Raku Explained

Arne Sommer

Version 0.01, 22.12.2019

Table of Contents

Introduction 1

The Little Print 1

Reading Tips 2

Content 3

1. About Raku 5

1.1. Rakudo 5

1.2. Running Raku in the browser 6

1.3. REPL 6

1.4. One Liners 8

1.5. Running Programs 9

1.6. Error messages 9

1.7. use v6 10

1.8. Documentation 10

1.9. More Information 13

1.10. Speed 13

2. Variables, Operators, Values and Procedures. 15

2.1. Output with say and print 15

2.2. Variables 15

2.3. Comments 18

2.4. Non-destructive operators 18

2.5. Numerical Operators 19

2.6. Operator Precedence 21

2.7. Values 23

2.8. Variable Names 25

2.9. constant 26

2.10. Sigilless variables 27

2.11. True and False 27

2.12. // 30

3. The Type System 33

3.1. Strong Typing 33

3.2. ^mro (Method Resolution Order) 35

3.3. Everything is an Object 36

3.4. Special Values 38

3.5. :D (Defined Adverb) 40

3.6. Type Conversion 41

3.7. Comparison Operators 44

4. Control Flow 49

4.1. Blocks 49

4.2. Ranges (A Short Introduction) . . . 49
4.3. loop . . . 50
4.4. for . . . 51
4.5. Infinite Loops . . . 55
4.6. while . . . 55
4.7. until . . . 56
4.8. repeat while . . . 57
4.9. repeat until . . . 57
4.10. Loop Summary . . . 58
4.11. if . . . 59
4.12. given . . . 60
4.13. with . . . 60
4.14. ?? !! . . . 63
4.15. do . . . 64
4.16. when . . . 64
4.17. Loop Manipulation . . . 65
5. Numbers . . . 69
5.1. Octal, Hex, Binary … . . . 69
5.2. Unicode Numbers . . . 70
5.3. Not a Number . . . 71
5.4. N_U_M_B_E_R_S . . . 71
5.5. Floating Point Numbers . . . 71
5.6. Rational Numbers . . . 72
5.7. narrow . . . 73
5.8. sign . . . 73
5.9. Rounding . . . 74
5.10. NaN (Not a Number) . . . 75
5.11. =~= . . . 76
5.12. is-prime (Prime Numbers) . . . 76
5.13. Modulo and variants . . . 77
5.14. Other Operators . . . 78
6. Basic Input and Output . . . 83
6.1. Newlines . . . 83
6.2. Stringification . . . 84
6.3. Output . . . 86
6.4. printf . . . 88
6.5. Input from the user . . . 93
7. Strings . . . 97
7.1. Unicode . . . 97
7.2. join . . . 101
7.3. split . . . 101

7.4. words 101
7.5. comb 102
7.6. flip 103
7.7. substr (Partial Strings) 103
7.8. Changing Case 105
7.9. x (String Repetition Operator) 107
7.10. succ 108
7.11. pred 108
7.12. Quoting 109
7.13. Multi-line Strings (Heredocs) 110
8. Arrays and Lists 113
8.1. , (List Operator) 113
8.2. [ ] (Array Constructor) 114
8.3. <xxx> (Quote Word) 114
8.4. Empty 114
8.5. List Elements 114
8.6. pop / push / shift / unshift 115
8.7. rotate (List Rotation) 118
8.8. List of Lists 118
8.9. Flattening Lists 118
8.10. splice 119
8.11. map 121
8.12. sort 123
8.13. reverse 124
8.14. Array with Limits 125
8.15. Typed Array 125
8.16. Shaped Array 125
8.17. unique (Lists Without Duplicates) 126
8.18. xx (List Repetition Operator) 127
8.19. List Selection 128
8.20. min / max 130
8.21. Random Values 130
8.22. permutations 135
8.23. combinations 135
9. Pair and Hashes 137
9.1. Pair 137
9.2. Hash 138
9.3. Hash Constructor { } 139
9.4. Hash Assignment and Values 140
9.5. keys 140
9.6. values 141

9.7. kv (keys + values) 141
9.8. Typed Hash 142
9.9. Shaped Hash 143
9.10. invert 144
9.11. antipairs 145
9.12. Hash Slices 145
9.13. Hash Lookup 146
9.14. Hash Deletion 146
9.15. Hash Duplicate Values 146
9.16. Hash Usage 147
9.17. Grep and Smartmatch 148
9.18. Hash (method) 148
10. Procedures 151
10.1. Procedures Without Arguments 151
10.2. Procedures With Arguments 151
10.3. @_ 152
10.4. Placeholder Variables 152
10.5. Procedures as variables 153
10.6. Type Constraints 154
10.7. return 155
10.8. @*ARGS 157
10.9. MAIN 158
10.10. WHY 159
10.11. IntStr Gotcha 162
10.12. Multiple Dispatch 162
10.13. Procedure Arguments 165
10.14. * (Slurpy Operator) 171
10.15. Blocks Revisited 175
10.16. Calling a procedure specified in a variable 176
10.17. Procedures in Procedures 177
11. Regex Intro 179
11.1. What is a Regex? 179
11.2. Making a Regex 180
11.3. ~~ (Smartmatch Operator) 181
11.4. Partial Strings 181
11.5. Beginning or end of a string 183
11.6. Regex Metacharacters 184
11.7. $/ (Match Object) 185
11.8. Special Characters 185
11.9. Capturing and Grouping 186
11.10. Character Classes 188

11.11. Custom Character Classes . . . 191
11.12. Non-greedy . . . 191
11.13. Backwards References . . . 192
11.14. Using a Regex . . . 193
11.15. String Substitution . . . 193
11.16. Substitution Tuning . . . 197
11.17. Transliteration . . . 198
11.18. trim / trim-leading / trim-trailing . . . 199
11.19. split and grep . . . 200
11.20. Comments . . . 200
12. Modules . . . 203
12.1. Precompilation . . . 203
12.2. Module Administration with zef . . . 203
12.3. Using Modules (use) . . . 207
12.4. Writing Modules . . . 209
13. Files and Directories . . . 211
13.1. Reading Files . . . 211
13.2. slurp . . . 213
13.3. open / close . . . 214
13.4. INPUT OUTPUT - IO . . . 214
13.5. Writing Files . . . 215
13.6. get . . . 217
13.7. Temporary Files . . . 219
13.8. File tests . . . 222
13.9. Binary Files . . . 223
13.10. Directories . . . 229
14. Date and Time . . . 237
14.1. time . . . 237
14.2. now . . . 237
14.3. Leap Seconds . . . 237
14.4. Instant . . . 238
14.5. Timing . . . 244
15. Writing a Module . . . 247
15.1. unit module . . . 247
15.2. is export . . . 247
15.3. pm6 . . . 247
15.4. use lib . . . 248
15.5. Timing Fibonacci . . . 249
15.6. Dictionaries . . . 251
16. Ranges and Sequences . . . 255
16.1. Ranges . . . 255

16.2. lazy 256
16.3. Sequences 259
16.4. state 263
16.5. Truly Random Flip-Flop 264
16.6. Flip-Flop Problems 264
16.7. gather / take 266
16.8. Closures 272
17. Classes 275
17.1. has 275
17.2. new 275
17.3. method 276
17.4. Named Arguments 279
17.5. Public Class Variables 279
17.6. self 279
17.7. Custom «new» 279
17.8. Custom BUILD 280
17.9. Wrong Start Value 281
17.10. Object Comparison 282
17.11. A Person Class 284
17.12. Output 289
17.13. Private Methods 292
17.14. Inheritance 294
17.15. Roles 297
17.16. Multiple Dispatch 299
17.17. A Fallback Method 300
17.18. .? 303
17.19. .+ 303
17.20. .* 304
17.21. handles (Delegation) 305
17.22. Calling a method specified in a variable 306
18. Advanced Regexes 307
18.1. Named Regexes 307
18.2. IPv4 Validation 307
18.3. Regex boolean condition check 308
18.4. Named Capturing 309
18.5. Subrules 310
18.6. || (Alternation) 311
18.7. && (Conjunction) 312
18.8. Anchors 313
18.9. Zero-width assertions 314
18.10. Backwards References Revisited 315

18.11. match 316
18.12.)> (End of Capture) 317
18.13. <) (Start of Capture) 317
18.14. More about Regexes 317
19. Warnings, Failures and Exceptions 319
19.1. Warnings 319
19.2. Failures 320
19.3. Exceptions 321
19.4. Custom Exceptions 325
19.5. Soft Errors 326
19.6. Signals 327
20. Writing Modules 329
20.1. Export Tags 331
20.2. Testing Code 332
20.3. Possible Changes 342
20.4. Other Files 342
20.5. The api tag 344
20.6. Writing Documentation 344
20.7. Using Github 344
20.8. use 344
20.9. require 344
20.10. need 345
20.11. import 345
20.12. CompUnit::Repository 345
21. Running External Programs 347
21.1. run 347
21.2. shell 348
21.3. Parrot Shell 348
21.4. given / when 349
21.5. Getting rid of one level of braces 350
21.6. Command History 351
21.7. Command Line Completion 352
21.8. Executing Programs 353
21.9. Changing Directories 356
21.10. Without «run» 357
21.11. Configuration 358
21.12. Catching Interrupts 359
21.13. A Better Command List 359
21.14. An Even Better Command List 361
21.15. Dynamic Shell 363
21.16. Aliases 363

21.17. Proc . . . 366
21.18. qx/.../ (Quote eXecute) . . . 367
22. Asyncronous Execution . . . 369
22.1. Proc::Async and Promises . . . 369
22.2. Promise . . . 383
22.3. File Events (IO::Notification) . . . 383
22.4. Parallel Execution . . . 384
22.5. channel . . . 387
22.6. Supply . . . 388
22.7. Atomic Int / Operators . . . 389
23. Extending Perl 6 . . . 391
23.1. subset of . . . 391
23.2. Adding Operators . . . 393
23.3. Extending Built in Classes . . . 398
23.4. Procedures, Methods and Operators . . . 400
23.5. term:<> . . . 401
24. Hashes and Relatives . . . 403
24.1. Mutable and Immutable . . . 403
24.2. Map . . . 405
24.3. Set . . . 406
24.4. SetHash . . . 408
24.5. Bag . . . 409
24.6. BagHash . . . 411
24.7. Next Word Frequency . . . 412
24.8. A Gibberish Generator . . . 415
24.9. Mix . . . 423
24.10. MixHash . . . 423
24.11. Coercers . . . 423
25. Other Builtin Types . . . 425
25.1. Enum . . . 425
25.2. Junction . . . 427
25.3. Set vs Junction . . . 429
26. Introspection (and Reflection) . . . 433
26.1. Introspection Methods . . . 434
26.2. Introspection Variables . . . 434
26.3. ^methods . . . 438
26.4. Procedure Introspection . . . 442
26.5. Object Introspection . . . 443
26.6. WHERE . . . 445
26.7. The Debugger . . . 445
26.8. HOW . . . 445

27. Web Programming . . . 447
27.1. A File Downloader . . . 447
27.2. Web Frameworks . . . 456
27.3. Cro::WebApp . . . 462
27.4. A Web Publication System . . . 462
28. Files and Directories, Part 2 . . . 467
28.1. Files and times . . . 467
28.2. More about files . . . 467
28.3. More about Blob . . . 473
28.4. seek . . . 473
28.5. tell . . . 473
28.6. IO::CatHandle . . . 474
28.7. opened . . . 474
28.8. lock . . . 474
28.9. splitdir . . . 474
28.10. splitpath . . . 474
28.11. PDF . . . 475
29. Calling Other Languages . . . 477
29.1. Nativecall . . . 477
29.2. Other Languages . . . 477
29.3. Native Types . . . 478
30. Unicode . . . 479
30.1. unival . . . 479
30.2. Unicode Properties . . . 480
30.3. unimatch . . . 480
30.4. samemark . . . 480
30.5. Unicode Normalization Form . . . 481
30.6. Collation . . . 481
31. Arrays and Lists, part 2 . . . 485
31.1. rotor . . . 485
31.2. zip . . . 485
31.3. cross . . . 487
31.4. polymod . . . 488
31.5. Map Variations . . . 490
31.6. Reduction . . . 491
31.7. ff (Flipflop Operator) . . . 494
32. Misc . . . 495
32.1. Variable declaration . . . 495
32.2. Currying . . . 498
32.3. Multiple Dispatch Revisited . . . 501
32.4. .+ . . . 502

32.5. Feed Operators 504
32.6. tree 504
32.7. proceed / succeed 505
32.8. Higher Order Functions 505
32.9. classify 505
32.10. classify-list 506
32.11. Complex Numbers 506
32.12. produce 507
32.13. EVAL 507
32.14. Databases 509
32.15. Phasers 509
Appendix 1. Docker 513
Installing Rakudo Star with Docker 513
Appendix 2. Solutions 517
Chapter 1 517
Chapter 2 517
Chapter 3 518
Chapter 4 519
Chapter 5 520
Chapter 6 524
Chapter 7 524
Chapter 8 527
Chapter 9 531
Chapter 10 531
Chapter 11 533
Chapter 12 536
Chapter 13 537
Chapter 14 544
Chapter 15 546
Chapter 16 550
Chapter 17 551
Chapter 19 555
Chapter 20 556
Chapter 21 556
Chapter 23 563
Chapter 24 567
Chapter 25 571
Chapter 26 572
Chapter 27 573
Chapter 28 575
Chapter 30 578

Chapter 32 . . . 579
Appendix 3. Beware of . . . 583
A3.1 length . . . 583
A3.2 Objects are not strings . . . 583
A3.3 See also . . . 583
A3.4 Syntax Summary . . . 583
Appendix 4. Raku Background and History . . . 585
6.a and 6.b . . . 585
6.c . . . 585
6.d . . . 585
6.e . . . 585
About Versions . . . 585
Appendix 5. Advanced Docker . . . 587
Terminology . . . 587
Offline Perl 6 Documentation . . . 587
Docker Security Issues . . . 587
Fixing Less . . . 587
Shipping Applications as Docker Images . . . 588
Docker Aliases . . . 589
More Information About Docker . . . 590
Index . . . 591

Index

@

!

Mandatory Argument, 170

Negation, 28

Private Attribute, 275

!! (?? !!) if-then-else, 63

!!! (Stub Operator), 163

!=, 44

!~ (Negated Smartmatch Operator), 181

# (Comment), 18

$

Anonymous State Variable, 265

Anything, 16

$! (Error Variable), 323

$*ARGFILES, 217

$*COLLATION, 481

$*CWD (Current Directory), 231

$*DEFAULT-READ-ELEMS, 222

$*DISTRO, 437

$*ERR (STDERR), 214

$*HOME (Home Directory), 351

$*IN (STDIN), 214

$*KERNEL), 436

$*KERNEL.signals, 327

$*OUT (STDOUT), 214

$*PERL, 435

$*SPEC.tmpdir, 220

$*TMPDIR, 220

$*TOLERANCE, 76

$*TZ (Time Zone), 243

$*USER, 435

$*VM, 435

$/ (Match Object), 185

$?CLASS, 435

$?FILE, 435

$?LINE, 435

$?NL, 83

$_ (Procedure Argument), 152

$_ (Topic Variable), 53

%

Anonymous State Variable, 265

Hashes, 16

Modulo Operator, 77

%% (Divisibility Operator), 77

%*ENV, 229

%*ENV<PATH>, 229

&

Code, 16

Conjunction (Regex), 313

Junction Operator, 427

&&

And, 29

Conjunction (Regex), 312

&?ROUTINE, 435

( )

Capturing (Regex), 186

Grouping Operator, 31

)>

End of Capture (Regex), 317

*

Multiplication Operator, 20

Slurpy Operator, 171

Whatever Star, 122, 500

** (Exponentiation Operator), 78

+

Addition Operator, 19

Numeric Prefix, 42

++

Decrement Operator, 20

Increment Operator, 20

, (List Operator), 113

- (Subtraction Operator), 19

--> (Return Value Constraint), 157

--doc, 159

-> (Read Only Block Parameters), 175

.

Method Call, 277

Public Attribute, 275

.* (method invocation), 304

.+ (method invocation), 303, 502

.. (Ranges), 49, 255

...

Sequences, 259

Stub Operator, 163

Stubbed Class, 276

.? (method invocation), 303

.orig (Regex), 188

.postmatch (Regex), 188

.prematch (Regex), 188

.target (Regex), 188

/ (Division Operator), 20
/.../ (Regex), 180
// (Defined Or Operator), 30
3 4, 579
:= (Binding), 21
:ALL (Module import tag), 332
:D (Defined Adverb), 40
:DEFAULT (Module import tag), 331
:MANDATORY (Module export tag), 332
:U (Undefined Adverb), 41
:delete (Hashes), 146
:exists (Hashes), 146
:g (global), 194
:global, 194
:i (:ignorecase), 193
:ignorecase, 193
:ii (:samecase), 197
:my (Regex), 316
:nth (Regex), 197
:samecase, 197
:with
  cross, 488
  zip, 486
:x (Regex), 197
<, 44
<( )> (Capture Markers) Regex, 187
<)
  Start of Capture (Regex), 317
<-> (Read Write Block Parameters), 175
<< (Hyper Operator), 387
<=, 44
<== (Feed Operator), 504
<=> (Numeric Comparison), 46
<xxx> (Quote Word), 114
= (Assignment Operator), 21
=:= (Container Identity Operator), 47
== (Numeric Equal operator), 44
===
  Objects, 283
  Value Identity Operator, 47
==> (Feed Operator), 504
=> (Pair Constructor), 137
=~= (Approximately-equal Operator), 76
>, 44
>=, 44
>> (Hyper Operator), 387
>>. (Hyper Method Call Operator), 386
? (Boolean Prefix), 28
?&, 42
?? (?? !!) if-then-else, 63
??? (Stub Operator), 163
?^, 42
?|, 42
@
  Anonymous State Variable, 265
  Arrays, 16
@*ARGS, 157
@_, 152
[ ]
  Array Constructor, 114
  Non-capturing grouping (Regex), 187
  Reduction Metaoperator, 492
^ (Junction Operator), 428
^Int (Range), 55
^^ (Exclusive Or), 30
^can, 299
^methods, 438
^mro (Method Resolution Order), 35
^name, 34
_ (in numbers), 71
{ }
  Block Constructor, 49
  Code Block (Regex), 315
  Hash Constructor, 139
|
  Flattening Operator, 120
  Junction Operator, 428
  Longest Alternation (Regex), 312
||
  Alternation (Regex), 311
  Or, 29
~
  String Concatenation Operator, 24
  String Prefix, 43
~~ (Smartmatch Operator), 181
∞ (Infinity), 39

A

A Web Publication System, 462
Adding Operators, 393

Addition Operator (+), 19
Administration Modules, 203
Advanced Regexes, 307
Adverbial Syntax, 170
Adverbs, 170
Alternation (Regex) ||, 311
Anchors (Regex), 313
And (&&), 29
Anonymous State Variable
  $, 265
  %, 265
  @, 265
Any, 38
Anything ($), 16
Approximately-equal Operator (=~=), 76
Array Constructor [ ], 114
Array Slice, 114
Arrays
  @, 16
  Shaped, 125
Arrays and Lists, 113, 485
Assignment Operator (=), 21
Asyncronous Execution, 369
Atomic Int / Operators, 389
add (IO.add), 469
after, 47
all (Junction Operator), 427
also
  does (Roles), 298
  is (Inheritance), 296
and, 29
anon, 154, 498
antipair (Pair), 138
antipairs (Hashes), 145
any (Junction Operator), 428
api (Modules), 344
append, 118
arity (Introspection), 443
assuming (Currying), 499
augment, 398
await (Promises), 374

B

BUILD, 280
Backwards References (Regex), 192, 315
Bag, 409
BagHash, 411
Bailador, 456
Beware of, 583
Binary
  Binary Large Object (Blob), 225
  Files, 223
  Numbers, 69
Binary Files
  Detecting, 227
Binding (:=), 21
Bitwise Operators, 427
Blob (Binary Large Object), 225
Block Constructor { }, 49
Blocks, 49, 175
Bool, 28, 42
Bool is an Enum, 426
Boolean, 27
  Operators, 28
Buf (Buffer), 224
base, 69
before, 46
bless, 280
but
  Objects, 298
  Scalar, 37

C

CATCH, 323
CPAN, 204
Calling Other Languages, 477
Capture Markers <( )>, 187
Character Classes (Regex), 188
Classes, 275
Closures, 272
Code (&), 16
Collation, 481
Colon Synatx for Method calls, 278
Colon Syntax (Methods), 70
Command Line Completion, 7, 352
Command Line Options, 9
Comments (#), 18
CompUnit::Repository, 345
Comparison Operators, 44
Compile Time Errors, 9

Complex, 506
Conjunction (Regex)
  &, 313
  &&, 312
Constants, 26
Container, 15
Container Identity Operator (=:=), 47
Cro, 458
Cro::WebApp, 462
Cross Meta Operator (X*), 488
Cross Operator (X), 488
Current Directory ($*CWD), 231
Currying, 498
  Placeholder Variables, 500
  Whatever Star, 500
  assuming, 499
Custom Exceptions, 325
Custom Operator Precedence, 396
callsame, 503
callwith, 503
ceiling, 74
channel, 387
chars, 97
chmod (File Change Mode), 467
chomp, 84
chop, 83
chr (Unicode), 100
chrs (Unicode), 100
class, 275
classify, 505
classify-list, 506
close (File Close), 214
cmp (Smart Comparison), 45
codes, 98
coll, 483
collate, 483
comb, 102
combinations, 135
constant, 26
contains (Partial Strings), 182
copy, 472
count (Introspection), 442
cross, 487

D

DEFINITE, 510
Databases, 509
Date, 238
Date and Time, 237
DateTime, 240
Debugger, 445
Deck of Cards, 269
Decrement Operator (--), 20
Defined Or Operator (//), 30
Delegation (handles), 305
Denominator, 73
Directories, 229
Directory
  Current ($*CWD), 231
  Home ($*HOME), 351
  Make (mkdir), 234
  Remove (rmdir), 235
  Test (IO.d), 222
Divisibility Operator
  %% (Numeric), 77
Division Operator
  /, 20
  div (Int), 20
Docker, 513, 587
decode (Blob), 473
deepmap, 122
default, 349
defined, 41
delay (Supply), 388
die, 321
dir, 229
div (Integer division Operator), 20
do, 64
docs.raku.org, 10
does, 298
  Objects, 299
  Scalar, 38
duckmap, 490

E

EVAL, 507
Eager, 255
Embedded Comments, 18
Empty, 114

Enum, 425
Enums
  value, 425
Error Messages, 9
Exceptions, 321
Exclusive Or
  ^^, 30
  xor, 30
Exercise
  1.1, 7, 517
  1.2, 9, 517
  1.3, 13, 517
  2.1, 21, 517
  2.2, 26, 518
  2.3, 30, 518
  3.1, 40, 518
  3.2, 47, 519
  4.1, 52, 519
  4.2, 53, 519
  4.3, 66, 520
  4.4, 68
  5.1, 69, 520
  5.2, 72, 521
  5.3, 76, 521
  5.4, 78, 522
  5.5, 81, 523
  5.6, 81, 523
  6.1, 95, 524
  7.1, 100, 524
  7.2, 102, 525
  7.3, 103, 525
  7.4, 107, 526
  8.1, 121, 527
  8.2, 128, 527
  8.3, 131, 528
  8.4, 132, 529
  8.5, 133, 529
  8.6, 134, 529
  8.7, 135, 529
  9.1, 148, 531
  9.2, 149, 531
  10.1, 161, 531
  10.2, 162, 532
  10.3, 174, 533
  11.1, 192, 533
  11.2, 199, 534
  11.3, 200, 535
  12.1, 207, 536
  12.2, 209, 536
  13.1, 214, 537
  13.2, 226, 538
  13.3, 231, 538
  13.4, 231, 540
  13.5, 232, 541
  13.6, 234, 542
  14.1, 240, 544
  14.2, 240, 545
  15.1, 251, 546
  15.2, 252, 547
  15.3, 252, 548
  15.4, 252, 548
  15.5, 252, 549
  15.6, 252, 549
  16.1, 257, 550
  16.2, 270, 550
  16.3, 270, 551
  16.4, 273, 551
  17.1, 285, 551
  17.2, 287, 552
  17.3, 287, 553
  17.4, 288, 554
  17.5, 297, 554
  19.1, 327, 555
  19.2, 328, 555
  20.1, 342, 556
  21.1, 350, 556
  21.2, 358, 556
  21.3, 358, 559
  21.4, 363, 562
  21.5, 366, 562
  21.6, 367, 562
  21.7, 367, 563
  23.1, 397, 563
  23.2, 399, 564
  23.3, 400, 565
  23.4, 400, 567
  24.1, 407, 567
  24.2, 412, 568
  24.3, 419, 569
  24.4, 419, 570

  25.1, 426, 571
  25.2, 429, 572
  26.1, 440, 572
  27.1, 455, 573
  27.2, 455, 574
  27.3, 462, 575
  28.1, 467, 576
  28.2, 475, 577
  30.1, 480, 578
  31.1, 488, 579
  31.2, 490, 579
  32.1, 507, 579
  32.2, 508, 580
Exponentiation Operator
  **, 78
  exp, 78
Extending Perl 6, 391
e, 72
eager, 257
elems, 116
else, 59
elsif, 59
emit (Supply), 389
encode (Blob), 473
end, 116
ends-with (Partial Strings), 184
enums, 425
eof, 474
eq (String Equal operator), 44
eqv (Equivalence), 283
exit, 321
exp (Exponentiation Operator), 78
expmod, 79
extension (IO.extension), 470

F

FALLBACK (method), 300
Factorial Operator, 393
Failures, 320
Fallback Method, 300
False, 27
Feed Operator
  <==, 504
  ==>, 504
Feed Operators, 504
Fibonacci
  Numbers, 163
  Sequence, 260
  Timing, 249
File
  Change Mode (chmod), 467
  Close (close), 214
  Events (IO::Notification), 383
  Open (open), 214
  Remove (unlink), 220
  Test (IO.e IO.r ...), 222
File tests, 222
Files and Directories, 211, 467
Files and times, 467
Flattening Operator (|), 120
Flip-Flop Sequence, 262
Floating Point Numbers, 71
Frugal Regex (non-greedy), 191
fail, 320
fc (Fold Case), 106
ff (Flipflop Operator), 494
fff (Flipflop Operator), 494
first, 129
flat, 120
flatmap, 490
flip, 103
floor, 74
fmt, 93
for, 51

G

GitHub, 204
Graphemes (Unicode), 97
Grouping Operator ( ), 31
gather, 266
gcd (Greatest Common Divisor), 75
ge, 44
get, 217
get_value, 444
getc, 220
gist, 84
given, 60, 349
grab, 419
grabpairs, 421
grep, 128, 200

gt, 44

H

HOW, 445
Hash, 138
Hash (method, 148
Hash Constructor { }, 139
Hashes
    %, 16
    antipairs, 145
    delete, 146
    exists, 146
    invert, 144
    keys, 140
    pairs, 145
    values, 141
Hashes and Relatives, 403
Heredocs, 110
Hexadecimal Numers, 69
Home Directory ($*HOME), 351
Hyper Method Call Operator (>>.), 386
Hyper Operator
    <<, 387
    >>, 387
handles (Delegation), 305
has, 275
has_accessor, 443
head, 129
hidden, 445
hides, 445
hyper, 384

I

INIT, 245
INIT (Phasers), 510
IO
    IO.d (Directory Test), 222
    IO.e (File Exist Test), 222
    IO.lines, 211
    IO.r (File Readable Test), 222
    IO.w (File Writable Test), 222
    IO.words, 212
IO.accessed, 467
IO.add, 469
IO.changed, 467
IO.copy, 472
IO.extension, 470
IO.modified, 467
IO.move, 471
IO.parent, 469
IO.resolve, 470
IO.sibling, 469
IO::CatHandle, 474
IO::Notification, 383
IO::mode (File Permissions), 468
IPv4, 307
Immutable, 403
Increment Operator (++), 20
Inf (Infinity), 39
Infinite Loops, 55
Infinity, 39
Inheritance, 294
    is, 295
Inheritance Tree
    Int, 35
    Str, 94
Instant, 238
Int, 34
    Inheritance Tree, 35
Introspection, 433
if, 59
if-then-else (?? !!), 63
im (Complex), 506
import (Module Loading), 345
indent, 112
index (Partial Strings), 182
indices(Partial Strings), 183
indir, 230
infinite (Range), 257
invert (Hashes), 144
is (Inheritance), 295
is copy (Procedure), 166
is dynamic, 498
is equiv, 396
is export, 247, 331
is hidden, 445
is looser, 396
is required, 170
is rw
    Block, 175

Class, 279
Procedure, 166
is tighter, 396
is-int (Range), 257
is-lazy, 255
is-prime, 76
is-win, 438
isNaN, 76
isa, 48

J

Junction, 427
join, 101

K

key (Pair), 138
keyof (Hash), 142
keys (Hashes), 140
kv, 141
kxxv (Bag), 411
kxxv (BagHash), 422

L

LABEL, 67
Lazy, 255
Leap Seconds, 237
Linenoise, 7
List Coersion, 257
List Operator (,), 113
List Repetition Operator (xx), 127
List Repetition Operator and Sequences, 263
Longest Alternation (Regex) |, 312
Lookahead assertions (Regex), 314
Lookbehind assertions (Regex), 315
Loop Manipulation, 65
Loop Summary, 58
last, 66
lastcall, 504
lazy, 256, 261
lazy vs gather/take, 269
lc (Lower Case), 105
lcm (Least Common Multiple), 75
le, 44
leg (String Comparison), 46
length (not implemented), 583
let, 496
lines, 212
lines (IO.lines), 211
lines (Method), 214
link, 471
live (Supply), 389
lock (IO), 474
log, 79
log10, 79
lookup (Introspection), 445
loop, 50
lt, 44

M

MAIN, 158
Map, 405
Match Object ($/), 185
Math::Matrix (Module), 126
Math::Trig (Module), 80, 209
Matrix, 126
Metaoperator, Reduction [ ], 492
Method Call (.), 277
Methods
Colon Syntax, 70
Mix, 423
MixHash, 423
MoarVM, 5
Module Loading
import, 345
need, 345
require, 344
use, 207, 344
Module Manager
panda, 203
zef, 203
Modules, 203
Modules, Administration, 203
Modulo Operator
% (Numeric), 77
mod (Int), 77
Multi-line Comments, 18
Multi-line Strings (Heredocs), 110
Multiple Dispatch, 162, 299, 501
Multiplication Operator (*), 20
Mutable, 403

m/…/ (Regex), 181
map, 121
match (Regex), 316
max, 130
maxpairs, 422
merge (Supply), 389
method, 276
method_table ((Introspection), 445
min, 130
minmax (Range Constructor), 258
mkdir (Make Directory), 234
mod (Modulo Operator), 77
move, 471
msb (Most Significant Binary), 75
multi
  Methods, 299
  Procedures, 162
my, 17
my (dynamic variable), 497

N

NFC (Unicode), 481
NFD (Unicode), 481
NFKC (Unicode), 481
NFKD (Unicode), 481
NQP, 5
NaN (Not a Number), 75
Native Types, 478
NativeCall, 477
Negated Smartmatch Operator (!~), 181
Negation
  !, 28
  not, 28
Newlines, 83, 474
Nil, 38
Non-capturing grouping (Regex) [ ], 187
Non-greedy Regex, 191
Not a Number (NaN), 75
Num, 71
Numbers, 25, 69
Numerator, 73
Numeric, 42
Numerical Operators, 19
name, 445
name (^name), 34
narrow, 73
ne, 44
need (Module Loading), 345
new, 275
next, 66
nextcallee, 504
nextsame, 504
nextwith, 504
nodemap, 490
none (Junction Operator), 428
not (Negation), 28
note, 215
now, 237
nude, 73

O

Object (Everything is an Object), 36
Object Comparison, 282
Octal Numbers, 69
One Liners, 8
Operator Precedence, 21
Operator Types, 393
Operators
  Boolean, 28
  Junction, 427
Or (||), 29
Other Builtin Types, 425
Other Operators, 78
Output, 86
of
  keyword, 33
  method, 34
  subset of, 391
of (Hash), 142
on-close (Supply), 389
on-switch (IO::CatHandle), 474
once, 65
one (Junction Operator), 428
open (File Open), 214
opened (IO), 474
or, 29
ord (Unicode), 99
ords (Unicode), 99
orwith, 62
our, 495

P

Pair, 137
  Constructor (=>), 137
  antipair, 138
  key, 138
  value, 138
Pair and Hashes, 137
Parallel Execution, 384
Partial Strings
  contains, 182
  ends-with, 184
  index, 182
  indices, 183
  rindex, 183
  starts-with, 184
  substr, 103
  substr-rw, 104
Path (%*ENV<PATH>), 229
Perl 6, 5
Phasers, 245, 509
Placeholder Variables, 121, 152
Placeholder Variables (Currying), 500
Pod, 344
Precompilation, 203
Preventing Runtime errors with try, 43
Prime Numbers, 76
Private Methods, 292
Proc, 366
Proc::Async (module), 369
Procedures, 151
Promise, 383
Promises, 369
Public Attribute (.), 275
Public Class Variables, 279
p6doc, 10
pack (Blob), 473
pairs (Hashes), 145
panda (Module Manager), 203
parent (IO.parent), 469
parents ((Introspection), 445
parse-base, 70
path-sep, 438
perl, 86
permutations, 135
pi, 27, 71
pick, 131
pickpairs, 422
pm6 (Filename Extention), 247
poll (Channel), 388
polymod, 488
pop, 115
pred, 108
prepend, 118
print, 15, 87
printf, 88
proceed, 505
produce, 507
prompt, 93, 216
proto, 502
prove6, 332
push, 115
put, 87

Q

Q (Quoting), 109
Queue, 117
Quote
  Execute (qx), 367
Quoting, 23, 109
q (Quoting)), 109
q:c (Quoting), 109
qq (Quoting), 109
qqw (Quote Words with interpolation), 110
qqx (Quote eXecute), 367
quietly, 319
qw (Quote Words), 110
qx (Quote eXecute), 367

R

REPL, 6
Raku Background and History, 585
Rakudo, 5
Rakudo Star, 5
Random Primes, 172
Random Values, 130
Range Constructor
  .., 255
  minmax, 258
Range Operator (..), 255
Ranges, 49, 255

Ranges and Sequences, 255
Rat, 72
Rational Numbers, 72
Reading Files, 211
Recursion, 164
Reduction, 491
    reduce, 491
Reduction Metaoperator [ ], 492
Reflection, 433
Regex (type), 182
Regex Adverbs, 196
Regex Intro, 179
Return Value Constraint
    -->, 157
    returns, 157
Roles, 297
Rotate 13, 199, 398
Rounding, 74
Running External Programs, 347
Running Programs, 9
Runtime Errors, 9
race, 386
rand, 131
re (Complex), 506
react, 389
read, 224
readchars), 221
rebless, 297
receive (Channel), 388
redo, 67
reduce, 491
regex, 310
rename, 472
repeat until, 57
repeat while, 57
repeated, 127
require (Module Loading), 344
resolve (IO.resolve), 470
return, 155
return-rw, 156
returns (Return Value Constraint), 157
reverse, 124
rindex (Partial Strings), 183
rlwrap, 8
rmdir (Remove Directory), 235
role, 297
roll, 133
roots (Complex), 506
rotate, 118
rotor, 485
round, 74
roundrobin, 486
run, 347
rx/.../ (Regex), 193

S

S/.../.../ (String Substitution), 195
STDERR ($*ERR), 214
STDIN ($*IN), 214
STDOUT ($*OUT), 214
Scalars ($), 16
Sequence Operator (...), 259
Sequences, 259
Set, 406
Set Operators, 407
SetHash, 408
Shaped Array, 125
Shaped Hash, 143
Sigilless variables, 27
Sigils, 15
Signals, 327
Slurpy Operator (*), 171
Smartmatch Operator (~~), 181
Soft Errors, 326
Special Values, 38
Stack, 117
State Variable (state), 263
Str, 34, 43
    Inheritance Tree, 94
Str (Method), 85
String Concatenation (~), 24
String Repetition Operator (x), 107
String Substitution, 193
    subst, 195
Stringification, 84
Strings, 23, 97
Strong Typing, 33
Stub Operator
    !!!, 163
    ..., 163

???, 163
Stubbed Class (...), 276
Subrules (Regex), 310
Subtraction Operator (-), 19
Supply, 388
Switch (given/when), 349
s/.../.../ (String Substitution), 194
samemark (Unicode), 480
samewith, 504
say, 15, 86
say on Object, 289
seek (IO), 473
self, 279
send (Channel), 388
set, 406
set_name, 445
set_value, 444
shape, 126
share (Supply), 389
shell, 348, 450
shift, 115
sibling (IO.sibling), 469
sign, 73
signal, 327
signature (Introspection), 443
sink, 375
sleep, 373
sleep-timer, 374
sleep-until, 374
slurp, 213
so, 28, 42
sort, 123
splice, 119
split, 101, 200
splitdir (IO), 474
splitpath (IO), 474
sprintf, 92
spurt, 216
sqrt, 78
squish, 126
srand, 134
stable (Supply), 389
started (Proc::Async), 376
starts-with (Partial Strings), 184
state (State Variable), 263
status (Promise), 376
submethod, 296
subset, 391
subst (String Substitution), 195
substr (Partial Strings), 103
substr-rw (Partial Strings), 104
succ, 108
succeed, 505
sum, 147
symlink, 471

T

TODO, 149, 151, 163, 201, 209, 318, 320, 321, 327, 328, 343, 344, 345, 367, 369, 374, 376, 383, 386, 388, 389, 392, 403, 407, 419, 420, 424, 433, 434, 436, 438, 439, 440, 442, 443, 444, 445, 447, 455, 456, 458, 462, 473, 474, 475, 477, 479, 480, 481, 491, 492, 493, 495, 498, 502, 503, 504, 505, 506, 507, 509, 556, 563, 567
TR/.../.../ (Transliteration), 199
Tags (Modules), 331
Temporary Files, 219
Testing Code, 332
Time Zones, 243
Time-Code (Module), 247, 374, 429
Timing, 244
Code, 245
Fibonacci, 249
Programs, 244
Topic Variable ($_), 53
Transliteration, 198
Trigonometric Functions, 80
True, 27
Twigils, 16
Type Constraints
Procedure Arguments, 154
Procedure Return Value, 157
Type System, 33
Typed Array, 125
Typed Hash, 142
tail, 129
take, 266
take-rw, 271
tap (Supply), 389
tap (on Signals), 327

tau, 72
tc (Title Case), 105
tclc (Title Case Lower Case), 106
tell (IO), 473
temp, 496
term:<>, 401
then (Promise), 383
throw, 322
time, 237
tmpdir ($*SPEC.tmpdir), 220
total, 423
tr/.../.../ (Transliteration), 198
trans (Transliteration), 198
tree, 504
trim, 199
trim-leading, 199
trim-trailing, 199
truncate, 74
trusts
  Classes, 292
  Methods, 294
try, 43, 322

U

Unicode, 97
  Graphemes, 97
  NFC, 481
  NFD, 481
  NFKC, 481
  NFKD, 481
  Numbers, 70
  chr, 100
  chrs, 100
  ord, 99
  ords, 99
  samemark, 480
  unicmp, 45
  unimatch, 480
  uniname, 98
  uninames, 99
  uniparse, 99
  unival, 479
  univals, 480
uc (Upper Case), 106
unicmp (Unicode), 45
unimatch (Unicode), 480
uniname (Unicode), 98
uninames (Unicode), 99
uniparse (Unicode), 99
uniprop, 191
uniprops, 191
unique, 126
unit
  unit class, 290
  unit module, 247
  unit procedure, 158
unival (Unicode), 479
univals (Unicode), 480
unless, 59
unlink (Remove File), 220
unpack (Blob), 473
unpolar (Complex), 506
unshift, 115
until, 56
unwrap, 503
use
  Module Loading, 207, 344
  use MONKEY-SEE-NO-EVAL, 507
  use MONKEY-TYPING, 398
  use lib, 248
use fatal, 321
use v6, 10

V

VAR, 35
Value Identity Operator (===), 47
Values, 23
Variable Names, 25
Variables, 15
value
  Enums, 425
  Pair, 138
values (Hashes), 141

W

WHAT, 34
WHERE, 445
WHY, 159
Warnings, 319
  Failures and Exceptions, 319

Web Frameworks, 456
Web Programming, 447
Whatever Star
- *, 122
- Currying, 500

Writing Documentation, 344
Writing Modules, 329
Writing a Module, 247
warn, 319
when, 64, 349
while, 55
with, 60
without, 63
wordcase, 106
words, 101
words (IO.words), 212
wrap, 503
write, 226

X

X (Cross Operator), 488
X* (Cross Meta Operator), 488
x (String Repetition Operator), 107
xor (Exclusive Or), 30
xx (List Repetition Operator), 127

Y

Yada, yada, yada Operator, 163
yada, 163

Z

Z (Zip Operator), 486
Z* (Zip Meta Operator), 486
Zero-width assertions (Regex), 314
Zip Meta Operator (Z*), 486
Zip Operator (Z), 486
zef (Module Manager), 203
zip, 485
zip-latest (Supply), 389