Sela.

IosToSwift

Swift Fundamentals for iOS Developers

college@sela.co.il 03-6176666

Swift Fundamentals for iOS Developers

IosToSwift - Version: 2

1 days Course

Description:

Swift is Apple's new programming language for iOS and OS X, which builds upon functional and generic languages and interoperates with Objective-C. In this one-day course you will make the transition from Objective-C to Swift while learning about Swift's syntax in detail, including closures, protocols, extensions, optional types, and many other topics. We will also discuss interoperability between Swift and Objective-C, including within the same project. The course is accompanied by multiple hands-on labs, in which you will experiment with Swift syntax and learn to use the language in the most effective manner. Prior experience with Objective-C development for either iOS or OS X is required.

Intended audience:

IOS/OS X developers who want to learn developing with Swift.

Prerequisites:

At least one year of programming experience with Objective-C for iOS and/or OS X.

Objectives:

Topics:

Introduction to Swift

- Swift vs. Objective-C
- Swift language principles
- The REPL and Playgrounds
- LAB: Getting started with Swift

Variables, Types, and Control Flow

- Variables and constants
- Built-in types
- Conditional statements
- Basic loops
- LAB: Basic control flow

Optional Types

- Optional variables
- Testing and unwrapping

Collections

- Tuples
- Arrays
- Dictionaries
- Sets
- LAB: Collections

Functions and Closures

- Basic functions
- Higher-order functions, nested functions

- Closures
- LAB: Functions

Classes and Structures

- The Swift type system
- Properties
- Initializers
- Methods
- Property observers
- Access modifiers
- Inheritance and polymorphism
- Structures
- LAB: Classes and Structures

Enumerations

- Using enumerations
- Associated values (sum types)

Pattern matching

- Basic pattern matching
- Expression patterns
- Custom expression matching
- LAB: Pattern matching

Memory Management

- Automatic reference counting (ARC)

- Reference cycles
- Weak and unowned references
- Reference cycles with closures

Protocols

- Protocol inheritance and casting
- Some useful protocols

Extensions

- Extending protocols

Generics

- Generic functions
- Generic classes
- Constraints
- Protocols and associated types
- Complex constraints
- LAB: Generics

Operators

- Subscripts
- Overloading operators
- Custom operators
- Associativity and precedence

Error Handling

- Objective-C vs. Swift error handling
- Throwing errors
- Calling functions that throw
- Deferred execution

Interop with Objective-C

- Bridging basic types, classes, methods
- Bridging strings and collections
- Using id and optionals
- Error handling
- API availability
- Mix and match in the same project
- LAB: Mixing and matching

Swift 2.0

- Xcode migrator