

SwiftUI, développer des applications pour l'écosystème Apple

Cours Pratique de 3 jours - 21h

Réf : WUI - Prix 2024 : 1 870CHF HT

SwiftUI est le nouveau framework d'Apple destiné à la création des interfaces graphiques pour iPhone, iPad, Mac, Apple Watch, Apple TV et Apple Vision Pro. À l'issue de la formation, vous serez capable de développer des applications pour l'écosystème Apple, et particulièrement sur iPhone et iPad, avec SwiftUI.

OBJECTIFS PÉDAGOGIQUES

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

Comprendre l'intérêt et les particularités de SwiftUI par rapport à UIKit

Connaître des fonctionnalités du langage Swift utiles pour SwiftUI

Créer des interfaces riches, flexibles et polyvalentes

Comprendre le data flow de SwiftUI

Créer une application complète connectée et avec persistance

Architecturer ses vues autour des previews d'XCode

Intégrer SwiftUI dans des projets UIKit existants

LE PROGRAMME

dernière mise à jour : 04/2024

1) Concept de SwiftUI

- Qu'est-ce que SwiftUI et pourquoi un nouveau framework ?
- Les différences fondamentales avec UIKit.
- SwiftUI avec iOS, iPadOS, macOS, watchOS, tvOS et visionOS.

2) Fonctionnalités de Swift

- Les opaque return types.
- Les property wrappers.
- Les function builders.
- L'asynchrone avec async/await.
- Le MainActor.

Travaux pratiques : Mise en pratique des fonctionnalités Swift utiles pour SwiftUI.

3) Composition d'interfaces

- Architecture d'une application SwiftUI.
- Les previews d'XCode.
- Créer des vues.
- Les vues simples : textes, labels, images, boutons, formes...
- Les modificateurs de style.
- Les différents types de layout.
- Les interactions et les gestes.
- Séparer, réutiliser et tester ses vues.

Travaux pratiques : Création et composition de plusieurs vues. Utilisation des previews d'XCode.

PARTICIPANTS

Développeurs et architectes.

PRÉREQUIS

Connaissances sur le langage Swift et l'environnement XCode.

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.

4) Gestion des états

- Le fonctionnement du data flow en SwiftUI.
- Les states et les bindings.
- Les vues à état : textfields, toggles, sliders, pickers...
- Les animations et les transitions.
- La macro observable.
- Les environnements.
- L'injection de dépendances.
- L'AppStorage et le SceneStorage.
- Le cycle de vie d'une vue.

Travaux pratiques : Utilisation des states et des bindings. Mise en pratique des animations. Mise en place de l'injection de dépendances.

5) Listes et navigation

- Les listes dynamiques : listes, grilles, formulaires, lazy...
- La navigation entre les vues.
- Les vues modales et les alertes.
- Les toolbars et les menus.
- Les TabView et SplitView.

Travaux pratiques : Création de listes dynamiques avec une stack de navigation. Création de vues modales.

6) Réseau et persistance

- Faire des appels réseau avec SwiftUI.
- Gestion de l'asynchrone et des erreurs.
- Persistance des données avec SwiftData.

Travaux pratiques : Création d'une application connectée avec persistance en base avec SwiftData.

7) Intégration de SwiftUI

- Intégrer des vues SwiftUI dans un projet UIKit.
- Intégrer des composants UIKit dans un projet SwiftUI.

LES DATES

CLASSE À DISTANCE

2024 : 24 juin, 23 sept., 09 déc.