

Index

Symbols

- + operator, 95, 139
- += operator, 132
- == comparison operator, 93, 136–137, 193, 271
- < and > comparison operators, 94

A

- acos function, 236
- arc4random function, 178
- Array
 - append method
 - existing array, 131–132
 - item, 132–133
 - capacity, 153
 - concatenation, 139
 - convert JSON string to
 - array, 142–143
 - count property, 230
 - dictionary's values
 - property, 151–152
 - element, 133
 - empty property, 134
 - enumerated method, 139–140
 - equal, 135–136
 - filter method, 149–150, 174
 - filter element, by index, 173
 - first element, 154–155
 - index element, 150–151
 - index element by value, 154
 - index/indices of item, 156–158
 - initializer, 145
 - insert method, 164, 165, 170–171
 - join separator method, 165–166
 - to JSON string, 141–142
 - last element, 158–159
 - last X elements, 159–160
 - with literal, 146
 - loop, 168
 - map method, 169
 - maximum value, 160
 - minimum value, 161
 - NSArray object, 147–148
 - object type, 134–135
 - random element, 161–162
 - range creation, 144–145
 - reduce method, 171–172, 230, 232–233, 240
 - removal
 - array item by value, 174
 - duplicate elements, 174–175

INDEX

Array (*cont.*)

- removeAll method, 172–173
 - removeFirst method, 177
 - removeLast method, 170
 - reverse method, 166–167,
175–176
 - shuffle method, 178
 - size, 162–163
 - sorted method, 231
 - stride, 145
 - string object, 143–144
 - swap elements, 185–186
 - tuples
 - combined array, 137
 - equal, 136–137
 - index, 163
 - pattern matching, 164
 - valid indices, 152
 - X elements, 155–156
- asin function, 237
- Asynchronous task,
execution, 353–355
- atan function, 238

B

- Barebones class, 61
- Bundle
 - executables'
 - path, 339–340
 - frameworks'
 - path, 340–341
 - identifier, 338
 - main path, 343
 - resource path, 315–316

C

- Capacity property, 153
- Class
 - barebones, 61
 - defining, 61–62
 - existing, 73–74
 - inheritance, 72–73
 - initializers
 - convenience, 67–68
 - default, 66–67
 - multiple, 69–70
 - instance methods, 64–65
 - instance variables, 62–63
 - method overloading, 70–71
 - singleton, 75–76
 - type methods, 65–66
- Closure
 - with arguments, 77–78
 - defining, 76–77
 - inline method, 79
 - with arguments, 80
 - shorthand arguments, 80–81
 - with returns value, 78
- Command line
 - arguments, 338–339
 - read input, 348
- component(_:from:) method, 280,
282–283, 286
- components(separatedBy:)
method, 124, 126
- Conditional statement
 - if, 17
 - if/else, 18
 - if/else/if, 18–19

Constant, 9

Constructs and conventions

- creating Hello World, 7
- declare constants, 9
- declare multiple variables, 10
- declare range, 11–12
- declare Unicode
 - names, 10–11
- declare variables, 8
- enumeration, 12–13
- type alias, 14
- use comments, 15–16
- use semicolons, 14–15

Conversion, numbers

- angle units, 246–247
- area units, 247–248
- degrees to radians, 251
- double to integer, 252
- double to string, 252–253
- float to CGFloat, 253–254
- float to integer, 254–255
- float to string, 255
- integer to double, 256
- integer to float, 256–257
- integer to string, 257–258
- length units, 248–249
- radians to degrees, 258
- volume units, 249, 251

Cosine of an angle, 234

Custom error

- assert function, 295–296
- with associated
 - values, 289–290
- creation, 287–288

- throw and catch, 288–289
- throwing function
 - and catch, 290–292
 - guard statements, 294–295
 - optional values, 292–293

D

Date

- adding date, 266
- comparing two different
 - dates, 272
- components, 278–279
- creating
 - from given date, 273–274
 - date object from
 - current date, 273
 - from string format, 274–275
- day component, 279–280
- difference between
 - dates, 280–281
- equal, two different
 - dates, 270–271
- in future, 267–268
- month from date, 281–282
- month name from date,
 - 282–283
- past, 268–269
- seconds difference between
 - times, 283–284
- string format, 277
- today, 269–270
- Unix timestamp, 284–285
- year component, 285–286

INDEX

date(byAdding:value:to:)
 method, 266

DateComponents class, 279

DateFormatter class, 277

Delay, creating, 355

Dictionary

- add key-value pair, 203–204
- check equal, 207–208
- check key exists, 205–206
- check object, 206–207
- convert to array of
 tuples, 208–209
- convert to JSON
 string, 209–210
- create NSDictionary, 214–215
- definition, 203
- [:] empty dictionary
 expression, 213–214
- get key value pairs, 215–217
- get size, 217–218
- isEmpty property, 204–205
- to JSON string to swift, 211
- with literal, 212–213
- loop, 218
- merge dicB to
 dicA, 219–220
- read property list, 220–222
 Plist from string, 222–223
- remove elements, 223–224
- remove item, 224–225
- set value by key, 225–226
- values property, 152

Disjoint(with:) method, 194

E

Empty Array object, creation, 147

[:] empty dictionary expression
 approach, 213

enumerated method, 140, 167–168

Enumeration, 23

Enumeration type, 13

Error protocol, 288–289, 292

exp function, 235

F, G

Files

- attributes, 312–313
- copying to path, 303
- deletable, 298–299
- deleting at path, 306–307
- executable, 299
- exists, 297–298
- moving to path, 317–318
- readable, 300
- reading data, 319–321
- size of, 316–317
- writable, 301
- writing data, 322–323
- writing string contents, 323–324

Find date, specific string, 276

First property, 155

Functions

- with argument labels, 39–40
- with arguments, 38–39
- call nested functions, 56
- closure argument, 50–51

- with default parameter, 42–43
- defining, 37–38
- generics and type
 - constraints, 54–55
- generic type, 52–53
- in-out arguments, 47–48
- nested functions, 55
- returns multiple values, 44–45
- with return value, 43–44
- return value, ignorance, 45–46
- trailing closure argument, 51–52
- variadic arguments, 48–49
- without argument labels, 40–41

H

- Home directory path
 - current user, 341–342
 - for user, 342–343
- HTTP POST request, 334, 336
- HTTP web request, 325–327
 - with custom headers, 328–330
 - with JSON data, 331–333

I

- `index(_offsetBy:)`
 - method, 111–112, 115, 118
- `index(of:)` method, 113, 154, 157
- Indices property, 152
- Inheritance, definition, 73
- `insert(_at:)` method, 165
- Int, initializer, 252, 255
- `Int.random(in:)` function, 259–260

J, K

- `joined(separator:)` method, 166
- JSONSerialization class, 142–143
- JSON string, 210

L

- Last property, 159
- Lazy list comprehension, 187
- Lemma-definition approach, 203
- List comprehension, 186
- `_log` function, 239
- `_log2` function, 233
- logical and operator (`&&`), 28
- logical not operator (`!`), 29
- logical or operator (`||`), 30
- logical xor operator, 31
- Loops
 - for, 33–34
 - repeat-While, 35–36
 - while, 34–35

M

- `map` method, 209
- `max` method, 160
- `min` method, 161

N

- Nested functions, 55
- Nil coalescing operator (`??`), 28
- `NSArray(array:)` initializer, 148
- `NSArray` instance, 148

INDEX

NSCache, cache data with, 360–361
NSDataSelector class, 276
NSDictionary's hash property, 215
NSLinguisticTagger class, 126
NSSet(set:) initializer, 196
NSString(string:) initializer, 102
Numbers
 absolute value, 260–261
 average of elements,
 array, 229–230
 base-2 logarithm, 233
 decimal value down to whole
 number, 262–263
 decimal value to nearest whole
 number, 264
 decimal value up to whole
 number, 265
 exponential calculation, 234–235
 inverse cosine, 235–236
 inverse sine, 236–237
 inverse tangent, 237–238
 logarithm, 238–239
 maximum value, 261
 median of elements,
 array, 230–231
 minimum value, 262
 nth factorial, 239–240
 nth root, 240–241
 object is integer, 245
 power calculation, 241
 product of elements,
 array, 231–232
 square calculation, 242–243

square root calculation, 243–244
sum of elements, array, 232–233

O

Operator
 logical and, 28
 logical not, 29
 logical or, 29–30
 logical xor, 30–31
 nil coalescing, 27–28
 ternary conditional, 26–27
ordinality(of:in:for:)
 method, 281, 284

P, Q

padding(toLength:withPad:startingAt:) method, 109

Path

 copying file to, 303
 current directory, 313–314
 deleting file at, 306–307
 deleting folder at, 307–308
 directory at, 304
 directory contents, 314–315
 folder, 302
 moving file to, 317–318
 renaming file, 321–322
 URL from file, 305

Path string

 file extension, extract, 308–309
 filename, extract, 309–310

- folder, extract, 310
- path components, extract, 311
- pow function, 241
- Prefix method, 156
- propertyList method, 222–223

R

- Random number
 - generation, 259–260
- range(of:) method, 114, 121–122
- range(of:options:)
 - method, 87, 89, 92
- remove(at:) method, 173
- removeValue(forKey:) method, 225
- Repeat-While loop, 35
- replaceSubrange(,with:), 120–122
- Reversed method, 167, 176

S

- Scripting language, 3
- Sequence/array of characters, 144
- Sets
 - array element, 176–177
 - array literal, 148–149
 - constructor, 194
 - count property, 199
 - element, 188, 193–194
 - empty array, 194–195
 - empty property, 188–189
 - equal, 192–193
 - intersection method, 197

- loop, 200
- NSSet object, 195–196
- removeAll method, 200–201
- strict
 - subset, 189–190
 - superset, 190–191
- subset, 191
- subtracting method, 196
- superset, 192
- symmetricDifference
 - method, 197–198
- union method, 198–199
- Shorthand argument, 81
- sin function, 242
- Sine of an angle, 242
- Singleton class, 75
- Sorting, array
 - dictionary
 - ascending order, 179–180
 - descending order, 180–181
 - numbers
 - ascending order, 181–182
 - descending order, 182–183
 - strings
 - ascending order, 183–184
 - descending order, 184–185
- StrictSubset(of:) method, 190
- StrictSuperset(of:) method, 191
- String
 - append method, 84
 - array by separator, 126
 - capitalized property, 95–96
 - character, 83–84, 86

INDEX

String (*cont.*)

- character, specific
 - index, 110–113
- < and > comparison operators, 94
- concatenation, 94–95
- constructor, 101
- count property, 115–116
- data object, 96–97
- decimal places, 106–107
- double initializer, 97–98
 - empty, 90–91
- equal, 93
- float initializer, 98
- format
 - left padding, 107–108
 - right padding, 108–109
- hasPrefix method, 92–93
- hasSuffix method, 89–90
- index(.,offsetBy:) method,
 - 111–112, 114–115, 117–118
- initializer, 253, 255
- int initializer, 99
- interpolation, 128–129
- loop characters, 118
- lowercased method, 99–100
- NSDataDetector class, 103–104
- NSString object, 102
- object, 85
- += operator, 84–85
- range(of:) method, 87–88
- reading file contents, 318–319
- regular expression, 86–88, 91–92
- replaceSubrange(.,with:), 120

- replace substring, 122
- replace substring by range, 121
- reverse method, 123
- split by lines, 123–124
- split by words, 124–125
- String(format:.)
 - initializer, 105–106
- String(repeating:count:.)
 - initializer, 119
- substring, 113–114
- trim characters, 127
- trim whitespace, 127–128
- Unicode characters,
 - 116–117, 130
- uppercased method, 100–101
- URL in, 104–105
- URL from, 305–306
- UUID class, 109–110
- web page’s HTML, 324–325
- writing contents, 323–324
- String(format:.) initializer, 106
- String(repeating:count:.)
 - initializer, 108, 119
- Subset(of:) method, 191
- Superset(of:) method, 192
- swapAt function, 186
- Swift
 - compilation, 4
 - execution, 4
 - for Scripting, 5–6
- Switch statement, 19–20, 23
 - with enumerations, 22
 - with intervals, 20–21

- with matched values, 25–26
- with multiple values, 23–24
- using tuples, 24–25

`\(..)` syntax, 129

System

- currency code, 344
- language code, 344–345
- region code, 345–346
- temporary directory
 - path, 346–347

T

Tangent of an angle, 244–245

Terminal command execution, 349

- arguments, 350
- asynchronous, 351–354
- creating delay, 355
- defer code execution, 356–357
- get type of variable, 358
- NSCache, cache data
 - with, 360–361
- observe variable value
 - changes, 359

- handy Process and Pipe

- classes, 350

ternary operator (?), 27

`trimmingCharacters(in:)`

- method, 127, 128

Type alias, 14

U

URL, extract components, 336–337

Username, current user, 347

V

values property, Dictionary, 152

Variables, 8

W, X, Y

WeirdMath class, 70

While loop, 35

Z

zip function, 137–138