

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

Objectifs 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, l'iPod Touch, ou l'iPad réunis sous la bannière iOS.

Ce cours se déroule exclusivement en langage Swift. Une fois votre environnement de développement et le langage Swift maîtrisés, vous abordez les API du SDK iOS comme la gestion des données et l'interface utilisateur afin de disposer d'une bonne vue d'ensemble et de pouvoir développer vos premières applications. L'environnement de développement est Xcode 12 pour iOS 14.

Les spécificités de l'iPad, des écrans secondaires et des nouveaux matériels Apple seront abordées. Les nouveautés Swift 5 seront également présentées.

PRÉ-REQUIS :

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

PUBLIC :

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

Contenu du cours Programmation iOS Swift

Avant de démarrer son application iOS en Swift

- Le marché en chiffre
- iOS 14, quoi de neuf ?
- Les spécificités techniques du matériel
- Différences entre les équipements Apple (iPad, iPhone, Apple Watch, Apple TV)
- Impacts du firmware sur les applications
- 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

Nouveautés de Xcode 12

- Les changements au quotidien avec Xcode 10 (mode foncé, multi-curseurs, bibliothèque « library », etc.)
- Intégration avec GitHub, Gitlab et Bitbucket
- Compilation et débogage accélérés
- Prises-en charge des nouveautés par rapport aux dernières versions de iOS et MacOS
- Autres nouveautés ...

Architecture d'une application Swift

- Vue d'ensemble des design patterns fondamentaux
- Design pattern: Model View ViewModel (MVVM)
- Design pattern: Model View Controller (MVC)
- Design pattern: Protocol, Delegate
- 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 Structs
- Tuples, Closures, Enumerations
- Gestion des erreurs
- Conseils de nommage

Manipulation et stockage des données

- Gestion de la mémoire
- Ressources du projet (textes, images, sons, vidéos...)
- Manipulation de fichiers
- Introduction à CoreData (SQLite)
- Realm Swift
- Stockage au format JSON et PLIST
- NSKeyedArchiver/ NSKeyedUnarchiver, protocol Codable en Swift

Interface utilisateur avec UIKit

- Design avec storyboard et segue
- La classe UIView
- Répondre à des évènements 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
- Auto Layout

Applications connectées en Swift

- Se connecter à un serveur avec une API REST
- Networking
- Flux JSON
- Téléchargement de contenu
- Problématiques
- Mécanismes proposés par Apple
- Mise en œuvre
- Gestion des applications avec iTunes Connect
- Les éléments à fournir en plus de l'application
- Contraintes imposées par Apple
- Processus de validation des applications
- Mises à jour de l'application
- Gestion des Web Services

Pour aller plus loin

- Présentation du développement pour l'Apple Watch (Interface Builder, WatchKit)
- L'environnement Xcode pour l'Apple Watch
- Applications universelles
- Ecosystème Apple en ligne : iCloud, Game Center, APNS, iTunes
- Des frameworks spécialisés : exemples de la géolocalisation avec CoreLocation et UIMapKit et du son avec AVFoundation
- Google Maps, Google Place
- Migrer une application vers Swift 5 et iOS 14