Swift language basics for Mac OS X and iOS development

★★★★★

SII-361 **2 Days (14 Hours)**

Description

New programming language created by Apple, Swift presents itself as a modern and more efficient alternative to Objective-C. In this training, you will learn how to develop Swift 3 in a MacOS environment. You will discover the basic concepts of Swift 2 and its innovative aspects.

Who is this training for ?

For whom

Developers, project managers.

Prerequisites

Aucune

Training objectives

- Discover the basic concepts of the Swift language Create a Swift projectLearn Swift Standard Library Understand how to integrate Cocoa and Objective-C code into a Swift application

Training program

Introduction à Swift

- Why a new programming language in the Apple universe? The different use cases of Swift (Mac OS X, iOS).
- The contributions and new features of Swift .
- Swift's relationship with classic Apple tools: Objective-C, XCode.
- Openness to Open Source and other systems.

L'environnement de développement

- Discovery of the XCode environment, new features of XCode 8, migrating from Swift 1 and 2 (Swift Migrator).
- Installation and configuration of the development environment, targeting a specific OS version.
- Creation of a Swift project and basic operations (editing, execution.
-).
- Discovery of the Swift project 'REPL evaluation and development (Read-Eval-Print-Loop).
- Practical work Setting up the development environment and discovery of a first project.

Les concepts de base de Swift

- Philosophy of language: syntax, memory management, type inference, object or functional orientation.
- Classic mechanisms: typing, variables and constants, instructions, functions.
- Object concepts: concept of classes, method.
- The manipulation of character strings.
- Practical work Creation of several applications implementing the concepts presented.

Swift Standard Library

- Types: String, Array, Dictionary and numeric.
- Protocols: Equatable, Comparable, Printable.
- Free functions and algorithms.
- Practical work Implementation of concepts.

Les aspects novateurs et avancés

- Swift's typing mechanisms, genericity, protocol support.
- The concept of a playground (Interactive Playground).
- Closures).
- Memory management: constructor, destructor, reference management, lazy instantiation.
- Error management.
- Multi-threading.
- Introduction to graphical interfaces with Swift (Cocoa, Cocoa Touch, iOS) The future with Swift 3.
- Practical work Discovery of the Playground.
- Putting closures and genericity into practice.
- Using the Cocoa and Cocoa Touch frameworks.

Les aspects novateurs et avancés

- Swift's typing mechanisms, genericity, protocol support.
- The concept of a playground (Interactive Playground).
- Closures).
- Memory management: constructor, destructor, reference management, lazy instantiation.
- Error management.
- Multi-threading.
- Introduction to graphical interfaces with Swift (Cocoa, Cocoa Touch, iOS) The future with Swift 3.
- Practical work Discovery of the Playground.
- Putting closures and genericity into practice.
- Using the Cocoa and Cocoa Touch frameworks.

L'intégration avec Cocoa et Objective-C

- Using the Cocoa and Cocoa Touch frameworks from a Swift application.
- The joint use of Swift and Objective-C.
- The three operating modes : Interoperability, Mix and Match, Migration.
- Practical work Integration of Swift and Objective-C code.