

# References

- Boo05. George Boole. *An Investigation of the Laws of Thought*. Gutenberg, <https://www.gutenberg.org/files/15114/15114-pdf.pdf>, Rel. 2005.
- Dur88. Peter Duren, editor. *A Century of Mathematics in America*, volume 1. American Mathematical Society, 1988.
- Dur89a. Peter Duren, editor. *A Century of Mathematics in America*, volume 2. American Mathematical Society, 1989.
- Dur89b. Peter Duren, editor. *A Century of Mathematics in America*, volume 3. American Mathematical Society, 1989.
- HBS73. Carl Hewitt, Peter Bishop, and Richard Steiger. A universal modular actor formalism for artificial intelligence. In *International Joint Conference on Artificial Intelligence*, pages 235–245, 1973.
- Hod05. Luke Hodgkin. *A History of Mathematics: From Mesopotamia to Modernity*. Oxford University Press, Oxford, 2005.
- Kle56. S. C. Kleene. Representation of events in nerve nets and finite automata. *Automata Studies*, pages 3–41, 1956.
- OLC17. One laptop per child. <http://one.laptop.org/>, 2017.
- Ray65. John C. Reynolds. *COGENT Programming Manual*. Argonne National Laboratory, Argonne, IL, 1965.
- UNL13. United nations adult literacy rate. <http://data.un.org/Data.aspx?d=SOWC&f=inID%3A74>, 2013.

# Index

- abstract, 92
- abstract class, 80
- abstract method, 90
- actor
  - Actor trait*, 151
  - ActorSystem*, 152
  - hierarchy, 157
  - parent, 159
  - reference, 153
- actor model, 151
- actors, 151
- Akka, 151, 152, 157
- algorithm, 101
- algorithms, 102
- anonymous function, 101
- architecture, 7
- argument list, 104
- Array, 141
- array, 101
- ArrayBuffer, 141, 170
- assembly language, 6
- associative
  - right, 126
- associativity, 57
- automatically, 111
  
- batch mode, 5
- bidirectional, 171
- big data, 68, 69
- binding, 104
- bitwise operators, 55
- blocks, 90
- buffer, 66
- building blocks, 33
  
- calculation, 57
- calculator, 4
  
- child elements, 170
- child nodes, 168
- chunks, 67
- class
  - case, 111
  - definition, 33
  - field, 31
  - member, 31
  - method, 31
  - regular, 111
  - sealed, 112
- closure, 104
- collections
  - immutable, 137
  - mutable, 137
- communication, 153
- compact, 137
- compiler, 111
- composition, 82, 87
- composition
  - sequential, 181
- computation, 1, 151
- computational
  - capacity, 6
  - goal, 3
  - goals, 111
- computer, 99
- computer programming, 99
- computing, 1
- computing
  - machine, 3
- concrete, 92
- constructor, 78, 88, 92, 116
- constructor
  - parameters, 88
- constructor argument, 78
- constructor parameters, 33, 79

- constructors, 88
- control structures, 99, 106
- correctness, 13
- cryptography, 55
- currying, 104
- custom operator, 51
  
- data structure, 131, 140
- declaration
  - val*, 24
  - var*, 25
  - type, 26
- deep matching, 116
- dependency, 165
- deserialization, 171
- diamond inheritance, 91
- dimension, 133
- directory, 70
- directory listing, 71
- distributed, 104, 137
- do-while, 42
- driver program, 157
- DSL, 106
- dynamic binding, 81
  
- efficiency, 13
- encode, 99
- Erlang, 151
- error messages, 184
- exception, 155
- exception
  - NumberFormatException*, 44
  - RuntimeException*, 44
- exception handling, 43
- expression, 26
- extends, 77
- extractor, 118
  
- fault-tolerant, 159
- file listing, 70
- files, 90
- filtered, 101
- flag, 38
- flat file, 68
- for expression, 39
- framework, 151
- frameworks, 105
- free variables, 104
- function
  - as a parameter, 103
  - curried, 105
  - definition, 104
  - higher order, 103, 106
- function literal, 102
- functional programming, 9
- functional programming language, 99
- functional programming paradigm, 99
- functions, 99
  
- generic, 12
- grammar, 180
  
- has-a, 77, 82
- head, 129
- hierarchy, 87, 151
- history, 99, 151
  
- IDE, 53
- identifiers, 23
- if-else, 37
- immutable, 138, 139, 143
- import, 70
- in-memory, 174
- indexed, 140
- inherit, 77
- integer, 101
- interfaces, 87
- interpreter, 101, 129
- is-a, 77
  
- JSON, 180, 182
- JSON
  - data, 182
  - structure, 182
- JVM, 9
  
- Kleene, 63
  
- level
  - first, 116
  - second, 116
  - third, 116
- life cycle, 155
- linearization, 89
- Linux, 5
- List, 140
- list
  - class, 130
  - head, 126
  - operations, 126
  - tail, 126
- literals, 17
- literals
  - boolean, 20
  - characters, 17
  - comments, 21
  - integers, 19
  - lexical elements, 21

- newline characters, 22
  - strings, 18
  - symbol, 21
  - whitespace, 21
- loading, 173
- LOC, 8, 32
- logical methods, 54
- logical-and, 54
- logical-or, 54
- Mac OS, 5
- machine, 99
- main method, 33
- maintainability, 12
- map method, 102
- mathematical structures, 137
- mathematics, 99, 111
- member methods, 100
- memory, 3, 173
- message passing, 153
- messages, 151
- messages
  - immutable, 151
- meta-language, 52
- methods, 79, 90
- microprocessor, 7
- mix-in, 89
- model, 32, 33, 91
- modifier, 80
- modularity, 13
- module, 151
- multiple conditions, 38
- multiple inheritance, 78, 91
- mutable, 139, 141
- natural language, 137
- Node, 166
- object
  - companion, 132
  - definition, 33
  - member, 100
  - singleton, 81, 100, 118
- OOP, 9
- operating system, 5
- operator
  - cons*, 129
  - arithmetic, 52
  - concatenation, 130
  - custom, 51
  - extension, 127
  - unary, 51
- output device, 4
- overriding, 79, 92
- paradigms, 99
- parameter
  - step, 133
- parser, 179
- parser
  - building blocks, 180
  - combinators, 180
  - domain specific, 184
- pattern
  - variable, 113
  - wildcard, 113
- pattern matching, 111
- PCDATA, 166
- pipe, 106
- polymorphism, 81
- precedence, 56
- predicate, 132
- prefix, 157
- PrintWriter, 70
- process
  - asynchronous, 153
- program, 4
- program
  - driver, 153
- program attributes, 12
- program crashing, 43
- programming
  - multi-threaded, 152
- programming language
  - C, 7
  - high level, 6
  - Java, 8
  - low level, 6
  - Pascal, 7
  - Scala, 8
- programming languages
  - high level, 137
- propagated, 44
- Queue, 142
- re-usable, 32
- readability, 12
- regular expression, 64
- relational, 54
- REPL, 53
- reserved words, 23
- reserved words
  - =>, 23
  - extends*, 77
  - <-, 23
- restrict, 112
- rigorous, 99
- runnable application, 33

- save, [174](#)
- SBT, [180](#)
- Scala API, [184](#)
- Scala interpreter, [53](#)
- security, [55](#)
- sequence, [140](#)
- sequences, [119](#)
- serialization, [171](#)
- set, [138](#)
- shift operators, [56](#)
- simplicity, [13](#)
- space, [66](#)
- Stack, [142](#)
- storage, [4](#)
- string indexing, [51](#)
- sub classes, [112](#)
- super class, [80](#)
- super class
  - constructor, [80](#)
- supervisor, [159](#)
  
- tail, [129](#)
- target system, [171](#)
- term
  - closed, [104](#)
  - open, [104](#)
- text, [166](#)
- text file, [68](#)
- thinner traits, [92](#)
- thought, [99](#)
  
- threads, [152](#)
- tokens, [179](#)
- trait, [87](#)
- traits, [91](#)
- transmission, [66](#), [67](#)
- truth table, [54](#)
- try-catch, [70](#)
- try-catch-finally, [43](#)
- tuple, [117](#), [143](#)
- type
  - declaration, [24](#)
  - type inference, [51](#)
  - types, [23](#), [114](#)
  
- underscore, [103](#)
  
- variations, [39](#)
- vector, [102](#)
  
- while loop, [42](#)
- whole-part, [82](#)
- wild card, [117](#)
- Windows, [6](#)
  
- XML, [120](#), [165](#)
- XML
  - element, [165](#)
  - module, [165](#)
  - pattern, [170](#)
- XPath, [170](#)