

Formation iPhone / iPod Touch Développement



Cette formation iPhone intensive 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 maintenant sous la bannière iOS

Objectifs

- Comprendre comment concevoir une application mobile
- Être capable d'utiliser les API du SDK pour interagir avec l'ipod ou l'ipad
- Connaître les spécificités de compilation pour chacun des appareils
- Etre en mesure d'optimiser l'utilisation de la mémoire de l'application développée pour garantir les performances de l'appareil

Public concerné

- Ingénieur logiciel, développeur ou analyste programmeur

Pré requis

- Si vous ne vous estimez pas complètement à l'aise en programmation nous vous conseillons de suivre au préalable la formation "Objective C" (MB300) qui vous permettra de vous concentrer sur la partie purement iPhone du stage présenté ici tout en cons

Une formation de 5 jours

Caractéristiques	Paris
Tarif : 2420 € HT par personne	21/03/2011
Numéro de formateur : 11753687675	02/05/2011
Nombre d'heures : 35	20/06/2011
Référence : MB301	04/07/2011
Contact : Patrick LE GOFF	19/09/2011
Telephone : 01.76.60.66.10	21/11/2011
Email : contact@kaptive.com	12/12/2011

Description des modules

num	Module
1	Les architectures iPhone / iPod touch / iPad
Détails	<ul style="list-style-type: none"> - 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 plate-forme iPhone - Particularités du système d'exploitation iOS - Pourquoi utiliser Objective C ?
2	Le vertige des chiffres liés aux applications
Détails	<ul style="list-style-type: none"> - Configuration Mac nécessaire - Présentation des outils et IDE disponibles (Xcode, 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 XCode - Le simulateur iPhone - Où trouver de l'aide ?
3	Le langage Objective-C 2.0
Détails	<ul style="list-style-type: none"> - Présentation générale et vocabulaire - Comparatif langages orientés objet Objective C, C++ et Java - Philosophie des messages (message d'appel à une méthode) - 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
4	La programmation objet avec Objective C 2.0
Détails	<ul style="list-style-type: none"> - Catégories et extensions - Les propriétés déclarées (@property) - L'utilité des protocoles (@protocol) - Déclarations des interfaces donnant la responsabilité à d'autres pour l'implémentation - Une gestion de la mémoire performante - Comprendre et savoir utiliser le "runtime reference"
5	Les API Cocoa Touch du SDK iPhone
Détails	<ul style="list-style-type: none"> - 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 ?
6	La gestion des données avec le Framework Foundation
Détails	<ul style="list-style-type: none"> - Philosophie du framework foundation - Les classes fondamentales communes à tous les traitements - Utilité des collections

- String, text et fonts (NSStringSet)
- Dates, times et numbers (NSDate)
- Interfacer son application avec les réglages utilisateurs standards système

7 L'interface utilisateur pour vos applications iPhones

- Détails**
- Framework : Cocoa Touch Layer
 - 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
 - Les bases de Core Animation
 - 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

8 Professionnaliser son application

- Détails**
- Problématiques de l'internationalisation
 - Désignation des langues (lproj)
 - Présentation de la distribution de son application sur l'App Store
 - Les "contraintes" imposées par Apple
 - A quoi sert le certificat de distribution ?
 - Quels outils pour le versionning de l'application distribuée ?
 - Les grandes lignes de la publication sur iTunes App Store