



SwiftUI

———— Better apps. Less code ————

Nikita Barhate, iOS Developer
Mukesh Lokare, iOS Developer

What is SwiftUI?

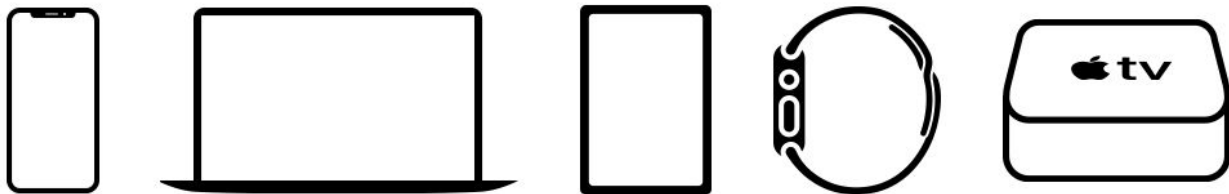
- **New Idea to make apps better and faster.**
- **Declarative UI framework.**
- **Keep your code and design perfectly in sync.**
- **Automatic support for Dynamic Type, Dark Mode, localization, and accessibility.**

Design Tools

- Xcode 11 and Catalina macOS
- Building interfaces with SwiftUI as easy as dragging and dropping.
- Preview UI
- [SwiftUI](#) requires a deployment target of iOS 13 or later, macOS 10.15 or later, tvOS 13 or later, or watchOS 6 or later. The framework contains many new types that don't exist on older versions of the OSs.

Native on All Apple Platform

SwiftUI was built on decades of experience in creating the most innovative and intuitive user interfaces in the world. Everything users love about Apple ecosystems, such as controls and platform-specific experiences, is beautifully presented in your code. SwiftUI is truly native, so your apps directly access the proven technologies of each platform with a small amount of code and an interactive design canvas.



Declarative Syntax

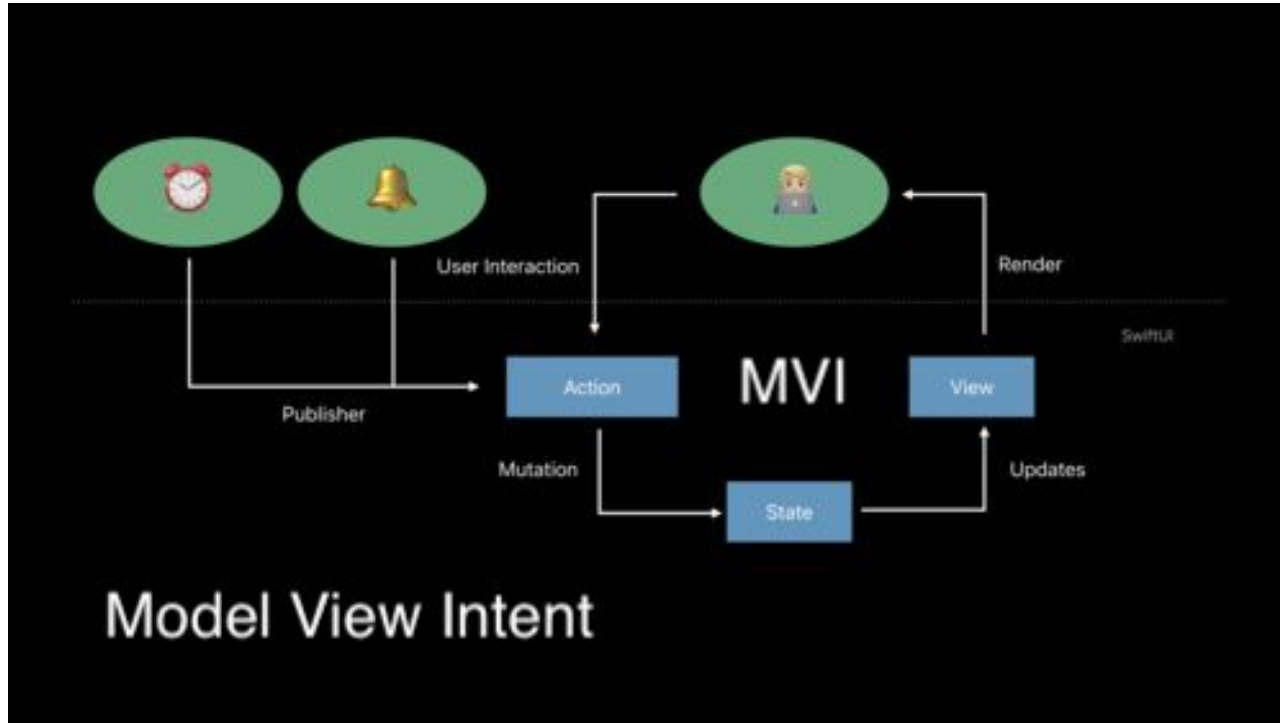
```
import SwiftUI

struct Content : View {

    @State var model = Themes.listModel

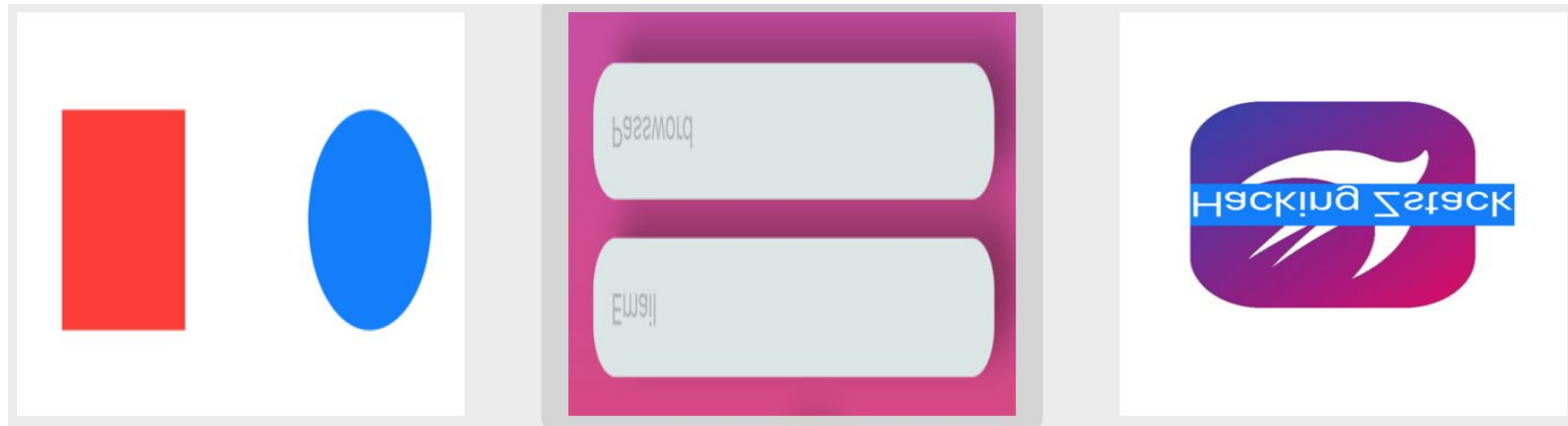
    var body: some View {
        List(model.items, action: model.selectItem) { item in
            Image(item.image)
            VStack(alignment: .leading) {
                Text(item.title)
                Text(item.subtitle)
                    .color(.gray)
            }
        }
    }
}
```

SwiftUI Data Flow



Stacks

- HStack
- VStack
- ZStack



Views and Controls

- **Text**
- **Image**
- **Rectangle**
- **Stepper**
- **Picker**
- **Slider**

Navigation View and Navigation Link

- Navigation views present new screens using **NavigationLink**, which can be triggered by the user tapping their contents or by using programmatically enabling them.
- **Passing data between views**
 - When you use `NavigationLink` to push a new view onto your navigation stack, you can pass any parameters that new view needs to work.
Eg. `NavigationLink(destination: ResultView(choice: "Your choice"))`
 - `@EnvironmentObject`

State Variables

- State is a value, or a set of values, that can change over time, and that affects a view's behavior, content, or layout.
- Use a property with the **@State** attribute to add state to a view.
- A State instance isn't the value itself; it's a means of reading and mutating the value.

Dark Mode

To preview your layout in Dark Mode, you need to add a modifier to the Content View in the Debug area. Please note that to make your design work well with Dark mode, you should use their default colors and controls like the Navigation View as much as possible.

**E.g. ContentView(courses: testData)
 .environment(\.colorScheme, .dark)**

Swift Package Manager

- Third-party, open-source code with ease
- Split your code into reusable, logical chunks
- Demo

SwiftUI Pros and Cons

Pros

- Easy to learn, simple and clean code.
- It can be mixed with UIKit using UIHostingController.
- Allows you to easily manage themes.
- SwiftUI provides mechanisms for reactive programming with BindableObject, ObjectBinding, and the whole Combine framework.
- Offers Live Preview
- SwiftUI no longer needs Interface Builder.
- Less number of crashes

SwiftUI Pros and Cons

Cons

- It supports only iOS 13 and Xcode 11.
- Less references
- Doesn't allow you to examine the view hierarchy

Swift UI with Examples

Mukesh will continue..