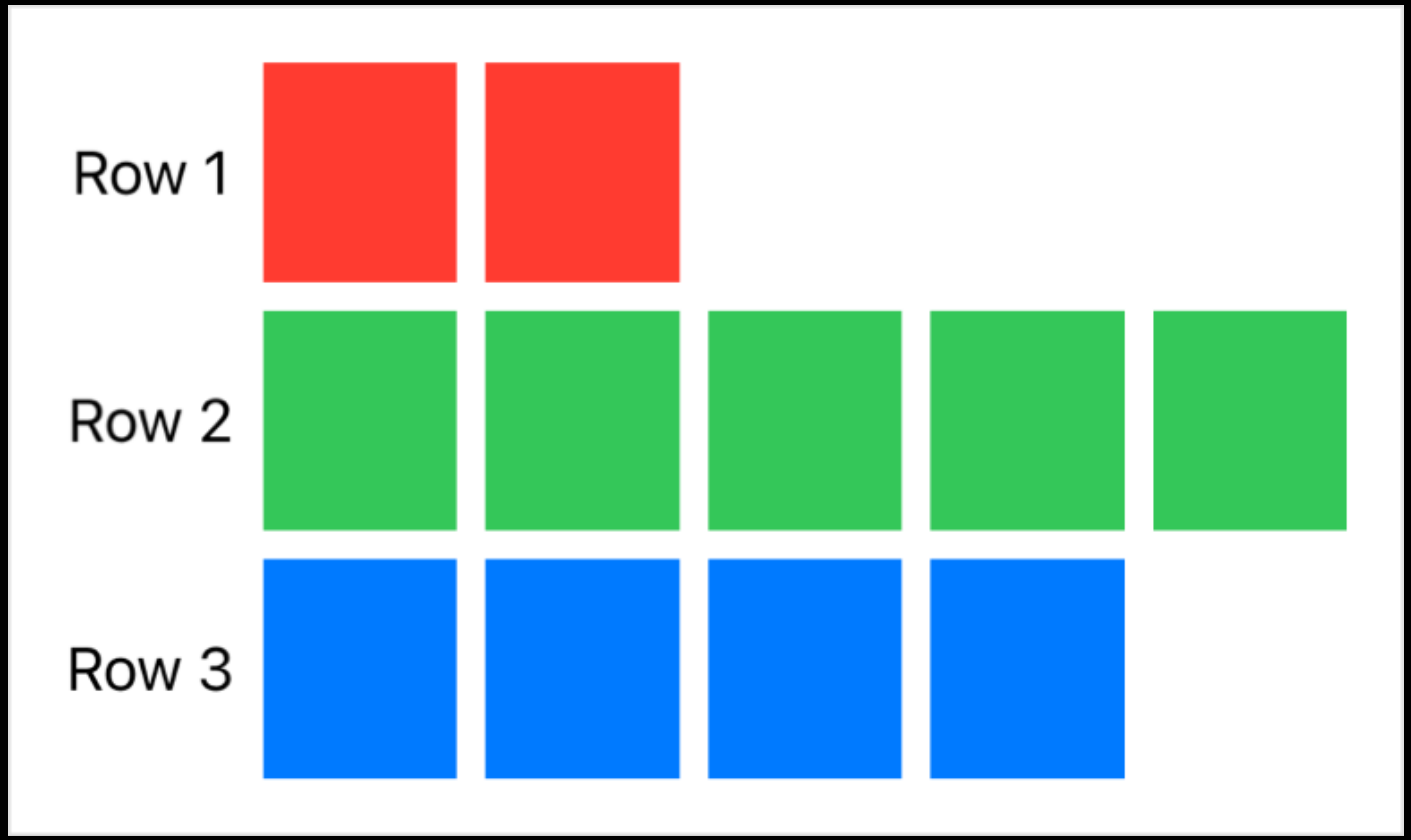
Week4

Grid, TabView, sheet

유가은

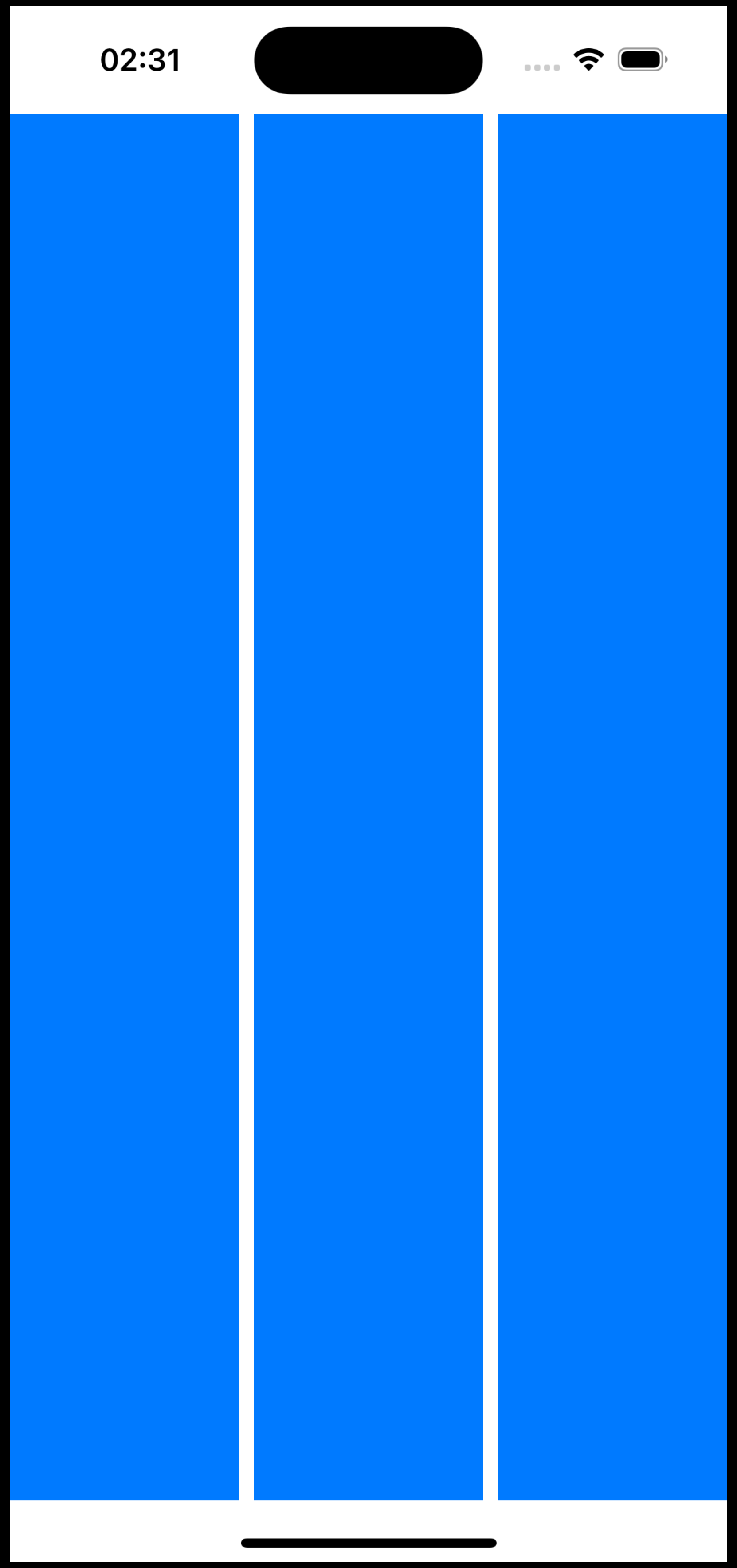
Grid

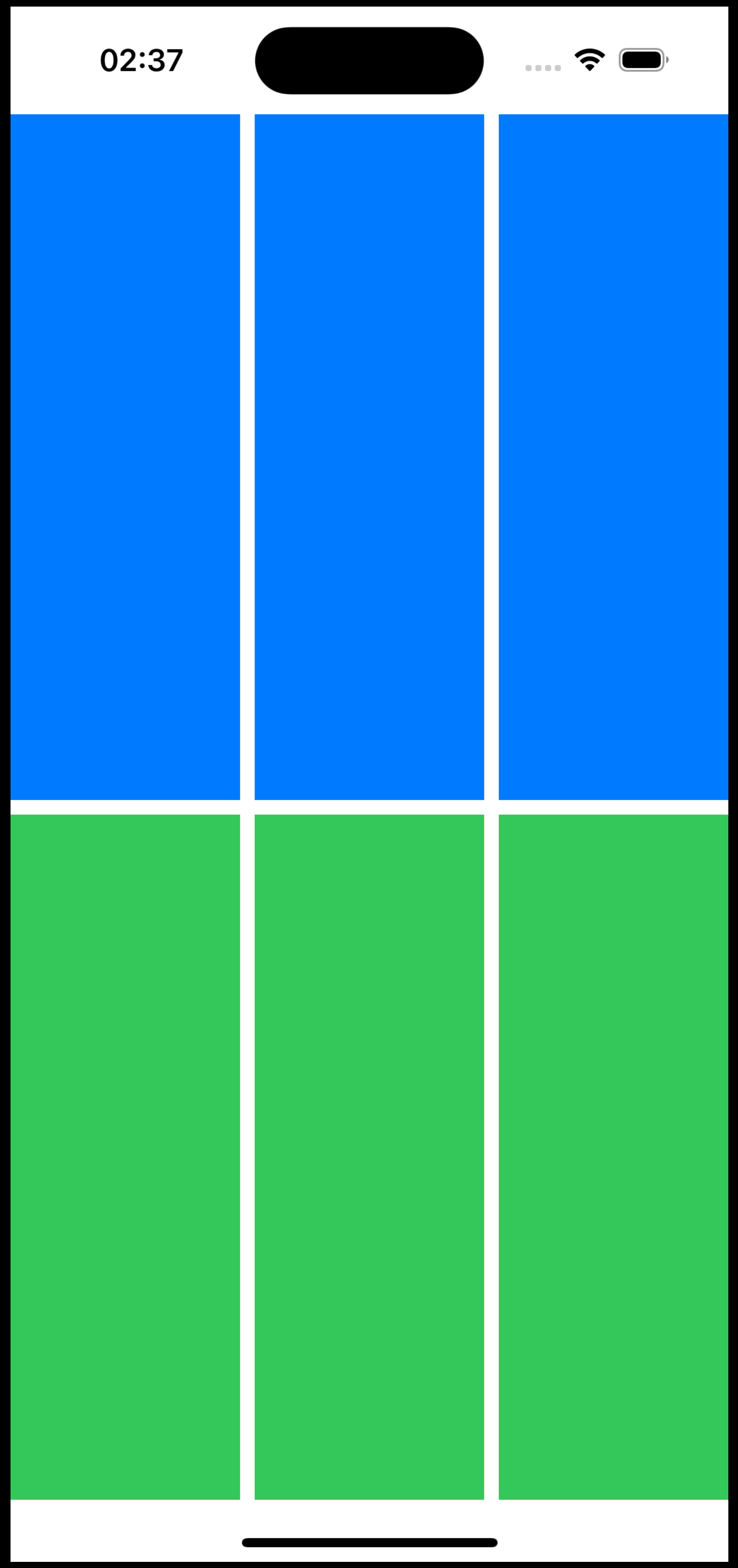
A container view that arranges other views in a two dimensional layout.



Grid

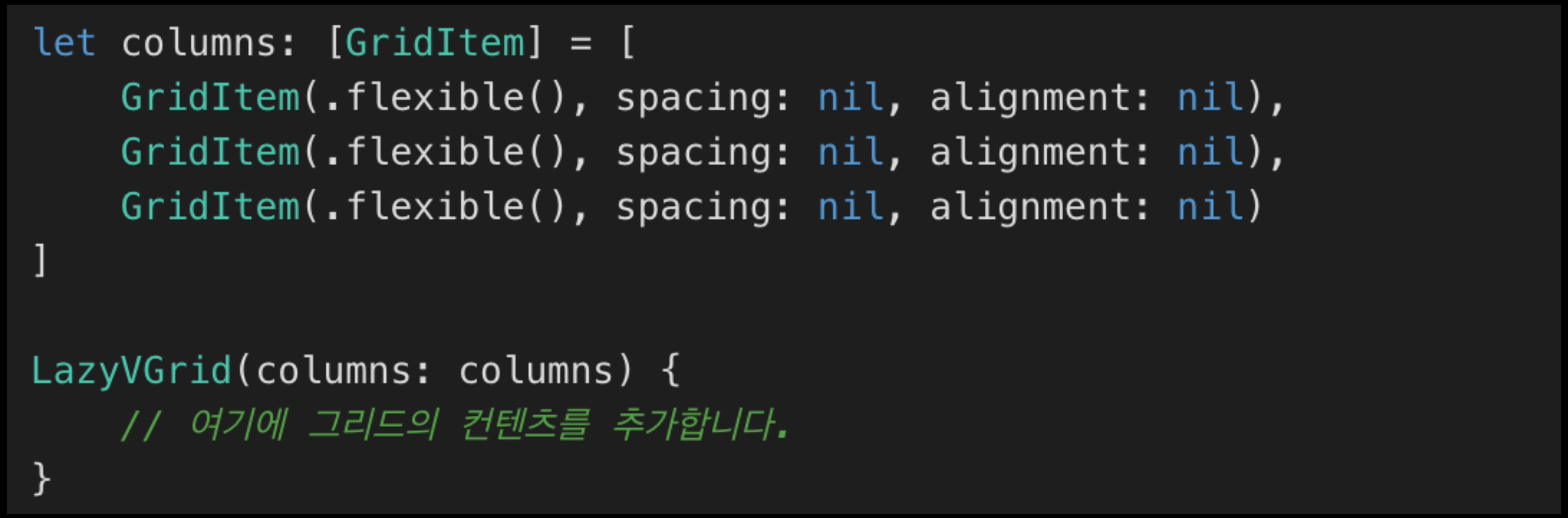
```
Grid {
  GridRow {
     ForEach(0..<3) { _ in
       Color.blue
     }
  }
}
```





LazyVGrid, LazyHGrid

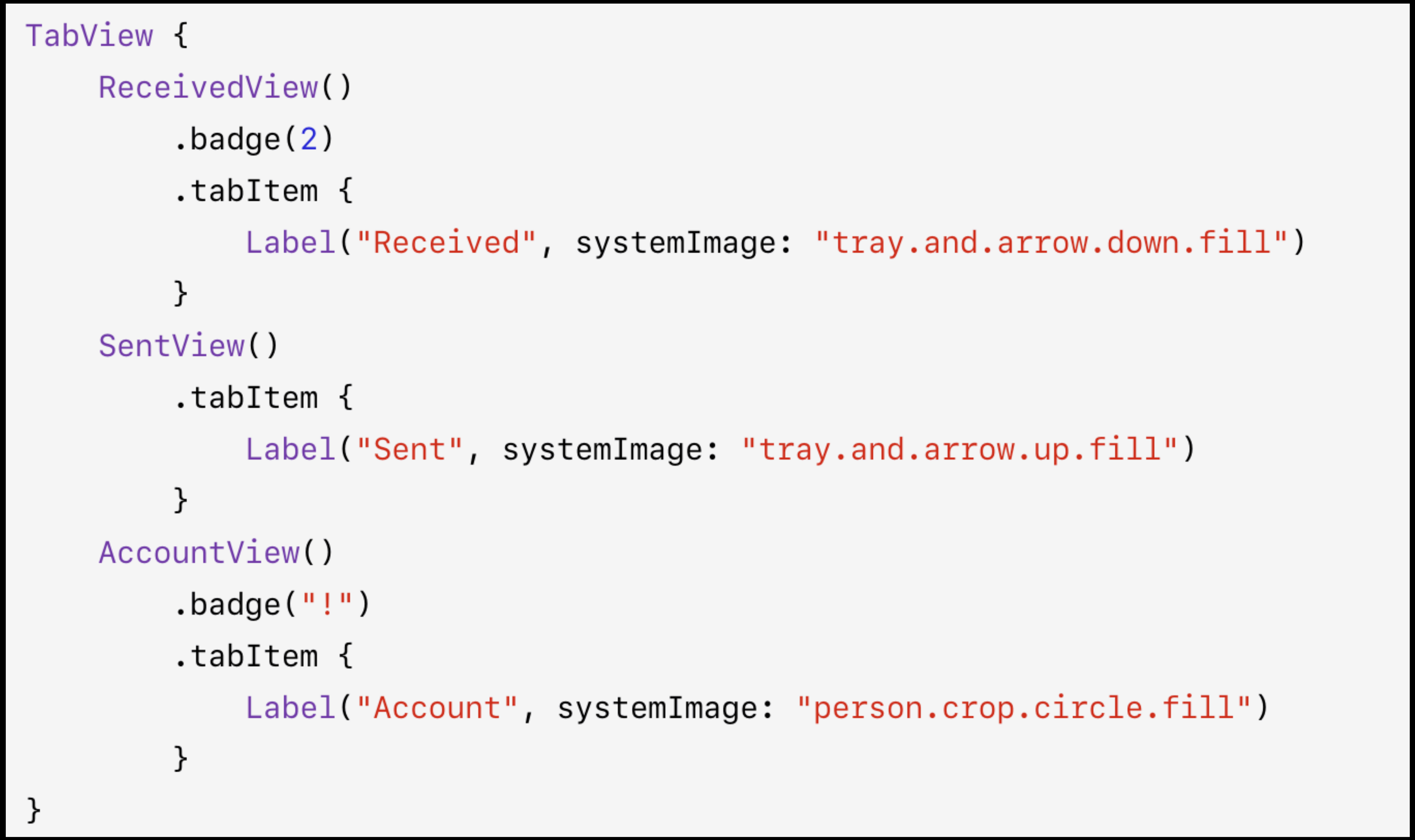
- LazyVGrid(columns:)
- LazyHGrid(rows:)



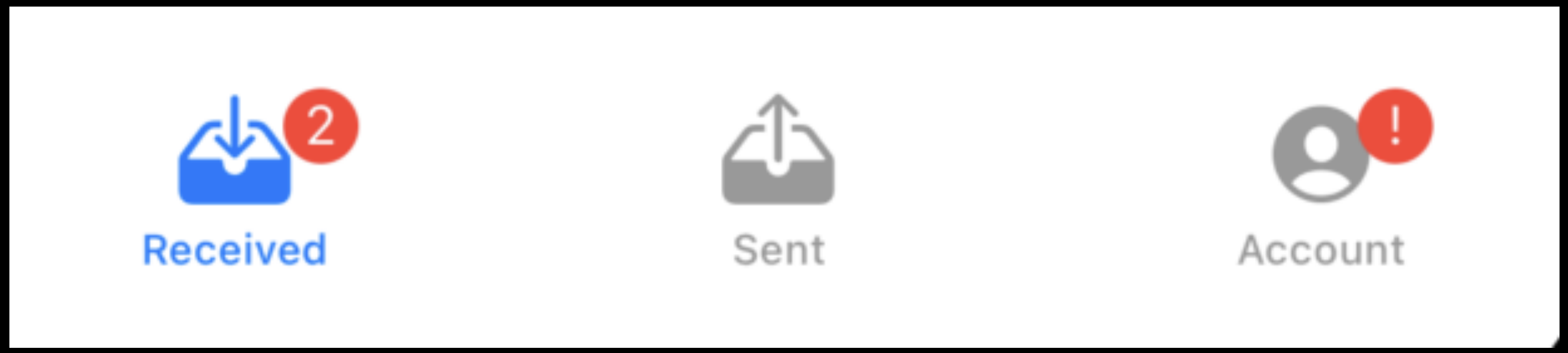
```
let columns: [GridItem] = [
    GridItem(.flexible(), spacing: nil, alignment: nil),
    GridItem(.flexible(), spacing: nil, alignment: nil),
    GridItem(.flexible(), spacing: nil, alignment: nil)
]

LazyVGrid(columns: columns) {
    // 여기에 그리드의 컨텐츠를 추가합니다.
}
```

TabView

A view that switches between multiple child views using interactive user interface elements.



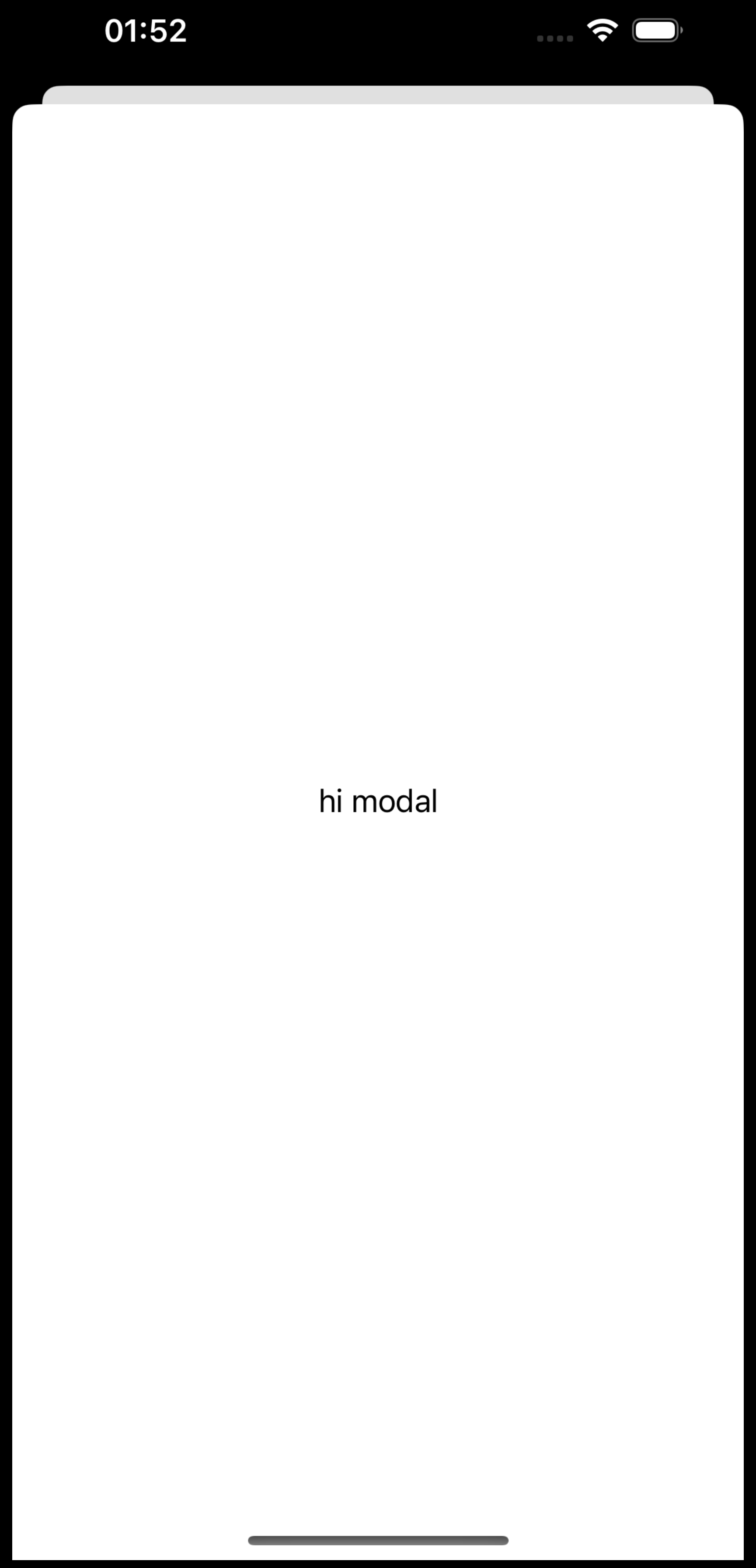
```swift
TabView {
    ReceivedView()
        .badge(2)
        .tabItem {
            Label("Received", systemImage: "tray.and.arrow.down.fill")
        }
    SentView()
        .tabItem {
            Label("Sent", systemImage: "tray.and.arrow.up.fill")
        }
    AccountView()
        .badge("!")
        .tabItem {
            Label("Account", systemImage: "person.crop.circle.fill")
        }
}
```



sheet

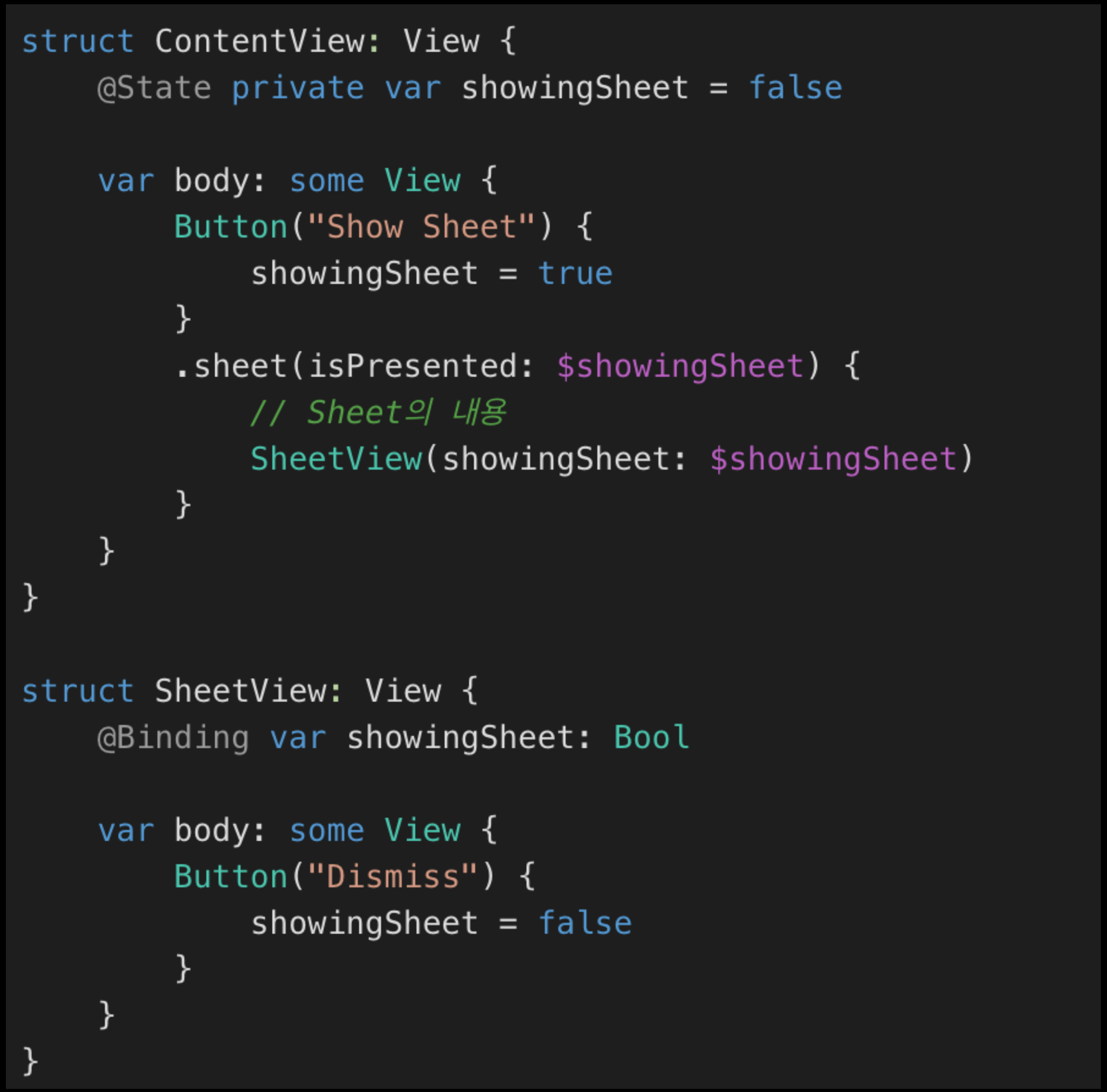
Prefer a sheet appearance when adapting for size classes.

```
Button {
    showModal = true
} label: {
    Text("modal button")
}
.sheet(isPresented: $showModal, content: {
    Text("hi modal")
})
```

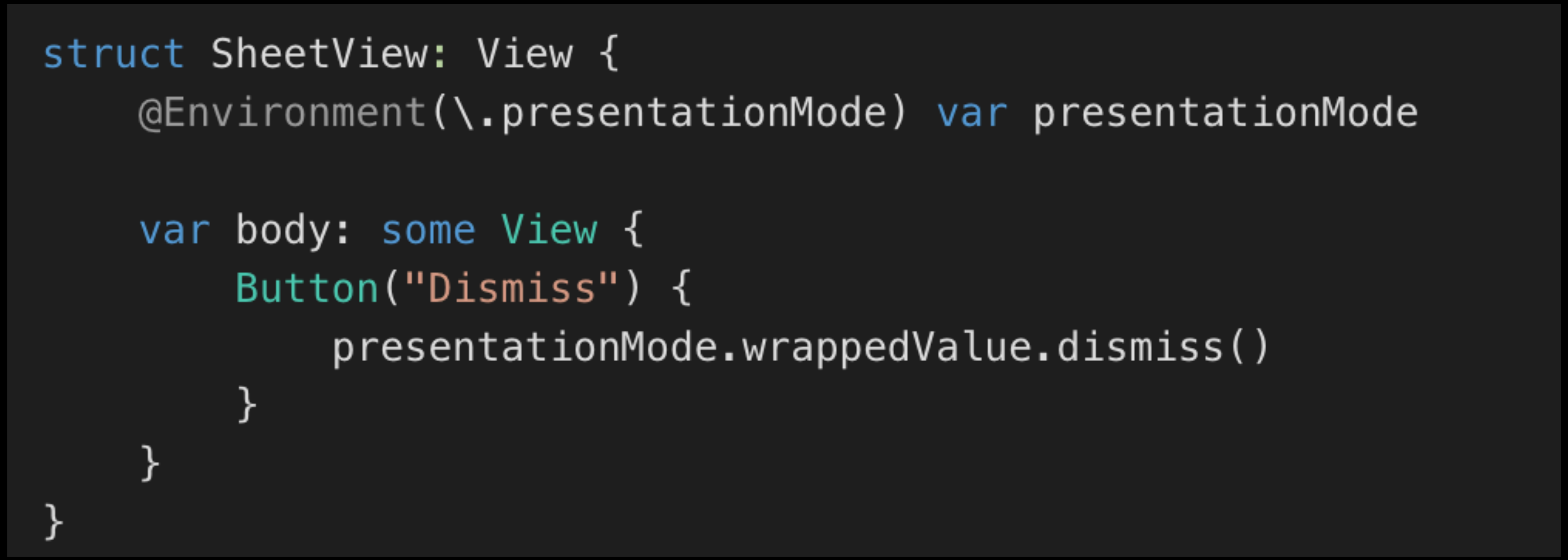


hi modal

Dismiss sheet



```swift
struct ContentView: View {
    @State private var showingSheet = false

    var body: some View {
        Button("Show Sheet") {
            showingSheet = true
        }
        .sheet(isPresented: $showingSheet) {
            // Sheet의 내용
            SheetView(showingSheet: $showingSheet)
        }
    }
}

struct SheetView: View {
    @Binding var showingSheet: Bool

    var body: some View {
        Button("Dismiss") {
            showingSheet = false
        }
    }
}
```



```swift
struct SheetView: View {
    @Environment(\.presentationMode) var presentationMode

    var body: some View {
        Button("Dismiss") {
            presentationMode.wrappedValue.dismiss()
        }
    }
}
```

Resize sheet

- .presentDetents([ ])

```
.medium         // 화면의 절반
.fraction(0.6)  // 비율
.height(500)    // 높이
```

- .presentationDragIndicator(.visible)
- .interactiveDismissDisabled()