Cryptokitlibrary, 228
- Ctypes library
  - arrays, 418
  - build directives for, 408
  - installation of, 405
  - lifetime of allocated Ctypes, 421
  - terminal interface example, 406
- curly braces ( { } ), 364
- curried functions, 36
- custom heap blocks, 433
- custom runtime mode, 474
- cyclic data structures, 143
- cyclic dependencies, 81
- data serialization
  - with JSON, 345
  - with s-expressions, 374
- data structures
  - arrays, 24
  - cyclic, 143
  - immutable, 24
  - key-value pairs, 263
  - lists, 17
  - mutable record fields, 25
  - options, 21
  - pattern matching and, 52
  - primitive mutable data, 138
  - recursive, 108
  - tuples, 16
- datatypes
  - covariant, 163
  - defining new, 22
  - fixed vs. variable structure of, 87
  - locally abstract types, 212
  - nullable, 22
  - record types, 22
  - variant types, 23, 99
- debugging
  - activating debug runtime, 485
  - backtraces, 130
  - interactive debuggers, 480
  - native code binaries, 480
  - Obj module warning, 430
  - s-expressions, 382
- decimals, formatting with printf, 156
- default arguments, 286
- Deferred.bind, 299
- Deferred.both, 318
- Deferred.never, 307
- Deferred.peek, 298
- Deferred.t, 298
- denial-of-service attacks, avoiding, 274

- dependencies, cyclic, 81
- depth subtyping, 229
- destructive substitution, 200
- dictionaries, imperative, 134
- dispatching, dynamic vs. static, 251
- documentation, generating from interfaces, 450
- Domain specific language, 354
- doubly-linked lists, 141
- Doubly-linked module, 144
- down casting, 234
- DuckDuckGo search engine
  - additional libraries needed, 309
  - exception handling in, 316
  - executing an HTTP client query in, 310
  - parsing JSON strings in, 310
  - URI handling in, 309
- dune, 68
  - initializing a project, 389
- dune-project, 68
- duplicates, removing, 40, 58
- dynamic dispatch, 251
- dynamic programming, 148
- dynamic type checking, 234, 249, 425
- echo servers, 303
- edit distance, 147
- elements
  - combining with List.reduce, 58
  - defining new, 143
  - inserting in lists, 143
  - partitioning with List.partition\_tf, 59
  - setting with Array.set, 138
  - traversing with iterator objects, 239
- ellipses (...), 226
- end-of-file condition, 304
- eprintf function, 156
- equal (=) operator, 279
- equal equal (==) operator, 279
- equality, tests of, 279
- error handling
  - and imperative data structures, 144
  - combining exceptions and error-aware types, 132
  - error-aware return types, 119
  - exception clean up, 128
  - exception detection, 129
  - exception handlers, 127
  - exception helper functions, 126
  - exceptions, 124
    - option types, 124
    - strategy choice, 133
- error-aware return types, 119, 132
- Error.of\_list, 122
- Error type, 121
- Error.tag, 122
- errors
  - cyclic dependencies, 81
  - detecting with match expressions, 54
  - detecting with type annotations, 456
  - error messages with s-expressions, 382
  - give up on first error approach, 371
  - missing field warnings, 88
  - missing module definitions, 80
  - module type definition mismatches, 80
  - module type mismatches, 79
  - reducing verbosity in, 464
  - runtime vs. compile time, 15
  - syntax errors, 450
  - transformation of, 122
  - type errors, 352
- evaluation, order of, 158
- event loops, 297
- exceptions
  - and error-aware types, 132
  - asynchronous errors, 313
  - backtraces for, 130
  - benefits and drawbacks of, 133
  - catching in match expression, 129
  - catching specific, 129
  - exception clean up, 128
  - exception handlers, 127
  - helper functions for, 126
  - in concurrent programming, 313
  - search engine example, 316
  - textual representation of, 125
  - usefulness of, 124
  - vs. type errors, 15
- exhaustion checks, 102
- existential type, 166
- existential types, 176
- exn type, 125
- explicit subtyping, 454
- expressions, order of evaluation, 158
- extensible variants, 250
- external libraries
  - Camlimages, 228
  - Cryptokit, 228
  - for graphics, 259
  - interfacing with, 140
- external memory, 433
- Field module
  - Field.set, 96
  - Field.get, 96
  - Field.name, 96
  - Field.setter, 96
- fields
  - adding to structures, 415
  - field punning, 22, 89
  - first-class fields, 95
  - mutability of, 135, 139
  - reusing field names, 90
- fieldslib, 96
- FIFO (first-in, first-out) queue, 205
- files

- chart of file extensions, 486
- cmi files, 461, 476
- cmt files, 465
- cmii files, 465
- cmx files, 476
- fileIO, 156
- ml files, 456
- mli files, 456
- mll files, 367
- mly files, 363
- multi-file programs, 69
- o files, 476
- relationship with modules, 461
- single-file programs, 66
- filter\_string function, 350
- finalizers
  - for C cleanup functions, 433
  - in garbage collection, 445
- find\_exe function, 125
- first-class fields, 95
- first-class modules
  - alternatives to, 222
  - polymorphism in, 212
  - vs. objects, 228
  - working with, 209
- first-fit allocation, 440
- flag functions, 290
- flags, 289, 465
- flexible find, 175
- floating-point values, 429
- for loops, 28, 140
- foreign function interface (FFI)
  - basic scalar C types, 410
  - basics of, 405
  - C bindings, 422
  - imperative operations and, 140
  - passing functions to C, 419
  - pointers and arrays, 411
  - structs and unions, 414
  - terminal interface example, 406
- format strings, 154
- Fortran libraries, 433
- fprintf function, 156
- frame pointers, 484
- fun keyword, 34
  - currying syntax, 36
  - multi-argument functions, 36
- function keyword, 41, 62
- functional code, 24
- functional combinators, 349
- functional iterators, 242
- functional updates, 93
- functions
  - anonymous functions, 24, 34
  - argument inference, 46
  - curried functions, 36
  - declaring with function keyword, 41
  - defining, 11, 41
  - exception warnings for, 125
  - hash functions, 274
  - higher-order and labels, 44
  - iteration functions, 144
  - labeled arguments, 42
  - member functions, 350
  - multi-argument functions, 36
  - name mangling of, 480
  - non-returning, 305
  - optional arguments, 45
  - passing to C, 419
  - prefix and infix operators, 38
  - recursive functions, 37
  - to\_init function, 350
  - to\_string function, 350
  - with multiple arguments, 11
- functors
  - basic mechanics of, 191
  - benefits of, 191
  - interval computation with, 193
  - module extension with, 205
- GADT, 166
- garbage collection
  - and boxed values, 425
  - finalizer functions, 445
  - generational, 436
  - mark and sweep collection, 435, 438
  - of allocated C types, 421
  - of longer-lived values, 438
  - of short-lived values, 436
  - opaque bytes and, 427
- Gc module, 436
- gdb debugger, 480
- Generalized Algebraic Data Type, 166
- generational garbage collection, 436
- generational hypothesis, 436
- geometric shapes, 251
- GitHub API, 354, 357
- GNU debugger, 480
- gprof code profiler, 483
- grammar, 364
- graphics libraries, 259
- hash tables
  - basics of, 263, 274
  - polymorphic hash function, 276
  - time complexity of, 274
  - vs. maps, 276
- Hashable.Make, 208
- Hashtbl module, 274
- heaps
  - definition of, 435
  - heap blocks, 435
  - Heap\_block module, 445
  - major heaps, 438

- minor heaps, 436
  - regions of, 436
- hex numbers, formatting with `printf`, 156
- higher-order functions, and labels, 44
- Hindley-Milner algorithm, 456
- HTML generators, 451
- HTTP client queries, 310
- identifiers
  - adding to modules, 77
  - dealing with multiple, 74
  - open modules and, 76
- imperative programming
  - arrays, 24
  - benefits of, 134
  - benign effects and, 145
  - doubly-linked lists, 141
  - drawbacks of, 144
  - for and while loops, 28
  - imperative dictionaries, 134
  - input and output, 153
  - mutable record fields, 25
  - order of evaluation, 158
  - primitive mutable data, 138
  - ref type, 26
  - side effects
    - weak polymorphism, 159
- In\_channel module, 153
- In\_thread module, 321
- infix operators, 38
- inheritance, 243, 254
- initializers, 254
- install keyword, 291
- installation instructions, 5
- integers, 425, 428
- interactive input
  - concurrent programming for, 297
  - prompts for, 293
- interfaces
  - foreign function interface (FFI), 405
  - generating documentation from, 450
  - hiding implementation details with, 71
  - object types as, 239
  - synonyms for, 71
  - with OCaml binaries, 424
- intergenerational pointers, 443
- interval computation
  - abstract functor for, 196
  - comparison function for, 195
  - generic library for, 193
  - multiple interfaces and, 201
- invariance, 231
- IO operations
  - copying data, 303
  - file IO, 156
  - formatted output, 154
  - terminal IO, 153
- Iobuf module, 433
- irrefutable patterns, 34, 87
- iteration functions, 144
- iterators, 239
- ivars, 301
- js\_of\_ocaml library, 259
- JSON data
  - automatic mapping of, 354
  - basics of, 345
  - benefits and drawbacks of, 346
  - constructing values, 351
  - nonstandard extensions for, 353
  - parsing with Yojson, 346
  - selecting values from, 348
- Just-in-Time dynamic patching, 425
- kernel-level threads, 321
- key
  - value pairs, 345
- key-value pairs, 263
- label punning, 42, 89
- labeled arguments, 17, 42, 46
- Lab1GL library, 259
- Lab1gtk library, 259
- Lacaml library, 433
- lambda form code
  - basics of, 468
  - pattern matching benchmarking, 470
  - pattern matching optimization, 468
- LAPACK bindings, 140
- LAPACK mathematical library, 433
- late binding, 228
- laziness, 145
- lazy keyword, 145
- let binding
  - local, 27
- let bindings, 10
  - scope of, 31
  - top-level, 31
- let rec, 143, 151
- let syntax
  - function definition with, 11
  - functions and, 35
  - nested bindings, 32
  - nonrecursive vs. recursive functions, 38
  - pattern matching, 33
  - top-level bindings, 31
- let%bind, 300
- let%map, 300
- Levenshtein distance, 147
- lexers
  - optional OCaml code for, 367
  - recursive rules, 370
  - regular expressions collection, 368
  - rules for, 368
  - specification of, 367
  - Unicode parsing, 370

- lexical analysis (lexing), 362
- libasmrun.a library, 485
- libffi library, 405
- libraries
  - Camlimages, 228
  - Cryptokit, 228
  - defining libraries, 392
  - for graphics, 259
  - interfacing with external, 140
- limitations
  - let rec, 152
- linear algebra, 433
- linear mixins, 257
- linkpkg, 68
- List module
  - creating tables with, 55
  - List.append, 59
  - List.filter, 58
  - List.fold, 56
  - List.init, 60
  - List.map, 55
  - List.map2\_exn, 56
  - List.partition\_tf, 59
  - List.reduce, 58
  - String.concatand, 57
- List.Assoc module
  - List.Assoc.add, 66
  - List.Assoc.find, 66
- List.dedup\_and\_sort, 40
- lists
  - adding new bindings in, 66
  - association lists, 263
  - combining, 59
  - combining elements in, 58
  - computing length of, 60
  - doubly-linked lists, 141
  - duplicate removal, 40, 58
  - extension of, 51
  - extracting data from, 51
  - filtering values in, 58
  - finding key associations in, 66
  - generation of, 50
  - List module, 55
  - memory representation of, 430
  - operator :, 17, 50
  - partitioning elements in, 59
  - pattern matching, 18
  - recursion, 21
  - recursive list functions, 20
  - structure of, 50
- lit suffix, 353
- local open, 12
- local opens, 76
- locally abstract types, 172, 212
- looping constructs, 140
- LR(1) grammar, 364
- main function, 67
- major heaps
  - allocating on, 439
  - controlling growth of, 439
  - garbage collection in, 438
  - heap compaction, 442
  - intergenerational pointers in, 443
  - marking and scanning, 441
  - memory allocation strategies, 440
- malloc(3), 439
- Map.to\_tree, 273
- mapping
  - complex values with views, 413
  - of JSON to OCaml types, 354
  - of OCaml types to runtime values, 425
- maps
  - basics of, 263
  - polymorphic comparison in, 269
  - tree structure, 273
  - vs. hashtables, 276
- mark and sweep garbage collection, 435, 438
- match expressions, 51, 54
- match%map, 317
- MD5 one-way cryptographic hash function, 280
- member function, 350
- memoization
  - benefits and drawbacks of, 147
  - example of, 147
  - of function, 146
  - recursive, 148
- memory
  - allocation for pointers, 412
  - and allocated Ctypes, 421
  - layout for structs, 423
  - major heap allocation strategies, 440
  - managing external, 433
  - memory management, 435
  - reducing fragmentation of, 440, 442
- menhir dune stanza, 367
- Menhir parser generator, 364
  - built-in rules of, 366
  - error handling in, 371
  - invoking, 367
  - left-recursive definitions, 366
  - vs. ocaml yacc, 361
- methods
  - and virtual classes, 251
  - binary methods, 247
  - method overloading, 249
  - private methods, 246
- minor heaps
  - allocating on, 437
  - garbage collection in, 436
  - setting size of, 437
- mixin patterns, 255
- ml files, 456
- mli files, 456

- mll files, 367
- module keyword, 209
- modules
  - basics of, 69
  - benefits of, 454
  - cyclic dependencies, 81
  - defining search paths, 461
  - first-class modules, 209
  - hiding implementation details, 71
  - in static type checking, 454
  - including, 77
  - missing definitions in, 80
  - module type, 71
  - naming of, 70
  - nested modules, 74
  - opening, 76
  - packing together, 462
  - separate compilation in, 460
  - type definition mismatches, 80
  - type mismatches in, 79
  - type-specific functionality in, 205
- Monad.Make, 208
- monads, 300
- monitors, 314
- multi-argument functions, 11, 36
- multicore, 321
- multiple inheritance
  - displaying animated shapes with, 258
  - mix in pattern and, 255
  - name resolution in, 254
- mutable data, 138
- mutable record fields, 25, 94
- mutexes, 323
- name mangling, 480
- narrowing, 234, 249
- native-code compiler
  - benefits of, 476
  - debugging binaries, 480
  - embedding code in C, 485
  - inspecting assembly output, 477
  - performance profiling, 483
- Ncurses terminal toolkit, 405
- negation operators, 39
- nested let binding, 32
- nested modules, 74
- never\_returns, 305
- next-fit allocation, 440
- non-terminal symbols, 364
- Nothing.t, 185, 306
- o files, 476
- OAuth web protocol, 354
- Obj module, 430
- object-oriented programming (OOP), 223
- object\_fields, 365
- objects
  - benefits and drawbacks of, 228
  - coercions, 234
  - immutable, 227
  - in object-oriented programming, 223
  - in OCaml, 224
  - narrowing and, 234
  - object types as interfaces, 239
  - polymorphism of, 225
  - subtyping, 236
  - subtyping and, 228
- OCaml
  - benefits of, 2
  - code examples for, 6
  - history of, 2
  - installation instructions, 5
  - key features of, 1
  - operating system support, 5
  - third-party libraries for, 4
- OCaml toolchain
  - dune, 68
  - ocamlc, 68, 472
  - ocamldoc, 450
  - ocamldoc-generators, 451
  - ocamlfind, 68
  - ocamllex, 367, 370
  - ocamlloginfo, 461
  - ocamlopt, 476
  - ocamlrun, 472
  - overview of, 447
- ocaml-getopt, 296
- OCAMLRUNPARAM, 436
- ocamyacc parser generator, 361
- opam package manager, 291
  - creating a local switch, 389
  - initializing, 388
  - listing switches, 388
  - local switch, 389
- opaque bytes, 427
- open hashing, 135
- open object types, 225
- open recursion, 228, 245
- operators
  - :: operator, 17, 50
  - application operator, 41
  - controlling pointers, 419
  - negation operators, 39
  - prefix and infix operators, 38
  - reverse application operator, 40
  - subtraction operators, 39
- optional arguments
  - and default arguments, 286
  - explicit passing of, 45
  - inference of, 46
- options, 21
- or-patterns, 37, 188
- order of evaluation, 158
- Out\_channel module
  - Out\_channel.stderr, 153

- Out\_channel.stdout, 153
- parallelism, 321
- parametric polymorphism, 14, 27
- parsing
  - lexer and parser composition, 371
  - lexer definition, 367
  - of source code, 449
  - parser definition, 363
  - parser generators, 361
- partial application, 36, 40, 161
- pattern matching
  - and exhaustiveness, 87
  - and let, 33
  - benchmarking of, 470
  - catch-all cases, 102, 113
  - extracting data with, 51
  - fundamental algorithms in, 470
  - in lists, 18
  - optimization in lambda form code, 468
  - terser and faster patterns, 62
  - vs. lexing rules, 369
- performance profiling, 483
- phantom type, 414
- phantom types, 169
- phys\_equal function, 279
- physical equality, 279
- pipes, 307
- point-free programming, 351
- pointers
  - allocating typed memory for, 412
  - intergenerational pointers, 443
  - operators controlling, 419
  - support for in C types, 411
  - values for, 425
  - word-aligned, 426
- polymorphic compare, 63
- polymorphic comparisons, 269, 477
- polymorphic recursion, 173
- polymorphic variant types
  - and catch-all cases, 113
  - automatic inference of, 112
  - basic syntax of, 111
  - drawbacks of, 117
  - in JSON data, 347
  - memory representation of, 431
  - type checking and, 352, 457
  - upper and lower bounds of, 112
  - vs. ordinary variants, 113, 117
- polymorphism
  - class parameters and, 238
  - in first-class modules, 212
  - in locally abstract types, 213
  - in objects, 225
  - polymorphic equality, 249
  - row polymorphism, 226, 235
  - weak polymorphism, 159
- POSIX functions, 411
- PPX syntax extensions, 378
- ppx\_assert, 327
- ppx\_compare, 271, 378
- ppx\_fields\_conv, 95
- ppx\_inline\_test, 326
- ppx\_jane, 121, 151, 202, 214, 271, 336
- ppx\_let, 300
- ppx\_sexp\_conv, 202, 214, 271, 377, 378
  - default, 385
  - directives, 383
  - sexp.list, 384
  - sexp.opaque, 383
  - sexp.option, 385
  - sexp\_drop\_default, 386
- ppx\_sexp\_value, 121, 189
- prefix operators, 38
- pretty printers, 375
- primitive mutable data
  - array-like data, 138
  - foreign functions, 140
  - object fields, 139
  - record fields, 139
- principal type checking, 459
- printf function, 154
- private methods, 246
- product types, 104
- profiling, 483
- programming
  - concurrent programming with Async, 297
  - immutable vs. imperative, 24, 134
  - imperative programming, 134
  - language interfaces, 405
  - object-oriented programming (OOP), 223, 237
  - simple standalone example, 29
- programs
  - multi-file programs, 69
  - single-file programs, 66
- protected methods, 246
- punning, 42
- pure code, 24, 134
- qsort binding, 420
- query-handlers
  - and first-class modules, 214
  - dispatching to multiple, 216
  - executing an HTTP client query, 310
  - implementation of, 215
  - loading and unloading of, 219
- raise, 124, 131
- raise\_notrace, 131
- Random module, 28
- Reader module, 303
- rec keyword, 37
- record field accessor functions, 95
- records
  - basic syntax for, 85



- construction of, 89
  - field punning in, 89
  - first-class fields in, 95
  - functional updates to, 93
  - label punning in, 89
  - memory representation of, 428
  - missing field warnings, 88
  - mutable fields in, 94
  - patterns and exhaustiveness in, 87
  - record types, 22, 26
  - reusing field names, 90
- recursion
  - in json types, 347
  - in lexers, 370
  - open recursion, 228, 245
  - tail recursion, 60
- recursive data structures, 108
- recursive functions
  - definition of, 37
  - list functions, 20
- ref cells, 139
- refactoring, 102
- refutation case, 186
- regular expressions, 368
- remembered sets, 443
- remote keyword, 291
- representation types, 249
- Result.t option, 120
- return function, 299
- returning function, 417
- rev\_object\_fields, 365
- RFC3986, 309
- RFC4627, 310, 345
- risky type, 459
- row polymorphism, 226, 235, 457
- runtime exceptions vs. type errors, 15
- runtime memory representation
  - blocks and values, 427
  - custom heap blocks, 433
  - importance of, 424
  - polymorphic variants, 431
  - string values, 432
  - tuples, records, and arrays, 428
  - variants and lists, 430
- s-expressions, 214, 374
  - basic usage of, 374
  - deserializing a type from, 382
  - example of, 121, 201
  - format of, 378
  - generating from OCaml types, 376
  - preserving invariants in, 379
  - specifying defaults in, 385
- scalar C types, 410
- Scheduler.go, 305
- scope, 31
- search engines, 309
- security issues
  - denial-of-service attacks, 274
  - Obj module warning, 430
- sedlex lexer generator, 370
- segfaults, 323
- semantic actions, 364
- semicolons vs. commas, 18
- serialization formats
  - JSON, 345
  - s-expressions, 374
- set types, 265
- Sexplib package, 121
- shadowing, 32
- sharing constraint, 197, 212
- short paths heuristic, 465
- side effects, 145, 159
  - imperative, 138
- signatures
  - abstract types, 71
  - concrete types, 73
- source code
  - parsing of, 449
- sprintf function, 156
- stack frames, 61
- start symbols, 364
- State\_rpc, 186
- static checking, 346
- static dispatch, 251
- strict evaluation, 158
- string matching, 147
- String.concat, 57
- strings
  - concatenation of, 57
  - format strings, 154
  - memory representation of, 432
  - padding of, 57
  - vs. Char.t arrays, 138
- structs and unions
  - field addition, 415
  - incomplete structure definitions, 415
  - memory layout of, 423
  - structure definition, 414
  - time-printing command, 416
- structuralequality, 279
- subcommands, grouping of, 291
- subtraction operators, 39
- subtyping
  - basics of, 228
  - depth subtyping, 229
  - in static type checking, 454
  - variance and, 230
  - vs. row polymorphism, 235
  - width subtyping, 229
- sum types, 104
- syntax errors, 450
- syntax extension

- ppx\_assert, 327
- ppx\_compare, 271, 378
- ppx\_fields\_conv, 95
- ppx\_inline\_test, 326
- ppx\_jane, 121, 151, 202, 214, 271, 336
- ppx\_let, 300
- ppx\_sexp\_conv, 202, 214, 271, 377, 378
- ppx\_sexp\_value, 121, 189
- syntax extension with PPX, 378
- system threads, 297, 321
- tables, creating with List module, 55
- tail calls, 61, 304
- tail recursion, 60
- TCP clients
  - servers, 304
- textwrap library 309
- threads
  - benefits of, 321
  - kernel-level threads, 321
  - locking and, 323
  - preemptive vs. single-threaded programs, 297
  - thread-safety, 323
- thunks, 145, 302
- timeouts, 318
- to\_initfunction, 350
- to\_stringfunction, 350
- tokens, declaration of, 363
- top-level bindings, 31
- top-level printers, 376
- try\_with, 313
- tuples, 16, 353, 428
- type annotations, 47, 456
- type checking, 352, 424
- type definition mismatches, 80
- type errors vs. exceptions, 15
- type inference, 454
  - algorithm basis of, 456
  - benefits of, 352
  - drawbacks of, 454
  - error detection with, 456
  - generic types, 14
  - in static type checking, 454
  - principality checks, 459
  - process of, 13
- type mismatches, 79
- type variables, 14, 160, 226
- type, uninhabited, 185, 306
- typed syntax tree, 465
- unaligned memory access, 426
- unboxed integers, 428
- Unicode, parsing solutions for, 370
- Uniform Resource Identifiers (URIs), 309
- uninhabited type, 185, 306
- unions, 414
- unit tests, 346
- uri library, 309
- Utf8 Unicode codec, 370
- value restriction, 161
- values
  - allocation requests and, 435
  - boxing of, 425
  - copying with Array.blit, 138
  - filtering with List.filter, 58
  - finalizer functions for, 445
  - in JSON data, 345, 351, 365
  - integer vs. pointer, 425
  - mapping complex with views, 413
  - memory representation of, 424
  - selecting from JSON structures, 348
  - stored by bytecode compiler, 473
- variables
  - immutability of, 33
  - pattern matching in, 33
  - scope of, 31
  - shadowing of, 32
- variance, 230
- variant types
  - and recursive data structures, 108
  - basic syntax of, 99
  - combining multiple object types with, 23
  - memory representation of, 430
  - polymorphic, 111
  - usefulness of, 99
  - Yojson support for, 353
- variants and records, 104
- virtual classes, 251
- virtual methods, 251
- weak polymorphism, 159
- while loops, 28, 140
- width subtyping, 229
- word-aligned pointers, 426
- write barriers, 443
- Writer module, 303
- Yojson library
  - combinators in, 349
  - extended JSON format support, 353
  - installation of, 346
  - parsing JSON with, 310, 346