

Contents

Introduction	6
The Zen of Swift	
How to follow this quick start guide	
What's the difference between Swift and SwiftUI?	
Dedication	
Simple types	12
Why does Swift have variables?	
Why is Swift a type-safe language?	
Why does Swift have multi-line strings	
Why does Swift need both Doubles and Integers?	
Why does Swift have string interpolation?	
Why does Swift have constants as well as variables?	
Why does Swift have type annotations?	
Complex types	22
Why does Swift have arrays?	
Why are sets different from arrays in Swift?	
How are tuples different from arrays in Swift?	
When should you use an array, a set, or a tuple in Swift?	
Why does Swift have dictionaries as well as arrays?	
Why does Swift have default values for dictionaries?	
Why would you want to create an empty collection?	
Why does Swift need enums?	
Why would you want to associate a value with an enum case?	
Why do Swift's enums have raw values?	
Operators and conditions	36

Why can't Swift add a Double to an Int?
Why does Swift have a dedicated division remainder operator?
Why does Swift need operator overloading?
Why does Swift have compound assignment operators?
How does Swift let us compare many types of data?
What's the difference between if and else if?
How to check multiple conditions
When should you use the ternary operator in Swift?
When should you use switch statements rather than if?
Why does Swift have two range operators?

Looping

54

Why does Swift use underscores with loops?
When should you use a while loop?
When should you use a repeat loop?
Why would you want to exit a loop?
Why does Swift have labeled statements?
When to use break and when to use continue
Why would you want to make an infinite loop?

Functions

66

What code should be put in a function?
How many parameters should a function accept?
When is the return keyword not needed in a Swift function?
How can you return two or more values from a function?
Why does Swift use parameter labels?
When should you omit a parameter label?
When to use default parameters for functions
When to use variadic functions
When should you write throwing functions?
Why does Swift make us use try before every throwing function?
When should you use inout parameters?

Closures

84

What the heck are closures and why does Swift love them so much?
Why are Swift's closure parameters inside the braces?
How do you return a value from a closure that takes no parameters?
Why would you want to use closures as parameters?
Why does Swift have trailing closure syntax?
When would closures with parameters be used as parameters?
When would you use closures with return values as parameters to a function?
When should you use shorthand parameter names?
How many parameters can a closure take?
Returning closures from functions
Why do Swift's closures capture values?

Structs

105

What's the difference between a struct and a tuple?
When should you use a computed property or a stored property?
When should you use property observers?
When should you use willSet rather than didSet?
What's the difference between a function and a method?
Why do we need to mark some methods as mutating?
Why are strings structs in Swift?
Why do strings behave differently from arrays in Swift?
How do Swift's memberwise initializers work?
When would you use self in a method?
When should properties be lazy?
What's the point of static properties and methods in Swift?
What's the point of access control?

Classes

129

Why does Swift have both classes and structs?
Why don't Swift classes have a memberwise initializer?
When would you want to override a method?
Which classes should be declared as final?
Why do copies of a class share their data?
Why do classes have deinitializers and structs don't?

Why can variable properties in constant classes be changed?

Protocols and extensions **141**

Why does Swift need protocols?

When should we use protocol inheritance?

When should you use extensions in Swift?

When are protocol extensions useful in Swift?

How is protocol-oriented programming different from object-oriented programming?

Optionals **152**

Why does Swift have optionals?

Why does Swift make us unwrap optionals?

When to use guard let rather than if let

When should you force unwrap optionals in Swift?

Why does Swift need both implicitly unwrapped optionals and regular optionals?

When should you use nil coalescing in Swift?

Why is optional chaining so important?

When should you use optional try?

Why would you want a failable initializer?

When is type casting useful in Swift?