

# Index

*k*-ary, 147  
(\*\*), 87  
->, 55  
., 75  
//, 87  
:, 49, 50  
::, 99  
:>, 249  
:?, 250  
;, 49  
\_, 50  
Item, 232  
System.Console.ReadKey, 168  
System.Console.ReadLine, 168  
System.Console.Read, 168  
System.Console.WriteLine, 168  
System.Console.Write, 168  
System.ConsoleKeyInfo.KeyChar, 169  
System.ConsoleKeyInfo.Key, 169  
System.ConsoleKeyInfo.Modifiers, 169  
System.IO.Directory.CreateDirectory,  
183  
System.IO.Directory.Delete, 183  
System.IO.Directory.Exists, 183  
System.IO.Directory.GetCurrentDirectory,  
183  
System.IO.Directory.GetDirectories,  
183  
System.IO.Directory.GetFiles, 183  
System.IO.Directory.Move, 183  
System.IO.Directory.SetCurrentDirectory,  
183  
System.IO.File.Copy, 182  
System.IO.File.CreateText, 179  
System.IO.File.Create, 179  
System.IO.File.Delete, 182  
System.IO.File.Exists, 182  
System.IO.File.Move, 182  
System.IO.File.OpenRead, 179  
System.IO.File.OpenText, 179  
System.IO.File.OpenWrite, 179  
System.IO.File.Open, 179  
System.IO.FileMode.Append, 179  
System.IO.FileMode.CreateNew, 179  
System.IO.FileMode.Create, 179  
System.IO.FileMode.OpenOrCreate, 179  
System.IO.FileMode.Open, 179  
System.IO.FileMode.Truncate, 179  
System.IO.FileStream.CanRead, 180  
System.IO.FileStream.CanSeek, 180  
System.IO.FileStream.CanWrite, 180  
System.IO.FileStream.Close, 180  
System.IO.FileStream.Flush, 180  
System.IO.FileStream.Length, 180  
System.IO.FileStream.Name, 180  
System.IO.FileStream.Position, 180  
System.IO.FileStream.ReadByte, 180  
System.IO.FileStream.Read, 180  
System.IO.FileStream.Seek, 180  
System.IO.FileStream.WriteByte, 180  
System.IO.FileStream.Write, 180  
System.IO.Path.Combine, 184  
System.IO.Path.GetDirectoryName, 184  
System.IO.Path.GetExtension, 184  
System.IO.Path.GetFileNameWithoutExtension,  
184  
System.IO.Path.GetFileName, 184  
System.IO.Path.GetFullPath, 184  
System.IO.Path.GetTempFileName, 184  
System.IO.StreamReader.Close, 182  
System.IO.StreamReader.EndOfStream,  
182  
System.IO.StreamReader.Flush, 182  
System.IO.StreamReader.Peek, 182

- System.IO.StreamReader.ReadLine, 182
- System.IO.StreamReader.ReadToEnd, 182
- System.IO.StreamReader.Read, 182
- System.IO.StreamWriter.AutoFlush, 182
- System.IO.StreamWriter.Close, 182
- System.IO.StreamWriter.Flush, 182
- System.IO.StreamWriter.WriteLine, 182
- System.IO.StreamWriter.Write, 182
- abs, 31
- acos, 31
- asin, 31
- atan2, 31
- atan, 31
- ceil, 31
- cosh, 31
- cos, 31
- exp, 31
- floor, 31
- log10, 31
- log, 31
- max, 31
- min, 31
- pown, 31
- round, 31
- sign, 31
- sinh, 31
- sin, 31
- sqrt, 31
- stderr, 168
- stdin, 168
- stdout, 168
- swap, 263
- tanh, 31
- tan, 31
- \_, 78
- (), 11, 13
- <-, 190
- <<, 140
- >>, 140
- [], 40, 97, 199
- abstract class, 252
- [abstract member](#), 252
- [<AbstractClass>], 252
- accessors, 232
- aggregation, 274
- American Standard Code for Information Interchange, 305
- ancestor, 147
- and, 34
- [and](#), 111, 117
- anonymous functions, 55, 138
- anonymous variable type, 78
- ArgumentException, 170
- Array.append, 202
- Array.copy, 203
- Array.ofList, 203
- Array.toList, 203
- Array2D, 204
- Array2D.copy, 206
- Array2D.create, 206
- Array2D.init, 206
- Array2D.iter, 207
- Array2D.length1, 207
- Array2D.length2, 207
- Array2D.map, 207
- Array3D, 204
- Array4D, 204
- arrays, 198
- [as](#), 172
- ASCII, 305
- ASCIIbetical order, 39, 306
- assembly, 312
- assignment, 190
- association, 274
- asymptotic notation, 99
- base, 25, 299
- [base](#), 249
- base class, 248
- bash, 294
- Basic Latin block, 307
- Basic Multilingual plane, 307
- basic types, 25
- Big-O, 99
- binary number, 26, 299
- binary operator, 31
- binary tree, 147
- binary64, 301
- bit, 26, 299
- black-box testing, 217
- bool, 25
- branch, 17, 60, 62
- branching coverage, 220
- bug, 216
- byte, 299
- byte[], 27
- byte, 27
- call stack, 116
- call-back function, 4
- cast, 17
- casting exceptions, 171
- catching exception, 171
- cd, 291, 295
- char, 25, 27
- character, 27
- child, 147

- CIL, 311
- class, 29, 40, 228
- class diagram, 273
- CLI, 311
- Clone, 202
- close file, 166
- closure, 58, 138
- CLR, 311
- code block, 51
- code point, 27, 307
- Command Line Interface, 311
- comments, 21
- Common Intermediate Language, 311
- Common Language Infrastructure, 311
- Common Language Runtime, 311
- Common Type System, 311
- compile mode, 11, 125
- compile-time, 93
- compiles, 126
- composition, 274
- composition operator, 140
- computational complexity, 99
- condition, 194
- console, 289
- constructor, 229
- copy constructor, 235
- coverage, 220
- create file, 166
- CTS, 311
- currying, 141
  
- debugging, 12, 217
- decimal, 27
- decimal number, 25, 299
- decimal point, 25, 299
- declarative, 92
- declarative programming, 3
- `default`, 252
- `del`, 293
- delete file, 166
- derived class, 248
- descendant, 147
- digit, 25, 299
- `dir`, 291
- directory, 289
- discriminated unions, 243
- Dispose, 185
- DivideByZeroException, 170
- `do`, 59, 194, 195
- do-binding, 11, 59
- dot notation, 40
- double, 301
- double, 27
- downcast, 30, 249
  
- dynamic scope, 51, 93
- dynamic type pattern, 171
  
- echo, 292, 296
- efficiency, 216
- `elif`, 61
- `else`, 61
- empty set, 154
- encapsulate, 15
- encapsulation, 52, 57
- EntryPoint, 167
- environment, 63
- error message, 14
- escape sequences, 27
- event-driven programming, 4
- exception, 37
- exclusive or, 37
- executable file, 93
- exit status, 167
- `exn`, 25, 170
- expression, 3, 10, 30, 49
- Extensible Markup Language, 87
  
- `failwith`, 175
- field, 229
- file, 166
- first-class citizenship, 58, 93
- float, 25
- `float32`, 27
- floating point number, 25
- flushing, 180
- fold, 93
- foldback, 93
- folder, 289
- `for`, 194
- `for-downto`, 195
- `for-to`, 194
- formatting string, 11
- fractional part, 25, 30
- `fst`, 71
- `fun`, 55
- function, 3, 11
- function body, 53
- function coverage, 220
- functional programming, 3, 162
- functional programming paradigm, 92
- functionality, 216
- functions, 229
  
- generic function, 54
- graphical user interface, 289
- ground, 97
- guard, 60
- GUI, 289

- handling exception, 171
- has-a relation, 274
- hash maps, 143
- Head, 98
- Tail, 99
- head, 97
- hexadecimal number, 26, 300
- higher-order function, 93, 137
- how, 163, 272
- identifier, 46
- IEEE 754 double precision floating-point format, 301
- [if](#), 61
- immutable state, 93
- imperative, 92
- imperative programming, 4, 94, 162
- imperative programming paradigm, 92, 162
- implementation file, 13
- implementation files, 126, 128
- indentation, 17
- `IndexOutOfRangeException`, 170
- infinite sets, 154
- infix, 129
- infix notation, 31
- infix traversal, 147
- inheritance, 248, 278
- injective, 157
- [inline](#), 79
- input pattern, 59
- instantiate, 228
- `int`, 25
- `int16`, 27
- `int32`, 27
- `int64`, 27
- `int8`, 27
- integer, 25
- integer division, 36
- integer remainder, 36
- interactive mode, 11
- interface, 229, 254
- [interface with](#), 254
- interprets, 126
- `invalidArg`, 175
- is-a relation, 248, 278
- `IsEmpty`, 98
- `it`, 13, 25
- iter, 93
- jagged arrays, 203
- just-in-time, 311
- key-value pairs, 157
- keyword, 10, 48
- knows-about relation, 275
- Landau symbol, 116
- Latin-1 Supplement block, 307
- Latin1, 306
- [lazy](#), 93
- lazy evaluation, 93
- leaf, 147
- least significant bit, 299
- Length, 40, 98, 202
- length, 70
- [let](#), 50, 117
- let-binding, 10
- lexeme, 10
- lexical scope, 51, 55
- lexicographical scope, 93
- library file, 13
- lightweight syntax, 49
- list, 97
- list concatenation, 99
- list cons, 99
- list module, 101
- `List.concat`, 105
- `List.exists`, 102
- `List.filter`, 102
- `List.fold`, 103
- `List.foldBack`, 103
- `List.forall`, 103
- `List.init`, 103
- `List.isEmpty`, 105
- `List.iter`, 104
- `List.length`, 105
- `List.map`, 104
- `List.rev`, 101
- `List.sort`, 101
- `List.tryFind`, 104
- `List.tryFindIndex`, 104
- `List.tryHead`, 105
- `List.tryItem`, 105
- `List.tryLast`, 101
- `List.unzip`, 102
- `List.zip`, 102
- literal, 25
- literal type, 27
- lower camel case, 48
- `ls`, 294
- machine code, 162
- maintainability, 217
- manifest, 312
- map, 93, 156
- member, 29, 70, 228
- Metadata, 311
- method, 40, 228, 229

- mixed case, 48
- `mkdir`, 292, 295
- models, 228
- module, 126, 311
- `module`, 126
- most significant bit, 299
- move, 292
- multidimensional arrays, 203
- mutable, 19
- `mutable`, 190
- mutable value, 93, 162, 190
- mutable values, 189
- mutually recursive, 117
- `mv`, 296
  
- namespace, 29
- `NaN`, 301
- nested scope, 51
- `new`, 230, 240
- newline, 27
- node, 146
- not, 34
- not-a-number, 301
- `NotFiniteNumberException`, 170
- nouns, 273
- nouns and verbs, 271
- nouns-and-verbs method, 273
  
- obfuscation, 72
- `obj`, 25
- object, 4, 40, 228
- object-oriented analysis, 228
- object-oriented analysis and design, 272
- object-oriented design, 228
- Object-oriented programming, 163
- object-oriented programming, 4, 162, 228
- object-oriented programming paradigm, 92, 162
- octal number, 26, 300
- open file, 166
- operand, 30, 54
- operator, 10, 30, 33, 53
- operator overloading, 238
- option type, 74
- or, 34
- out-of-bounds exception, 97
- overflow, 35
- `OverflowException`, 170
- overloading, 238
- override, 248, 251
- `override`, 252
- overshadow, 249
  
- package, 280
  
- parent, 147
- partial specification, 141
- pascal case, 48
- pipng, 57
- portability, 217
- postfix, 129
- postfix traversal, 147
- precedence, 32, 33
- prefix operator, 32
- prefix traversal, 147
- primary constructor, 240
- primitive types, 70
- `printfn`, 11
- private, 231
- problem statement, 272, 280
- procedure, 57, 162
- programming paradigm, 92
- properties, 228, 229
- property, 40, 98
- public, 231
- pure function, 92
  
- queues, 143
  
- ragged multidimensional list, 100
- `raise`, 173
- raising exception, 171
- range expressions, 97, 199
- read file, 166
- reals, 300
- `rec`, 19
- `rec`, 111, 117
- recursion, 31, 93
- recursion step, 113
- recursive function, 111
- redirection, 292, 296
- reduce, 93
- reference types, 200
- referential transparency, 92
- relation, 146
- reliability, 216
- `rm`, 296
- `rmdir`, 293, 296
- root, 147
- rounding, 30
- runtime, 17
- runtime error, 37
- runtime resolved variable type, 78
  
- `sbyte`, 27
- scientific notation, 26
- scope, 50
- script file, 13
- search path, 293, 296

- self identifier, 229, 230
- seq, 93
- sequence expression, 97, 198
- set, 154
- sets, 143
- siblings, 147
- side-effect, 57, 162, 199
- signature file, 13, 128
- signature files, 126
- single, 27
- singleton set, 154
- slicing, 199
- snd, 71
- software testing, 217
- source code, 13
- stack, 130
- stack frame, 116
- state, 4, 92, 162
- statement, 4, 11, 59, 162
- statement coverage, 220
- statically resolved variable type, 78
- stdin, 16
- stopping condition, 113
- stopping step, 113
- stream, 16, 166
- string, 16, 27
- string, 25
- strongly typed, 93
- struct records, 75
- structured programming, 4
- subnormals, 301
- surjective, 156
- System.IDisposable, 185
- System.Object, 251, 253
- tail, 97
- tail-recursion, 93, 117
- terminal, 289
- The Heap, 73, 75, 228, 235
- The Stack, 116
- then, 61
- toString, 147
- trees, 143
- truth table, 34
- tuple, 70
- type, 12, 25
- type abbreviation, 77
- type aliasing, 77
- type annotations, 93
- type constraints, 80
- type declaration, 13
- type inference, 12, 13, 93
- type safety, 54
- typecasting, 29
- uint16, 27
- uint32, 27
- uint64, 27
- uint8, 27
- UML, 271, 273
- underflow, 35
- unfolding loops, 208
- Unicode, 27
- unicode block, 307
- Unicode general category, 307
- Unicode Standard, 307
- Unified Modelling Language 2, 273
- unit, 25
- unit testing, 217
- Universal Modelling Language, 271
- upcast, 30, 249
- upper camel case, 48
- usability, 216
- use, 185
- use case, 272
- user story, 272
- using, 186
- UTF-16, 308
- UTF-8, 308
- value-binding, 49
- variable, 19, 93, 189, 190
- variable types, 78
- verbatim, 29
- verbose syntax, 49
- verbs, 273, 283
- VES, 311
- Virtual Execution System, 311
- what, 163, 272
- when, 80
- while, 19
- while, 194
- white-box testing, 217, 220
- whitespace, 27
- whole part, 25, 30
- wildcard, 17, 60
- wildcard pattern, 50
- Windows command line, 289
- with, 76
- word, 299
- write file, 166
- XML-standard, 87
- xor, 37