Swift
(Part 2)

Functions

- Use the 'func' keyword to specify a function
- Named parameters
- Default parameter values
- Multiple returns values with tuples

Closures

- Equivalent to Lambdas
- Also known as Blocks in Objective-C
- Anonymous functions (can be assigned to var/let)
- Captures values from enclosing lexical scope
- Trailing closure syntax when used as the last parameter
Classes

- Stored properties (var / let)
- Stored property observation (willSet / didSet)
- Computed properties (var)
- Methods (func)
- Initializers / Deinitializers (init() / deinit())

All properties must have a value after init()

Classes, continued

- Polymorphism via inheritance
- Use 'override' keyword in subclasses to override methods and properties
- Use 'final' keyword to prevent overrides or subclasses
- Classes are reference types

Structs

- Similar to classes, but without polymorphism / inheritance
- Member-wise initializer provided by default
- Mutating keyword required for methods that change values
- Structs are value types
Enums

Can have a raw value assigned to each case
Raw values can be of any type
Associated values

Protocols

Defines a contract that classes, structs, and enums can conform to
Protocols are a type that can be used similarly to other types

Extensions

Applicable to classes, structs, and enums
Add methods and properties to existing types without subclassing
Documentation

“The Swift Programming Language”