

Swift, développer des applications pour iOS - iPhone/iPad

Cours Pratique de 5 jours - 35h
Réf : DAH - Prix 2024 : 2 970€ HT

À l'issue de la formation, l'apprenant sera capable de développer des applications iOS en Swift, le langage de programmation d'Apple, en exploitant les diverses fonctionnalités de l'iPhone et de l'iPad.

OBJECTIFS PÉDAGOGIQUES

À l'issue de la formation l'apprenant sera en mesure de :

- Comprendre l'écosystème du développement pour les équipements mobiles d'Apple
- Utiliser le langage Swift et les différentes API
- Maîtriser l'architecture d'une application mobile écrite en Swift
- Optimiser, construire et déployer une application performante
- Concevoir une interface graphique
- Echanger des données avec un serveur

Les exercices se feront en Swift avec XCode et Interface Builder sur Mac.

LE PROGRAMME

dernière mise à jour : 07/2022

1) Les bases du langage Swift

- Variables, constantes, tuples, chaînes de caractères.
- L'optional et sa gestion, l'unwrapped.
- Opérateurs et structures de contrôle.
- Closures et fonctions.
- Classes, structures, énumérations, propriétés.
- L'héritage et les protocoles.
- L'architecture d'une application mobile écrite en Swift.

Travaux pratiques : Mise en oeuvre des concepts de Swift.

2) L'écosystème du développement pour les équipements mobiles Apple

- XCode, l'interface Builder et les divers outils disponibles.
- Le débogueur.
- Les outils de mesure : instruments.
- Le portail du développeur Apple.
- L'optimisation et le déploiement d'une application performante.

Travaux pratiques : Mise en oeuvre avec Swift et Interface Builder (UI).

3) Concevoir et construire une interface graphique

- Architecture, mémoire, consommation et modes (Background, Foreground).
- AppDelegate : cycle de vie et fonctionnement.
- Interface Builder : storyboards et segues.
- Contrôleur de navigation, TableView Controller, CollectionView Controller.
- TabBar Controller : création et implémentation.

FINANCEMENT

Ce cours fait partie des actions collectives Atlas.

PARTICIPANTS

Développeurs, chefs de projet, architectes.

PRÉREQUIS

Avoir des connaissances pratiques de la programmation et d'un langage orienté objet (Objective-C, Java, C++, C#).

COMPÉTENCES DU FORMATEUR

Les experts qui animent la formation sont des spécialistes des matières abordées. Ils ont été validés par nos équipes pédagogiques tant sur le plan des connaissances métiers que sur celui de la pédagogie, et ce pour chaque cours qu'ils enseignent. Ils ont au minimum cinq à dix années d'expérience dans leur domaine et occupent ou ont occupé des postes à responsabilité en entreprise.

MODALITÉS D'ÉVALUATION

Le formateur évalue la progression pédagogique du participant tout au long de la formation au moyen de QCM, mises en situation, travaux pratiques...

Le participant complète également un test de positionnement en amont et en aval pour valider les compétences acquises.

MOYENS PÉDAGOGIQUES ET TECHNIQUES

- Les moyens pédagogiques et les méthodes d'enseignement utilisés sont principalement : aides audiovisuelles, documentation et support de cours, exercices pratiques d'application et corrigés des exercices pour les stages pratiques, études de cas ou présentation de cas réels pour les séminaires de formation.
- À l'issue de chaque stage ou séminaire, ORSYS fournit aux participants un questionnaire d'évaluation du cours qui est ensuite analysé par nos équipes pédagogiques.
- Une feuille d'émargement par demi-journée de présence est fournie en fin de formation ainsi qu'une attestation de fin de formation si le stagiaire a bien assisté à la totalité de la session.

MODALITÉS ET DÉLAIS D'ACCÈS

L'inscription doit être finalisée 24 heures avant le début de la formation.

ACCESSIBILITÉ AUX PERSONNES HANDICAPÉES

Vous avez un besoin spécifique d'accessibilité ? Contactez Mme FOSSE, référente handicap, à l'adresse suivante psh-accueil@orsys.fr pour étudier au mieux votre demande et sa faisabilité.

- Gestion des vues : UIView, les widgets disponibles.

- Dessiner dans les vues, animations.

Travaux pratiques : Mise en œuvre des concepts sur les segues, TableView Controller (listes).

4) Les gestes

- Gestures dans la vue.

- Gestures dans le simulateur : utilisation et limitation.

- Ajouter des gestes avec l'Interface Builder, par code.

Travaux pratiques : Mise en œuvre de plusieurs gestes dans un contrôleur.

5) Le réseau et les Web Services, échanger des données avec un serveur

- Gestions des notifications (Push).

- Asynchronisme avec Grand Central Dispatch.

- Gestion des appels de Web Services en JSON.

- Sérialisation et désérialisation au format JSON.

- Organisation des données.

- Gestion des préférences, des fichiers dans le bac à sable.

- ORM : création de modèles de données, implémentation.

Travaux pratiques : Appel d'un Web Services en JSON et de ressources sur Internet. Gérer les données avec CoreData.

6) API de géolocalisation et le géocoding

- Géolocalisation par GPS et réseau.

- Architecture de base : CLLocationManager et sécurité.

- Mise en œuvre de la géolocalisation.

- Gestion des erreurs.

Travaux pratiques : Mise en œuvre de la géolocalisation.

7) L'interaction Swift et Objective-C

- Héritage, syntaxes et types de données : comparaison.

- Points particuliers et limitations.

8) Introduction à SwiftUI

- SwiftUI et XCode.

- Créations de vues.

- Gestion des états et des données partagées.

- Gestion des listes et de la navigation.

- Gestures, animations et transitions.

- UIViewControllers et SwiftUI.

Travaux pratiques : Création d'un projet avec vues personnalisées, vues multiples, vues en liste.

LES DATES

CLASSE À DISTANCE

2024 : 10 juin, 09 sept., 18 nov.

PARIS

2024 : 02 sept., 04 nov.

LYON

2024 : 09 sept.

AIX-EN-PROVENCE

2024 : 09 sept.

LILLE

2024 : 09 sept.

NANTES

2024 : 09 sept.

TOULOUSE

2024 : 09 sept.