

Formation Swift, développer des applications iOS (iPhone/iPad)

Description de la formation Programmation iOS swift

Ce cours vous permettra de disposer de solides bases pour développer des applications professionnelles autour des incontournables phénomènes de société que sont devenus l'iPhone et l'iPad réunis sous la bannière iOS. Une fois votre environnement de développement et le langage Swift maîtrisés, vous abordez les API du SDK iOS et les aspects essentiels du développement iOS comme l'architecture de vos applications (modèle, vue, etc.), la gestion des événements, le réseau et les web services et la gestion des données afin de disposer d'une bonne vue d'ensemble et de pouvoir développer vos premières applications mobiles. L'environnement de développement est Xcode (dernière version en cours).

Objectifs

Concrètement, cette formation Swift vous permettra de :

- Comprendre l'éco système du développement pour les équipements mobiles d'Apple (vocabulaire, concepts, équipements, Xcode, Apple Store, etc.)
- Connaître le langage Swift et ses principales bibliothèques
- Maîtriser l'architecture d'une application mobile écrite en swift (modèle, vues, contrôleur, web services, etc.)
- Savoir intercepter les événements utilisateurs (gesture)
- Savoir gérer les Web Services de façon synchrone et asynchrone
- Savoir manipuler les données (CoreData, JSON, LIST, etc.)
- Mettre en œuvre la géolocalisation dans votre application
- Comprendre les liens entre Swift et Objective-C

- Comprendre l'apport de SwiftUI par rapport à Interface Builder

À qui s'adresse cette formation ?

Public :

Cette formation sur la programmation iPhone/iPad en langage Swift s'adresse aux ingénieurs logiciels, développeurs, analystes programmeurs, chefs de projets.

Prérequis :

De bonnes connaissances de la programmation et d'un langage orienté objet (Objective-C, C++, Java, C#, PHP5 objet, etc.) sont demandées pour suivre ce stage dans les meilleures conditions possibles.

Contenu du cours Programmation iOS swift

Avant de démarrer son application iOS en Swift

Le marché en chiffres

iOS dernière version, quoi de neuf ?

Les spécificités techniques du matériel

Différences entre les équipements Apple (iPad, iPhone, Apple Watch, Apple TV)

Comprendre le fonctionnement de iOS pour mieux développer (priorité des process, background, consommation électrique, etc.)

Swift vs Objective-C la synthèse

Environnement et outils de développement

Matériel et configuration Mac nécessaire

Rôles de Xcode et Interface Builder

Instruments

L'intégration continue (tests unitaires, etc.)

Apple Developer Program

Documentations et ressources techniques

Atelier :

Guidé par l'animateur, TP avec 1 bouton, 1 Label et 1 TextField pour présenter les différents sujets du chapitre.

Architecture d'une application Swift

Vue d'ensemble des design patterns fondamentaux

Design pattern : Model View Controller (MVC)

Design pattern : Delegate

Design pattern : Protocol

Design pattern : Observer

Structure d'un projet Xcode

Swift, vue d'ensemble

Cohabitation avec Objective-C

Les constantes et variables

Opérateurs, instructions de contrôle, fonctions, boucles

Classes et interfaces

Tuples, Closures, Enumerations

Gestion des erreurs

Conseils de nommage

Atelier :

Mise en œuvre d'un projet classique pour maîtriser son environnement de développement et les bases syntaxiques (erreurs de compilation, debugger, etc.)

Manipulation et stockage des données

Gestion de la mémoire

Ressources du projet (textes, images, sons, vidéos...)

Manipulation de fichiers

Introduction à CoreData (SQLite)

Stockage au format JSON et PLIST

NSKeyedArchiver/ NSKeyedUnarchiver, protocol Codable en Swift

Atelier :

Intégration de JSON et PLIST au sein du projet préalable

Interface utilisateur avec UIKit

Design avec storyboard et segue

La classe UIView

Répondre à des événements avec UIResponder et le mécanisme Target-Action

Interface simple avec UIButton, UIImage, UILabel et UITextField

Message bloquant avec UIAlertController

Enchaînement des écrans avec UITabBar, UINavigationController et UIStoryboardSegue

Liste de données avec UITableView

Animation des UIViews

Expérience Utilisateur enrichie avec UIGestureRecognizer

Atelier :

Mise en œuvre des différents composants graphiques présentés ici sous forme de synthèse dans plusieurs vues afin d'aborder également la navigation et le passage de données.

Géolocalisation

Recommandations générales sur la gestion de la géolocalisation dans les applications mobiles

Présentation du framework CoreLocation

Paramétrer le fichier info.plist pour les autorisations de l'utilisateur

Lancer et stopper la géolocalisation

Utiliser le GeoCoding

Lancer la map à partir des coordonnées géodésiques

Atelier :

Création d'une application utilisant la géolocalisation par réseau et GPS, récupérant latitude et longitude, utilisant le géocoding et demandant les permissions à l'utilisateur

Applications connectées et Web Services en Swift



ITgate

Training

Your Gateway to Excellence

Introduction au réseau et iOS

Push notifications et protocoles disponibles sous iOS

Architecture GCD et asynchronisme (threading)

Flux JSON, le format

Bonnes pratiques pour accéder au réseau

Le non blocage du thread UI (main thread) avec GCD

Mise en oeuvre, les différentes apis utilisables

Eviter les références cycliques dans les closures d'appels

Contraintes imposées par Apple

Gestion des Web Services en JSON

Atelier :

Mise en œuvre de Web Services, consommation de Web Services de façon synchrone et asynchrone. Téléchargement de photos d'un cloud sur internet

Présentation de Swift UI

Grandes différences entre SwiftUI et Interface Builder

Exemple de création de vues

Un petit mot sur la gestion des états et des données partagées

Comment gérer les listes en Swift UI ?

Gestures avec SwiftUI

Atelier :

Ce chapitre n'est qu'une présentation rapide des concepts SwiftUI (ils font l'objet d'une formation dédiée) mais nous vous montrons les grandes lignes d'une interface simple rapidement développée en Swift UI

Travaux Pratiques

Beaucoup de pratique autour du développement d'applications réalistes.