

Programme de formation iPhone Développement d'applications

Développement d'applications natives pour la plateforme iOS

Durée

5 jours / 35 heures

Description

Les applications mobiles sont la nouvelle façon de consommer de l'internet et des services. iOS est la plateforme reine sur ce secteur. Cette formation pratique vous permet de disposer de bases solides et complètes pour développer des applications mobiles pour iPhone, iPad et iPod Touch, en natif.

Une fois votre environnement de développement et le langage Objective-C maîtrisés vous abordez les API du SDK iPhone 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, il inclut entre autres IOS SDK 6 et les storyboards. Les spécificités de l'iPad, des écrans secondaires et des nouveaux matériels sous iOS seront abordées. Les nouveautés iOS 10 et Xcode 8 pour les développeurs seront également présentées.

Objectifs pédagogiques

- Mettre en oeuvre les fondamentaux nécessaires au développement d'une application à destination de l'iPhone, iPod, iTouch et iPad
- Utilisez les outils de développement iPhone
- Déployer une application sur l'AppStore

Public

Chefs de projet mobiles, Architectes techniques, Développeurs

Pré-requis

Bonnes connaissances de programmation orientée objet Connaissance du C, C++, C# ou Java

Méthodes pédagogiques

50 % théorie / 50 % pratique

Profil intervenant

L'ensemble de nos formations sont animées par des formateurs expérimentés possédant une expérience terrain éprouvée.

Modalités d'évaluation

L'évaluation des acquis se fait tout au long de la session au travers d'ateliers de mise en pratique des notions et concepts abordés pendant la formation.

Programme

Les architectures iPhone / iPod touch / iPad

- Un marché porteur
- La « sécurité » économique du modèle de développement
- Les spécificités du matériel (processeur, mémoire, disque, ...)
- Différences concrètes entre les modèles
- Les besoins en ressources selon l'utilisation
- Impacts du firmware sur les applications
- Contraintes de développement liées à la plateforme iPhone
- Particularités du système d'exploitation iOS
- Pourquoi utiliser Objective-C ?
- Comment créer une application pour iPhone ?
- Comment développer des applications pour iPhone ?

L'environnement de développement

- Configuration Mac nécessaire
- Présentation des outils et IDE disponibles (Xcode incluant Interface Builder, iPhone Simulator)
- Démarrer un nouveau projet avec Xcode
- Compiler à destination de l'iPhone / iPod touch / iPad
- Compiler à destination de l'iPhone / iPod touch (Organizer)
- Pourquoi faut-il signer un binaire (Target) ?
- Les pragma Xcode / compilateurs ObjC
- Concevoir ses propres modèles (templates) de projet
- Le simulateur iPhone/iPad
- Où trouver de l'aide ?

Le langage Objective-C 2.0

- Présentation générale et vocabulaire
- Comparatif langages orientés objet Objective-C, C++ et Java
- Philosophie des messages
- Les méthodes et les classes
- Création de classes et d'instances
- L'importance des conventions de nommage
- Différence entre classe et interface
- Accès et portée des variables d'instances
- Propagation des messages
- La directive @selector
- S'y retrouver dans la documentation officielle
- Objective-C 2.0 runtime référence
- Catégories et extensions
- Les propriétés déclarées (@property)
- L'utilité des protocoles (@protocol)
- Déclarations des interfaces
- Une gestion de la mémoire performante avec ARC (Automatic Reference Counting)

Les API Cocoa Touch du SDK iPhone

- Tour d'horizon des principales API par thème
- Les grandes lignes des méthodes de codage avec Cocoa Classes de base et hiérarchie
- Conventions de nommage et comportements attendus
- Héritage ou bien composition et connaissance ?

La gestion des données avec le Framework Foundation

- Philosophie du framework foundation
- Les classes fondamentales communes à tous les traitements
- Utilité des collections String, Text & Fonts (NSMutableCharacterSet) Dates, Times & Numbers (NSDate)
- Interfacer son application avec les réglages utilisateurs standards système

L'interface utilisateur pour vos applications iOS

- Description de l'interface builder : Storyboards et segues
- Les apports du framework UIKit
- Comprendre l'intérêt du modèle MVC (Model View Controller)
- Les classes Contrôleur de View (Controller View) UIView, sous-classe UIResponder (réponse/gestion événements utilisateurs)
- La hiérarchie des Views et leur géométrie
- Les interactions entre les objets View dans UIKit
- L'architecture de rendu des Views
- Le système de coordonnées des Views Hiérarchies des classes UIResponder / UIWindow / UIView
- Combiner une Barre d'onglets (Tab Bar) et des Contrôleurs de Navigation
- Exemple UIView / UIImageView
- Appliquer une transition à une View dans un bloc d'animation
- Utilisation des UIAlertController
- Utilisation des UITableViews et présentation des UICollectionViews

Compléments

- Méthodes de contact et gestes
- Gestion de la geolocalisation, Mapkit
- Diffuser du contenu (musique, photos, vidéos) en Wifi (Airplay, Apple TV)
- Les nouveautés iOS 10
- Migrer son application vers iOS 10

Professionaliser son application iPhone/iPad

- Problématiques de l'internationalisation
- Désignation des langues (lproj)
- Présentation de la distribution de son application sur l'Appstore
- Les « contraintes » imposées par Apple
- A quoi sert le certificat de distribution ?
- Quels outils pour le versionning de l'application distribuée ?
- Publication sur iTunes AppStore