

Formation **Swift** : développer des applications iOS

Cette formation vous permettra d'apprendre à développer votre propre application iOS tout en explorant la syntaxe ainsi que les fonctionnalités avancées de Swift et SwiftUI. Vous prendrez en main l'environnement de développement Xcode, verrez comment utiliser les nouveaux assistants d'Intelligence Artificielle Swift Assist, Claude, ChatGPT, et apprendrez à optimiser les performances de votre application.

Durée

4 jours

Objectifs pédagogiques

- ❖ Utiliser le langage Swift et les différentes API
- ❖ Maîtriser l'architecture d'une application mobile iOS écrite en Swift
- ❖ Concevoir des interfaces graphiques avec SwiftUI
- ❖ Echanger des données avec un serveur REST
- ❖ Optimiser et déployer une application iOS performante

Public

Développeurs, chefs de projet techniques, toute personne souhaitant développer des applications iOS avec Swift et SwiftUI.

Prérequis

Bonnes connaissances ou expérience en programmation, si possible POO.

Programme de formation

Phase d'inclusion

Accueil des participants, présentation des objectifs et contextes professionnels de chacun.

Ecosystème Swift, prise en main d'Xcode

L'écosystème de Swift, GitHub et Swift.org
Swift coté serveur, les micro-contrôleurs
Présentation des outils de développement
Xcode, REPL et Playgrounds
Les avantages du framework graphique SwiftUI
Les outils d'assistance artificielle Swift
Assist, Claude, ChatGPT
L'aide, la documentation Xcode, Swift et SwiftUI
Création d'une première application iOS, analyse de sa structure

Les bases du langage Swift

Constantes et variables
Les types de base
Sécurité des types avec les optionnels
Les opérateurs
Les instructions conditionnelles et les boucles
Tuples et collections
Fonctions et closures
Modélisation des données avec structs, enums, classes et actors

Exemple de cas pratiques : playgrounds d'exercices pour les notions abordées.

Swift - aspects avancés

Utiliser les fonctions map, filter et reduce
Les génériques, contraintes et types associés
La programmation par protocole, les extensions
Les bonnes pratiques de gestion d'erreur
Programmation des aspects asynchrones avec async / await, Tasks et TaskGroups

Exemple de cas pratiques : playgrounds d'exercices pour les notions abordées.

Xcode, création d'une première application avec SwiftUI

Les bonnes pratiques de création avec les Human Interface Guidelines
Le processus de maquettage, Keynote et les SF Symbols
Configuration de Xcode, visualisation avec le preview provider et le simulateur
Gérer plusieurs configurations de devices iPhone, iPad, Mac
Utiliser l'Asset Catalog
Les vues avec les stacks HStack, VStack et ZStack
Les modifieurs, les composants de texte

Exemples de cas pratiques :

Développement d'une application SwiftUI fil rouge
Créer des vues et interfaces complexes avec HStack, VStack et ZStack,
Créer un schéma de couleurs dans l'Asset Catalog, utiliser l'accent color
Styler avec les modifieurs, utiliser padding pour aligner les layouts

Gestion des états et des données partagées

Ajouter de l'interactivité avec boutons et contrôles
Gérer l'état d'une vue avec les variables d'état et le property wrapper "@State"
Partager des données entre vues avec "@Binding"
La notion d'état, gestion du flux de données avec property wrappers et bindings
Créer des vues et des composants réutilisables avec l'extraction, les styles.

Exemples de cas pratiques :

Suite du travail sur le projet fil rouge
Création d'un formulaire complexe, gestion de ses données avec « @State »
Ajout de composants, contrôles et boutons
Ajout d'une vue indiquant le montant total du panier

Création d'un binding « @Binding » dans la vue de détail

Affichage des données avec List et ScrollView

Création de listes dynamiques avec le protocole Identifiable
Création de ScrollViews robustes
Affichage des données dans une grille multi-colonnes
Documentation du code, les tests unitaires

Exemples de cas pratiques :

Suite du travail sur le projet fil rouge
Affichage des données dans des listes et scrollviews
Initiation aux tests unitaires
Les techniques de débogage avec LLDB, le View Debugger, Instruments

Navigation, architecture d'une application multi-écrans

Les bonnes pratiques de navigation avec les Human Interface Guidelines
Navigation en silo, navigation hiérarchique
L'affichage modal, les Alertes
L'affichage sur iPad avec SplitView

Application orientées données et multi-écrans

Créer une navigation robuste avec TabView, NavigationStack et NavigationLink
Passage de paramètres
Gestion des flux de données avec le framework Observation

Les objets observables et d'environnement
Bonnes pratiques et optimisation des performances de SwiftUI

Exemples de cas pratiques :

Suite du travail sur le projet fil rouge
Création de la navigation pour les écrans de listes et de détail
Implémentation de l'affichage modal avec ".sheet() »
Création de données fictives au format JSON
Modélisation des données avec @Observable

Les appels réseaux sur une API web

Appels réseaux, échange de données avec URLSession
Les traitements asynchrones avec async / await
Codage des données mobiles JSON avec le protocole Codable
Profilage de l'application avec les jauges et Instruments, efficacité énergétique et optimisation des performances

Exemples de cas pratiques :

Suite et fin du travail sur le projet fil rouge
Création d'une application qui se connecte à une API, décode et affiche les données.

La publication sur l'AppStore

Le compte AppStore Connect
Génération de l'archive
TestFlight et publication sur l'AppStore

Moyens et méthodes pédagogiques

- La formation alterne entre présentations des concepts théoriques et mises en application à travers d'ateliers et exercices pratiques (hors formation de type séminaire).
- Les participants bénéficient des retours d'expérience terrains du formateur ou de la formatrice
- Un support de cours numérique est fourni aux stagiaires

Modalités d'évaluation

- **En amont de la session de formation**, un questionnaire d'auto-positionnement est remis aux participants, afin qu'ils situent leurs connaissances et compétences déjà acquises par rapport au thème de la formation.
- **En cours de formation**, l'évaluation se fait sous forme d'ateliers, exercices et travaux pratiques de validation, de retour d'observation et/ou de partage d'expérience, en cohérence avec les objectifs pédagogiques visés.
- **En fin de session**, le formateur évalue les compétences et connaissances acquises par les apprenants grâce à un questionnaire reprenant les mêmes éléments que l'auto-positionnement, permettant ainsi une analyse détaillée de leur progression.