SpiralTrain
developer courses

iOS Development with SwiftUI

Duration: 4 days Price: € 2450

Open Schedule

Audience Course iOS Development with SwiftUI

The course iOS Development with SwiftUI is intended for developers who want to use the declarative SwiftUI framework to develop apps for iPhone and iPad.

Prerequisites iOS Development with SwiftUI

To participate in the course iOS Development with SwiftUI, prior knowledge of programming in the Swift language is desirable.

Realization Training iOS Development with SwiftUI

The theory is treated on the basis of presentation slides and demos. There is ample opportunity to practice. The course works with the latest version of the iOS SDK and XCode. The course times are from 9.30 to 16.30.

Certification iOS Development with SwiftUI

After successfully completing the course the participants receive a certificate iOS Development with SwiftUI.

Content Course iOS Development with SwiftUI

In the course iOS development with SwiftUI, participants learn to use the programming language Swift and the declarative SwiftUI framework for the development of apps for the iPhone and iPad. The XCode Development Environment is used for developing the iOS apps and the many possibilities of this Integrated Development Environment are discussed.

Swift Review

The course iOS Development with SwiftUI kicks off with a review of key elements of the Swift programming language. This includes the discussion of type inference, classes, structures, guards and closures.

Swift UI Architecture

Next the declarative and data driven SwiftUI syntax and the use of SwiftUI projects in XCode are discussed. Attention is also paid to the SwiftUI App and UI hierarchy, the various SwiftUI views, stacks, frames and also to event handling in Swift UI.

Data Persistence

There are several ways to store data in SwiftUI apps. Scene Storage and App Storage are treated and access to the file system of a device is discussed as well. Attention is also paid to storing data in a key value store and in a relational database such as SQLite. Lifecycle modifiers are also reviewed.

Navigation

Navigating between different screens in a SwiftUI app is also part of the program of the course. This section discusses the use of NavigationViews and NavigationLinks that can be included in List and Dynamic Lists.

Gestures

Gestures in mobile apps relate to interaction with the device through taps, clicks and swipes. The use of gestures in SwiftUI is treated as well as the combination with animations.

SwiftUI Widgets

Widgets are the visual building blocks of the user interface of a Swift UI App. Various widgets such as lists, grids, buttons, switches, tables, date pickers and maps are covered. Attention is also paid to the creation of User Defined widgets with the WidgetKit.

UI Kit Integration

Finally the course discusses how existing iOS apps based on the UI kit architecture can be integrated with the SwiftUI architecture. The role of UIViewControllers and Storyboards is covered here.

SpiralTrain
developer courses

Modules Course iOS Development with SwiftUI

Module 1 : Swift Review	Module 2 : SwiftUI Intro	Module 3 : SwiftUI Architecture
Type Inference Type Casting Data Structures Protocols Guards Classes Structures Optional Types Closures Extensions Property Wrappers Stored Properties Computed Properties	SwiftUI Projects SwiftUI in XCode UIKit and Interface Builder SwiftUI Declarative Syntax SwiftUI is Data Driven SwiftUI versus UIKit Xcode in SwiftUI Mode Preview Canvas and Pinning Multiple Device Configurations App on Simulators App on Physical Devices Build Errors UI Layout Hierarchy	SwiftUI App Hierarchy App and Scenes SwiftUI Views Basic Views Additional Layers Subviews Views as Properties Modifying Views Custom Modifiers Basic Event Handling Custom Container Views ContentView.swift File Assets.xcassets
Module 4 : Stacks and Frames	**Module 5 : Lifecycle Modifiers**	**Module 6 : SwiftUI Data Persistence**
SwiftUI Stacks Spacers Alignment and Padding Container Child Limit Text Line Limits Layout Priority Traditional Stacks Lazy Stacks SwiftUI Frames Frames and Geometry Reader Cross Stack Alignment Container Alignment Alignment Guides	onAppear Modifiers onDisappear Modifiers onChange Modifier ScenePhases onChange Modifier Adding Observable Object Designing ContentView Layout Adding Navigation Environment Objects State Properties State Binding Observable Objects State Objects	Using AppStorage Using SceneStorage @SceneStorage Property Wrapper @AppStorage Property Wrapper Adding Data Store Pathnames in Swift Directories and Files Reading and Writing from a File Key-Value Data Using SQLite Directly Managed Objects Persistent Store Coordinator Retrieving and Modifying Data
Module 7 : Lists and Navigation	**Module 8 : SwiftUI Grids**	**Module 9 : Gestures and Animation**
SwiftUI Lists SwiftUI Dynamic Lists NavigationView NavigationLink Editable List Hierarchical Lists Loading JSON Data Using OutlineGroup Using DisclosureGroup Sidebar List Style	SwiftUI Grids LazyVGrid LazyHGri GridItems Flexible GridItems Scrolling Support Adaptive GridItems Fixed GridItems Hierarchical Data Disclosures	Basic Gestures onChange Action Callback Updating Callback Action Composing Gestures Implicit and Explicit Animation Repeating an Animation Explicit Animation Animation and State Bindings SwiftUI Transitions Asymmetrical Transitions
Module 10 : Widgets with SwiftUI	**Module 11 : Integrating UIKit**	**Module 12 : UIViews and UIViewControllers**
Overview of Widgets WidgetKit The Widget Extension Widget Configuration Types Widget Entry View Widget Timeline Entries Widget Timeline Widget Provider Reload Policy Forcing Timeline Reload Widget Placeholders	SwiftUI and UIKit Integration Integrating UIViews into SwiftUI Adding a Coordinator Handling UIKit Delegation Handling UIKit Data Sources Wrapping the UIScrolledView Implementing the Coordinator Using MyScrollView Adding a Hosting Controller Embedding a Container View Testing the App	UIViewControllers and SwiftUI Wrapping UIImagePickerController Designing the Content View Completing MyImagePicker Completing the Content View Preparing the Storyboard Configuring the Segue Action Overview of the Hosting Controller UIHostingController Project Adding the SwiftUI Content View Embedding SwiftUI in Code

SpiralTrain BV
Standerdmolen 10, 2e verdieping
3995 AA Houten

info@spiraltrain.nl
www.spiraltrain.nl
Tel.: +31 (0) 30 – 737 0661

Locations
Houten, Amsterdam, Rotterdam, Eindhoven, Zwolle, Online