DAWAN

Plan de la formation

www.dawan.fr

Formation iPhone / iPad avancée, simplifier la persistance des données avec SwiftData

Durée :	3 jours (21 heures)
Tarifs inter-entreprise :	2 475,00 CHF HT (standard) 1 980,00 CHF HT (remisé)
Public :	développeurs, ingénieurs, architectes ou chefs de projet techniques ayant déjà une bonne expérience de développement d'applications iOS avec Swift et SwiftUI, et souhaitant se perfectionner pour utiliser des moyens de persistance de données rapides, modernes et puissants
Pré-requis :	maîtriser Swift et SwiftUI
Objectifs :	Maîtriser SwiftData pour construire rapidement des applications à persistance de données pour iPhone et iPad.
Modalités pédagogiques, techniques et d'encadrement :	• Formation synchrone en présentiel et distanciel. • Méthodologie basée sur l'Active Learning : 75 % de pratique minimum. • Un PC par participant en présentiel, possibilité de mettre à disposition en bureau à distance un PC et l'environnement adéquat. • Un formateur expert.
Modalités d'évaluation :	• Définition des besoins et attentes des apprenants en amont de la formation. • Auto-positionnement à l'entrée et la sortie de la formation. • Suivi continu par les formateurs durant les ateliers pratiques. • Évaluation à chaud de l'adéquation au besoin professionnel des apprenants le dernier jour de formation.
Sanction :	Attestation de fin de formation mentionnant le résultat des acquis
Référence :	DéV102426-F
Note de satisfaction des participants:	4,49 / 5

Contacts :	commercial@dawan.fr - 09 72 37 73 73
Modalités d'accès :	Possibilité de faire un devis en ligne (www.dawan.fr, moncompteformation.gouv.fr, maformation.fr, etc.) ou en appelant au standard.
Délais d'accès :	Variable selon le type de financement.
Accessibilité :	Si vous êtes en situation de handicap, nous sommes en mesure de vous accueillir, n'hésitez pas à nous contacter à referenthandicap@dawan.fr, nous étudierons ensemble vos besoins

Vue d'ensemble de la persistance dans iOS

L'écosystème Swift, SwiftUI, le système de fichiers
Persister des données simples avec @AppStorage et @SceneStorage
Property List, préférences et accès au système de fichiers
Le protocole Codable, transformation en JSON et vice-versa
Introduction à CoreData et SwiftData

Persistance avec CoreData

Anatomie d'un stack CoreData
L'éditeur de modèle et les entités
Le rôle des NSManagedObject subclasses et des extensions
Les relations One to Many et Many to Many
Intégrer CoreData dans SwiftUI avec le NSPersistentContainer
Lire et afficher les données avec le property wrapper @FetchRequest

Atelier :
Sauvegarder des préférences utilisateur simples avec @AppStorage
Archiver des données avec le protocole Codable
Créer une application iOS / CoreData et configurer le stack CoreData
Créer des entités avec leurs attributs et contraintes
Visualiser la base SQLite
Afficher les données dans un écran SwiftUI

Approfondissement CoreData

Implémentation des fonctionnalités d'ajout et suppression
Synchronisation
Tri et filtrage, requêtes dynamiques

Atelier :
Création des vues SwiftUI et des opérations CRUD
Trier, filtrer les éléments avec NSSortDescriptor et NSPredicate
Effectuer des requêtes dynamiques
Monitoring des performances de l’app

SwiftData, les fondamentaux

Le schéma de modèle
Anatomie de la macro @Model
Intégration avec SwiftUI
Les requêtes CRUD avec @Query
Le rôle du protocole PreviewModifier
Les relations One to Many et Many to Many

Atelier :
Création d’une application iOS avec SwiftData
Modélisation des données avec @Model
Visualisation de la base SQLite
Visualisation des données de test avec le protocole PreviewModifier
Création d’interfaces utilisateur pour afficher les données
Implémentation de fonctionnalités comme l’ajout, la modification, et la suppression d’éléments.

SwiftData, approfondissement

Les requêtes dynamiques avec les macros @Query et Predicate
Intégration des données mobiles
Sauvegarder des images
Stratégie pour réaliser une migration simple
Intégrer SwiftData dans une application CoreData existante

Atelier :
Mise en place de la synchronisation des données principales
Effectuer des requêtes dynamiques
Synchroniser les données avec iCloud / CloudKit
Déboggage, performance et qualité de l‘application
Utiliser Swift Testing pour démontrer la qualité du code
Débugger et profiler l’app avec Instruments