



SwiftUI

Overview & Introduction

By:
Andres Parejo
iOS Developer

What's new in SwiftUI

- Advanced app experiences and tools
- Accessibility improvements
- SwiftUI improvements on macOS
- Always-On Retina Display support
- Widgets for iPadOS
- The magic word is Multi-platform



<https://developer.apple.com/xcode/swiftui/>

Declarative syntax

- SwiftUI uses a declarative syntax, so you can simply state what your user interface should do. For example, you can write that you want a list of items consisting of text fields, then describe alignment, font, and color for each field. Your code is simpler and easier to read than ever before, saving you time and maintenance.

```
import SwiftUI

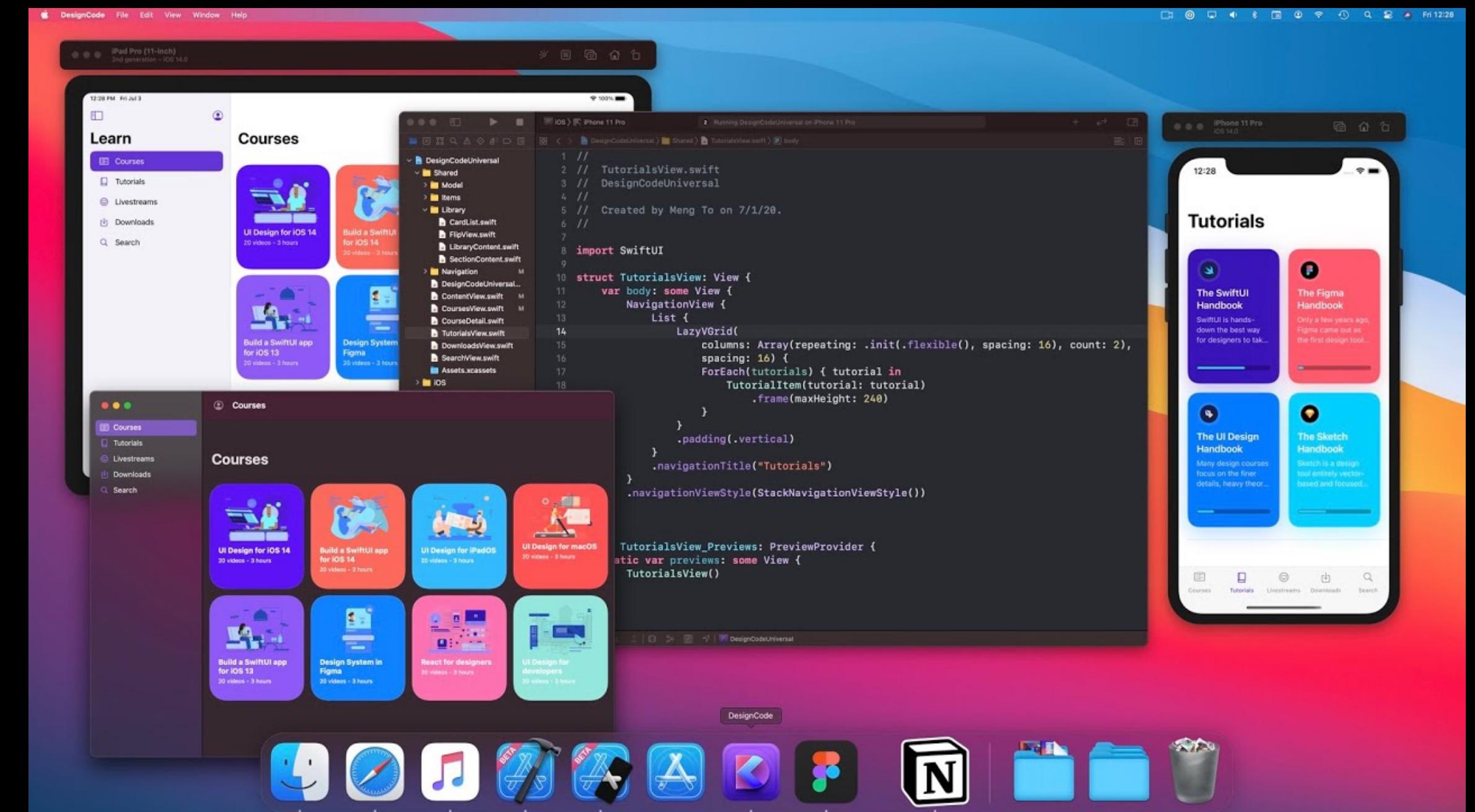
struct Content : View {

    @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)
            }
        }
    }
}
```

Personal overview

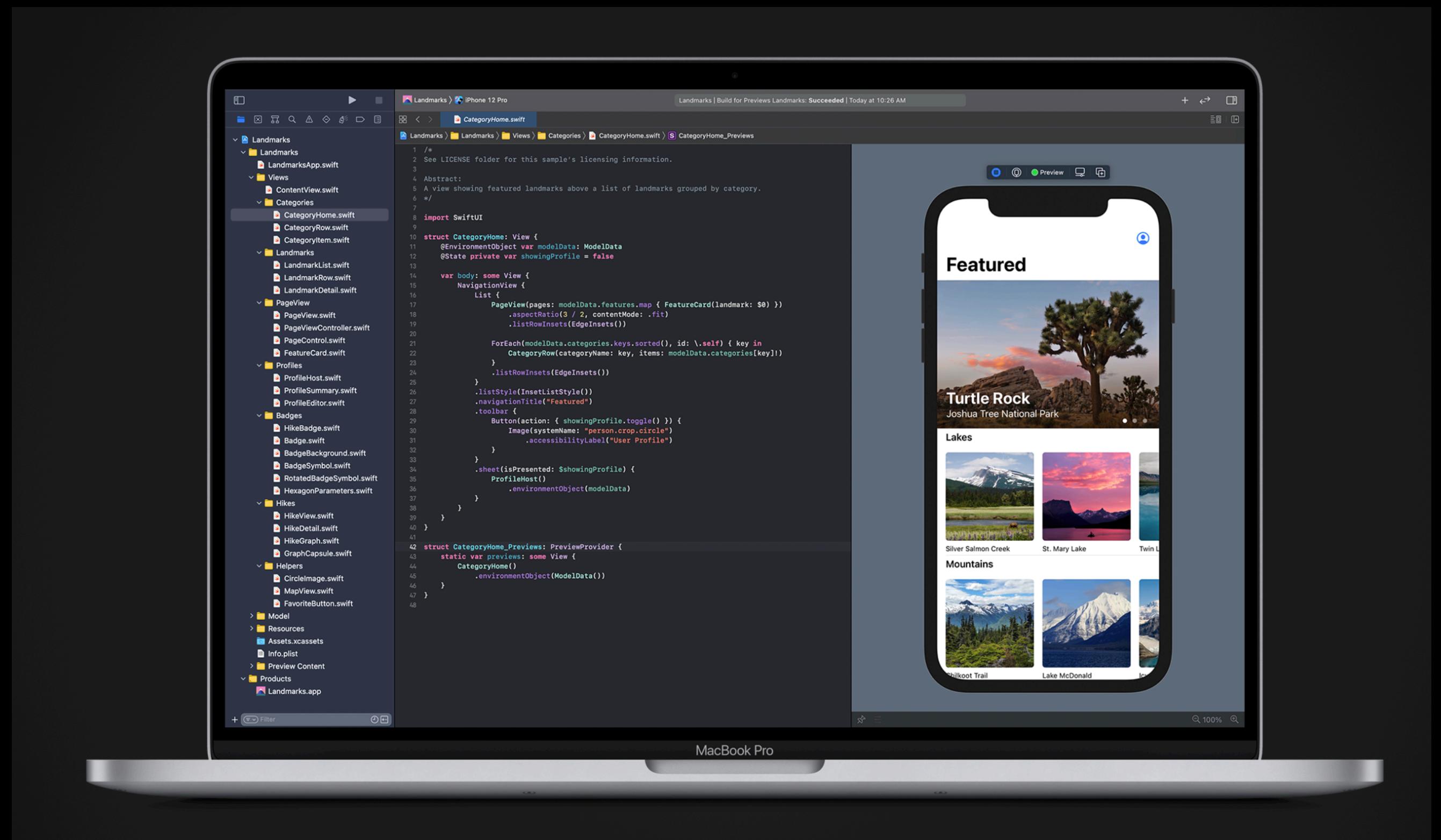
- Widgets or block compilation programming language
- Decrease building time 80%
- SwiftUI preview is a beautiful tool
- When you need a customize implementation you can link UIKit elements with SwiftUI
- VStacks and Spacer() are your best friends



DesignCode, Jul 4, 2020

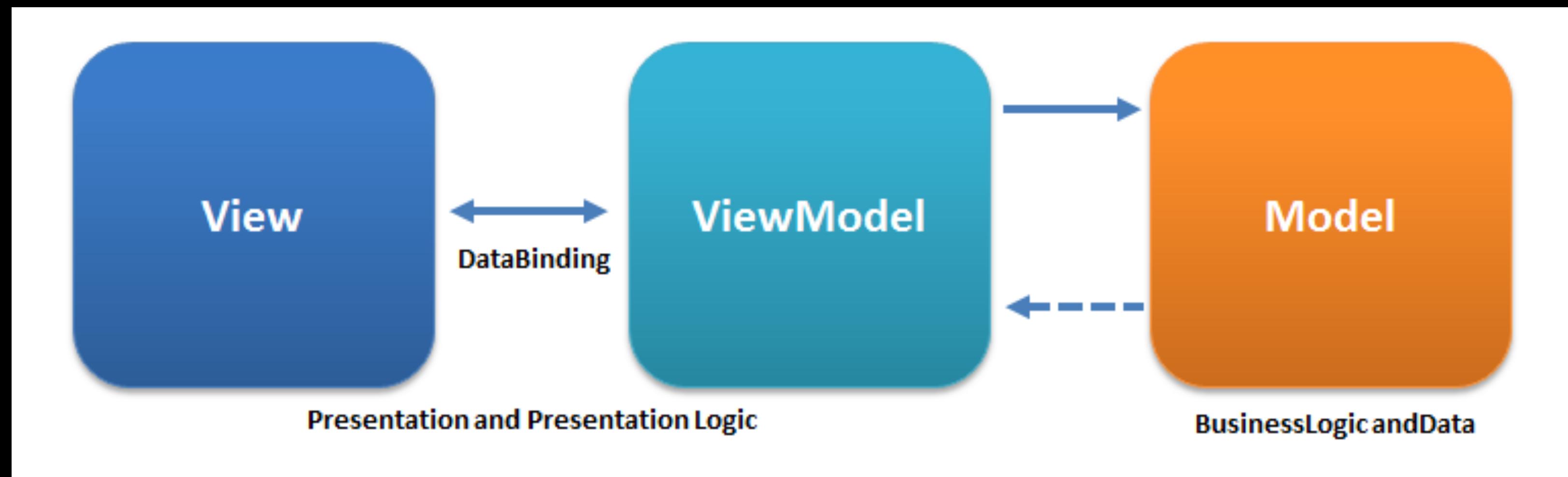
Documentation

- <https://developer.apple.com/documentation/swiftui/>
- <https://developer.apple.com/tutorials/swiftui>
- Literally search “<Input view here> in swiftUI” in Google and will appear a lot of possible ways to do what we want.



<https://developer.apple.com/xcode/swiftui/>

SwiftUI ❤️ MVVM



adictosaltrabajo.com/2020/06/05/patron-mvvm-en-swiftui/

Data Binding

- Without **@State** we wouldn't be able to change properties in our structs, because structs are fixed values.
- Without **StateObject** we wouldn't be able to create classes that stay alive for the duration of our app.
- Without **@EnvironmentObject** we wouldn't be able to receive shared data from elsewhere in our app.
- Without **ObservableObject** we wouldn't be notified when an external value changes.
- Without **\$property** two-way bindings we would need to update values by hand.

DEMO

<https://github.com/andresparejo/GlobantUniversitySwiftUISample/tree/main>

Thanks