

- 1: SWIFT 2.0 Compact Reference
- 2: ©2015 Allen I. Holub. All rights reserved.
- 3: About this reference
- 4: running as a script (in REPL*)
- 5: comments
- 6: ; print NSLog assert @/#available import @testable
- 7: ~=< &+ ??
- 8: operators(2): precedence
- 9: simple declarations
- 10: types, typealias
- 11: type conversion, bridged types
- 12: optionals
- 13: unwrapping optionals with let
- 14: literals
- 15: String
- 16: String Indexing
- 17: arrays, [i] []
- 18: Set, [v]
- 19: dictionaries [k:v] [:]
- 20: tuples
- 21: struct/class basics
- 22: value vs. reference objects
- 23: optional chaining: ?.?[x]
- 24: enum(1)
- 25: enum(2): associated values
- 26: enum(3): rawValue
- 27: enum(4): initializers
- 28: mutating, self= (in struct)
- 29: mutating, self= (in enum)
- 30: repeat, while, if, else, for
- 31: labels
- 32: switch(1)
- 33: switch(2): tuples, value bindings, where
- 34: pattern matching, wildcards (_,), tuples
- 35: pattern matching and enum
- 36: pattern matching and optionals
- 37: pattern matching and flow control
- 38: assert, precondition, etc.
- 39: ErrorType, throws, throw
- 40: defer, guard, do/catch, gen, try!
- 41: basic functions
- 42: internal/external argument names
- 43: default argument values
- 44: variadic parameters (...)
- 45: inout
- 46: function types, nesting
- 47: closures
- 48: OO vs. functional approaches
- 49: captures
- 50: dynamic initialization
- 51: curried functions (1): f(x)(y)
- 52: curried functions(2)
- 53: filter, map, reduce
- 54: @autoclosure, @noescape
- 55: computed properties
- 56: property observers
- 57: subclassing, overriding
- 58: initialization(1)
- 59: destruction (deinit)
- 60: initialization(2): delegation
- 61: initialization(3): phases, using self
- 62: initialization(4): circular references
- 63: initialization(5): failable
- 64: initialization(6): required, in protocols
- 65: access privileges(1)
- 66: access privileges(2)
- 67: type casting: is, as!, as?, AnyObject
- 68: Any
- 69: generics
- 70: extension(1)
- 71: extension examples

72: exact equivalence, === self
73: ARC
74: weak references
75: unowned references
76: capture lists, self in closure
77: basic protocols
78: @objc
79: optional protocol members
80: protocols and is, as, as!, as?
81: class protocols, implicit adoption
82: associated types: typealias, generic protocols
83: protocol extensions
84: subscripts, [â\200;]
85: extending String: str[1â\200;5], Range<T>
86: Matrix, m[i,j]
87: operator overloading(1)
88: operator overloading(2): in protocols
89: custom matching in switch, ~=
90: sequences: for e in list
91: dictionary keys: Hashable
92: Comparable, Equatable
93: CustomStringConvertible â\200;LiteralConvertible =10 ="s" \ (x)
94: ArrayLiteralConvertible =[â\200;]
95: DictionaryLiteralConvertible =[â\200;:â\200;]
96: misc. other â\200;Convertible
97: what is an optional, really? NilLiteralConvertible
98: what is an optional(2) Equatable
99: SWIFT