

---

# Index

- `()`, 23
- `(*)`, 3
- `->`, 3, 32
- `*`, 11
- `*)`, 3
- `. [...]`, 27, 186
- `//`, 3
- `///`, 3
- `::`, 71, 80
- `:=`, 182
- `;`, 179
- `;;`, 1
- `@`, 79, 80
- `@ "... "`, 26
- `|>`, 38
- `~`, 35
- `<-`, 176
- `<|`, 38
- `<<`, 32
- accumulating parameter, 206, 209
- add, 106, 114
- aggregation, *see* database
- alias, 182
- and, 141
- anonymous function, *see* function
- append, 79
- append, 254
- argument pattern, 8
- Array, 186
- array, 186
- assignment, 176
- association, *see* operator
- Async, 313
- async, 316
- Async<'a>, 316
- asynchronous computation, *see* computation expression
- asynchronous event queue, 321
- asynchronous operation, 313
- binary tree, 131, 133
- Bind, *see* computation expression
- binding, 14
- bool, *see* type constructor, 62
- builder class, *see* computation expression
- builder object, *see* computation expression
- cache, *see* memory
- cache, 254
- cached sequence, *see* sequence
- cancellation, 319
- cancellation continuation, 319
- cancellation token, 319
- Cartesian product, 44
- Cash register, 116
- cash register, 82
- catch exception, *see* exception
- char, *see* type constructor
- character, 24
- Chinese box, 121
- class, 156
- clause, 8, 74
- closure, *see* function
- coercion rule, 177
- collect, 254
- collection
  - iteration function, 184
- Collection Library
  - List, 93
  - Map, 114
  - Seq, 254
  - Set, 104
- Combine, *see* computation expression
- comment, 3
- compare, 37, 70
- compare function, 36
- comparison constraint, 36
- compilation, *see* module
- composite name, *see* module
- computation
  - delay, 288
  - start, 288
- computation expression, 279
  - Combine, 284
  - asynchronous, 316
  - Bind, 284, 286, 293
  - builder class, 280, 284
  - builder object, 280, 284
  - computation, 280
  - Delay, 284, 290
  - do!, 284
  - For, 282, 284, 291

- if-then, 284
- laws, 291
- let!, 284
- Return, 286, 293
- return!, 284
- ReturnFrom, 284, 286
- translation, 282, 284
- try-finally, 284
- TryFinally, 284
- use, 284
- Using, 284
- While, 284
- while, 284
- Yield, 284, 291
- yield, 284
- yield!, 284
- YieldFrom, 284
- Zero, 284, 286
- conjunction, *see* expression
- cons, 71
- constructor, 58, 59, 123
- contains, 106
- containsKey, 114
- ContentsOf, 176
- context-free grammar, 295
- continuation, 212
- conversion, 235
- count, 106
- Culture dependent information, 232
- customizing equality, *see* override
- customizing hash, *see* override
- customizing indexing, *see* override
- customizing ordering, *see* override
- customizing string, *see* override
- data capture, 221, 224
- data hiding, 150
- data.format, 235
- database, 267
  - aggregation, 272
  - creation, 274
  - join, 271
  - projection, 270
  - selection, 271
  - update, 274
- date, 235, 236
- DateTime, 236
- deadlock, 315
- debugging, 316
- decimal point, 21
- declaration, 1
  - exception, 53
  - execution, 14
  - infix operator, 35
  - local, 54
  - prefix operator, 35
  - recursive function, 8
  - type, 58
  - value, 2
- default value, 179
- Delay, *see* computation expression
- dialogue, 324
- dialogue design, 324
- dictionary imperative, 188
- difference, 108
- digit, 21, 24
- directory
  - current, 264
- disjunction, *see* expression
- division
  - quotient, 15, 22
  - remainder, 15, 22
- dll file, *see* module
- do!, *see* computation expression
- empty list, 68
- entry, *see* map
- enumeration
  - type, 62
- enumeration type, *see* type
- enumerator function, 190, 192
- environment, 14
  - actual, 15
  - Basis Environment, 15
  - local, 50
- equality, 36, 45, 51, 69
- equality constraint, 36
- equality type, *see* type
- error handling, 53
- escape sequence, 25
- Euclid's algorithm, 15
- evaluation, 8
  - environment, 17
  - infinite, 10
  - iterative function, 210
- event-driven program, 324
- exception, 53, 63
  - catch, 63
  - declaration, *see* declaration
  - failwith, 53
  - raise, 53
- execution, 14
- exists, 95, 108, 114
- exponent, 21
- expression
  - conjunction, 23
  - disjunction, 23
  - if-then-else, 28
  - let, 54
  - negation, 23
- expression tree, 137
- failwith, *see* exception
- false, 22, 62
- File, 229, 230
  - Close, 229
  - CreateText, 229

- Delete, 230
- EndOfStream, 229
- Exists, 230
- Flush, 229
- Move, 230
- OpenText, 229
- ReadLine, 229
- Replace, 230
- StreamReader, 229
- StreamWriter, 229
- Write, 229
- WriteLine, 229
- file handling, 230
- filter, 95, 108, 114, 254
- find, 114
- finite tree, 121
- float, *see* type constructor, *see* type constructor
- fold, 97, 108, 114
- foldBack, 97, 108, 114
- For, *see* computation expression
- forall, 95, 108, 114
- format.data, 235
- fprintf, 235
- free-standing program, *see* program
- fs file, *see* module
- fsi file, *see* module
- function
  - anonymous, 4
  - application, 13, 14
  - closure, 34
  - composition, 32
  - declaration, 2
  - higher order, *see* higher-order function
  - partial, 64
  - polymorphic, 48
- function composition, *see* function
- garbage, 200, 202
- garbage collection, 200, 202, 203
- gcd, 15
- grammar, 293, 295
  - left recursion, 297
- guarded pattern, 37, 65
- hash set, 188
- head, 68
- heap, 198
  - limit, 204
- hiding, 150
- higher-order function, 31
  - declaration, 33
- histogram, 188
- html, 339
- HTML-source, 243
- I/O, 229
- identifier, 1
- IDisposable, 232
- IEnumerable, 192
- IEnumerable<'a>, 266
- IEnumerator, 192
- if-then, *see* computation expression
- ignore function, 179
- imperative features, 175
- imperative map, 188
- imperative set, 188
- implementation file, *see* module
- incomplete pattern, *see* pattern
- indentation, 55
- infix, 49
- infix operator, 35
- infix operators, 153
- infix precedence, 49
- init, 254
- initInfinite, 254
- int, *see* type constructor, *see* type constructor
- integer, 21
- intersect, 108
- invariant, 57, 60
- IQueryable<T>, 270
- isSubset, 106
- it, 2
- iteration, 209, 211
- iterative function, 197, 206, 209
- join, *see* database
- key, *see* map
- label, 51
- layered pattern, *see* pattern
- let, *see* expression
- let!, *see* computation expression
- lexicographical ordering, 36, 46
- library
  - construction of, 149
- List, *see* Collection Library
- list, 67, 93
  - collect, 94
  - equality, 69
  - exists, 95
  - expression, 73
  - filter, 95
  - fold, 97
  - foldBack, 97
  - forall, 95
  - map, 94
  - ordering, 70
  - pattern, 71
  - tryFind, 95
- list, *see* type constructor
- list pattern, *see* list
- list recursion, *see* recursion
- List.iter, 184
- List.iteri, 184
- local declaration, *see* declaration
- location, 175
- loop, 211
- main memory, *see* memory
- Map, *see* Collection Library

- map, 113
  - add, 114
  - containsKey, 114
  - entry, 113
  - exists, 114
  - filter, 114
  - find, 114
  - fold, 114
  - foldBack, 114
  - forall, 114
  - key, 113
  - map, 114
  - ofList, 114
  - toList, 114
  - tryFind, 114
  - value, 113
- map, 94, 108, 114, 254
- Map colouring, 110
- map colouring, 85
- Map.iter, 184
- match on type, 160
- maxElement, 106
- memory
  - cache, 311
  - main, 311
- memory management, 198
- minElement, 106
- minus sign, 21
- module, 149, 150
  - compilation, 150, 152
  - composite name, 152
  - dll file, 152
  - fs file, 152
  - fsi file, 152
  - implementation file, 150
  - library, 150, 152
  - parameterized, 157
  - signature file, 150
- monad, 291
- monadic parser, 293
- monomorphic expression, 82
- multi-core processor, 311
- mutable, 175
- mutable record field, *see* record
- mutex, 314
- mutex object, 314
- mutually recursive functions, 141
- mutual recursion, 138
- n*-tuple, *see* tuple
- National string ordering, 232
- natural number, 21
- negation, *see* expression
- Newton-Raphson approximation, 260
- None, 64
- object, 156
- ofList, 106, 114
- operand, 22
- operator, 22
  - association, 23, 35, 71
  - dyadic, 22
  - infix, 23, 35
  - monadic, 22
  - overloading, 29, 31
  - precedence, 23, 35, 71
  - prefix, 35
- option, 64
- or pattern, *see* pattern
- ordering, 36, 46, 70
- overloaded operators, 153
- overloading, *see* operator
- override
  - equality, 159
  - hash, 159
  - index, 161
  - order, 161
  - string, 159
- pair, 11, *see* tuple
- panel, 165
- parameterized type, *see* type
- parser, 293, 298
- partial function, *see* function
- path and uri, 339
- pattern, 4, 8, 51, 123, 129
  - when clause, 37
  - composite, 11
  - constructor, 60
  - guarded, 37
  - illegal, 47
  - incomplete, 47
  - layered, 76
  - list pattern, 74
  - matching, 9, 12
  - or, 6
  - wildcard, 5
- piecewise linear curves, 162
- polymorphic
  - value, 81
- polymorphic function, *see* function
- polymorphic type, *see* type
- polymorphism, 48, 78
  - value polymorphism, 81, 195
  - value restriction, 81, 195
- precedence, *see* operator, 80
- predicate, 23
- prefix operator, 35
- printf, 235
- priority, *see* operator precedence
- process, 312
- product register, 267
- program
  - free standing, 19
- projection, *see* database
- property.setting, 167
- quadruple, *see* tuple

- query expression, 270
  - contains, 273
  - count, 272
  - exactlyOne, 272
  - for, 270
  - join, 271
  - select, 270
  - where, 271
- quotient, *see* division
- raise, *see* exception
- raise exception, 63
- range expression, 73, 266
- read web-page, 243
- real number, 21
- record, 50
  - equality, 51
  - field, 50
  - label, 50
  - mutable field, 180
  - ordering, 51
  - pattern, 51
  - type declaration, 180
- record field, 50
- record type, *see* type
- recursion, 6, 138
  - list, 74
- recursive data type, 128
- recursive function, 6, 8
- recursive type, *see* type
- ref-, 182
- reference, 182
- Regex, 224
  - Length, 224
  - Match, 224
  - Replace, 224
  - Success, 224
- regular expression, 221, 222
  - anchor, 223
  - capture, 224
  - match, 224
  - Regex, 224
- remainder, *see* division
- remove, 106
- reserved word, 2
- resource measurement, 203
- resource measures, 197
- return!, *see* computation expression
- ReturnFrom, *see* computation expression
- rev, 79
- reverse, 79
- save in file, 230
- scope, 55
- search tree, 133, *see* tree
- selection, *see* database
- selector, 51
- Seq, *see* Collection Library
- seq, *see* type constructor
- Seq.iter, 184
- Seq.iteri, 184
- sequence, 251
  - append, 254
  - cache, 254
  - caches, 253
  - collect, 254
  - combine, 262
  - delay, 256
  - filter, 254
  - init, 254
  - initInfinite, 254
  - map, 254
  - recursion, 256
  - side effect, 256
  - specialization, 266
- sequence expression, 262
  - for, 263
  - if-then, 263
  - if-then-else, 263
  - let!, 263
  - yield, 262
  - yield!, 262
- sequence expressions, 282
- sequential composition, 179
- Set, *see* Collection Library
- set, 104
  - add, 106
  - contains, 106
  - count, 106
  - difference, 105
  - difference, 108
  - empty, 104
  - exists, 108
  - filter, 108
  - fold, 108
  - foldBack, 108
  - forall, 108
  - intersect, 108
  - intersection, 105
  - isSubset, 106
  - map, 108
  - maxElement, 106
  - minElement, 106
  - ofList, 106
  - remove, 106
  - subset, 104
  - toList, 106
  - union, 105
  - union, 108
- set comprehension, 104
- Seq.iter, 184
- set imperative, 188
- shared mutable data, 314
- side-effect, 176
- Sieve of Eratosthenes, 258, 263
- signature file, *see* module
- Some, 64

- special character, 24
- sprintf, 235
- SQL database, 267
- stack, 198
  - limit, 204
- static binding, 34
- store, 175
- string, 26
  - conversion, 27
  - function, 26
  - index, 27
  - verbatim, 26
- string, *see* type constructor
- string matching, 224
- String.Format, 237
- string I/O, 229
- String ordering, 232
- symbolic differentiation, 127
- synchronous operation, 313
- table
  - domain, 113
- tagged value, 58
- tagged values
  - equality, 59
  - ordering, 59
- tail, 68
- tail recursion, 206
- tail recursive function, *see* iterative function
- tail-recursive function, 209, 212
- task, 313
  - work item, 313
- TextProcessing, 224, 230, 233, 346
  - captureCount, 224, 346
  - captureCountList, 224, 346
  - captureList, 224, 346
  - captureSingle, 224, 346
  - fileFold, 230, 346
  - fileIter, 230, 346
  - fileXfold, 230, 346
  - fileXiter, 230, 346
  - orderCulture, 233, 346
  - orderString, 233, 346
  - restoreValue, 230, 346
  - saveValue, 230, 346
  - string, 233, 346
- text I/O, 229
- thread, 312
  - create, 314
  - safe collections, 315
  - start, 314
- thread pool, 313
- time, 235, 236
- #time, 203
- token, 293
- toList, 106, 114
- tree
  - search tree, 134
  - variable number of sub-trees, 138, 140
- tree iteration, 185
- tree traversal, 133, 139, 185
  - breadth first, 139
  - depth first, 139
  - imperative, 185
  - in-order, 133
  - list trees, 139
  - post-order, 133
  - pre-order, 133
- triple, *see* tuple
- true, 22, 62
- truth value, 22
- try-finally, *see* computation expression
- TryFinally, *see* computation expression
- tryFind, 95, 114
- tuple, 43
  - equality, 45
  - expression, 44
  - n*-tuple, 43
  - ordering, 46
  - pair, 11, 43
  - pattern, 46
  - quadruple, 43
  - triple, 43
  - type, 44
- tuple equality, *see* tuple
- tuple ordering, *see* tuple
- type, 1, 2
  - augmentation, 153
  - check, 13
  - declaration, 50
  - equality, 36
  - equality type, 79
  - equality type variable, 79
  - function, 3
  - inference, 13, 31
  - instance, 84
  - monomorphic, 82
  - parameterized, 131
  - polymorphic, 48
  - record, 50
  - recursive, 121, 123
  - tree, 123
  - variable, 48
- type, 131
- type constructor
  - bool, 2, 22
  - char, 24
  - float, 3, 21
  - int, 2, 21
  - list, 68
  - seq, 251
  - string, 26
  - unit, 23
- type declaration
  - tagged values, 58
- type extension, 155

- type provider, 267
  - SQL, 268
- type variable, 36
  - comparison, 37
  - equality, 36
- type variable, *see* type
- union, 108
- unit, *see* type constructor
- URI, 339
- use, *see* computation expression
- use-binding, 232
- use keyword, 232
- Using, *see* computation expression
- value, 1
- value polymorphism, *see* polymorphism
- value restriction, 81, 195
- variable, 175
- Venn diagram, 105
- verbatim string, *see* string
- web page format, 339
- when, 37
- While, *see* computation expression
- while, 183
- while loop, 183
- wildcard, *see* pattern
- window, 165
- work item, *see* task
- workflows, 279
- Yield, *see* computation expression
- yield, *see* computation expression
- yield!, *see* computation expression
- YieldFrom, *see* computation expression
- Zero, *see* computation expression