

Composants SwiftUI

Composants SwiftUI

Layout

Listes et formulaires

Layout

HStack

```
struct HStackView: View {  
    var body: some View {  
        HStack {  
            Text("Hi")  
            Text("SwiftUI")  
        }  
    }  
}
```

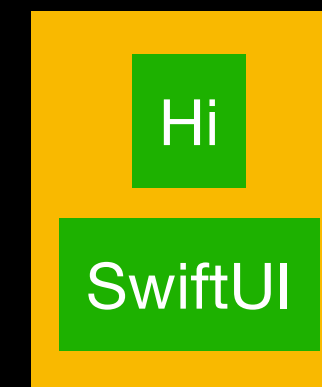
HStack

```
struct HStackView: View {  
    var body: some View {  
        HStack {  
            Text("Hi")  
            Text("SwiftUI")  
        }  
    }  
}
```



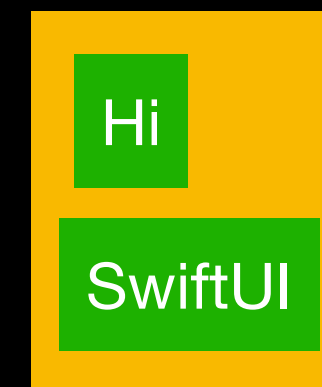
VStack

```
struct VStackView: View {  
    var body: some View {  
        VStack {  
            Text("Hi")  
            Text("SwiftUI")  
        }  
    }  
}
```



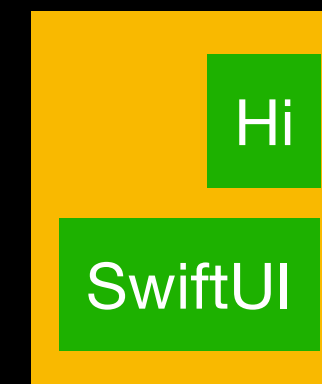
VStack

```
struct VStackView: View {  
    var body: some View {  
        VStack(alignment: .leading) {  
            Text("Hi")  
            Text("SwiftUI")  
        }  
    }  
}
```



VStack

```
struct VStackView: View {  
    var body: some View {  
        VStack(alignment: .trailing) {  
            Text("Hi")  
            Text("SwiftUI")  
        }  
    }  
}
```



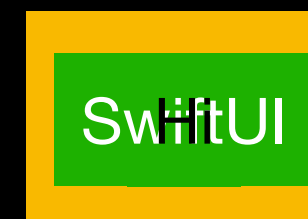
ZStack

```
struct ZStackView: View {  
    var body: some View {  
        ZStack {  
            Text("Hi")  
            Text("SwiftUI")  
        }  
    }  
}
```

The SwiftUI logo, which consists of a yellow square with a green square inside it, and the text "SwiftUI" in white.

ZStack

```
struct ZStackView: View {  
    var body: some View {  
        ZStack {  
            Text("Hi")  
            Text("SwiftUI")  
        }  
    }  
}
```



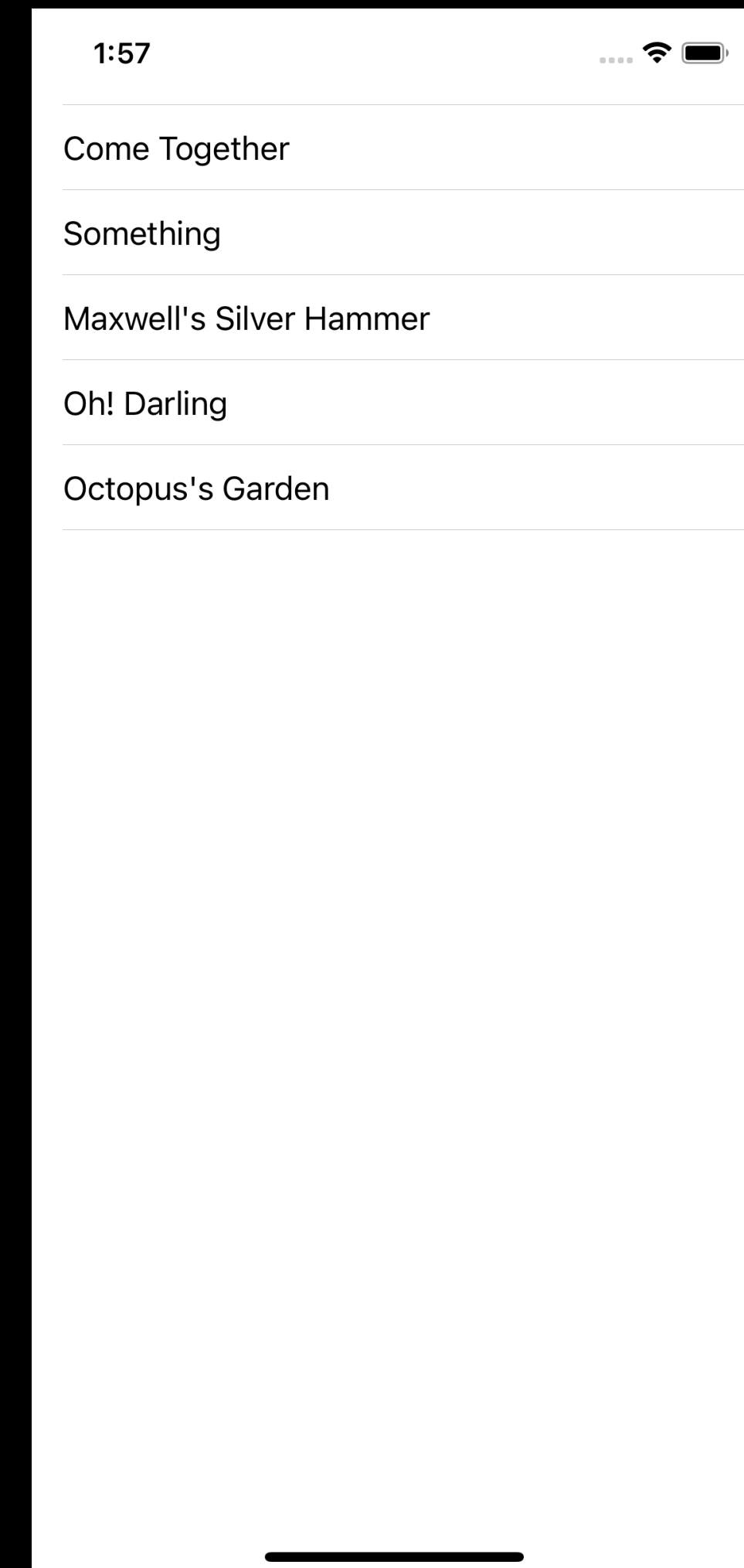
Listes et formulaires

Liste statique

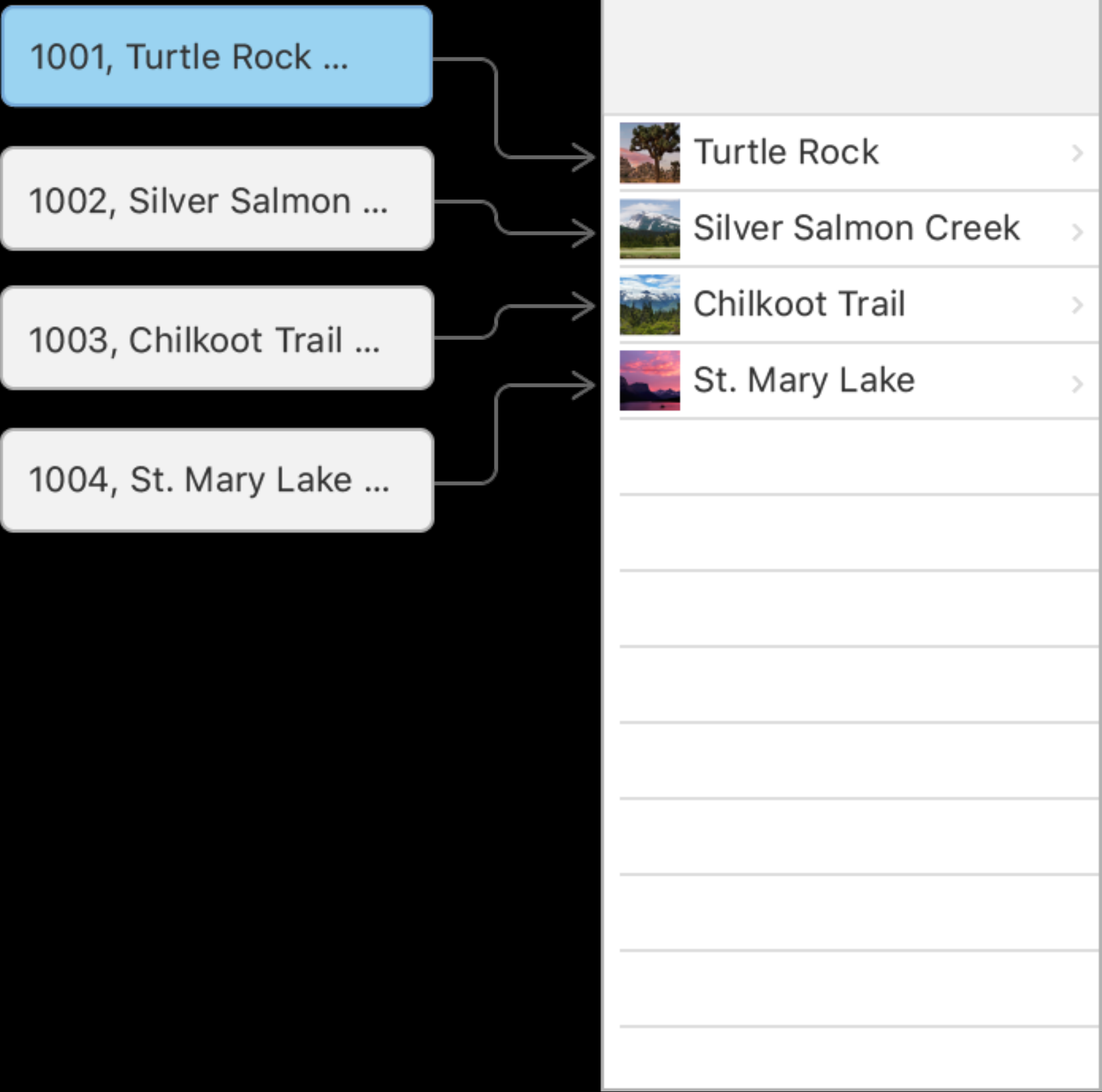
```
struct Lists: View {  
    var body: some View {  
        List {  
            Text("Come Together")  
            Text("Something")  
            Text("Maxwell's Silver Hammer")  
            Text("Oh! Darling")  
            Text("Octopus's Garden")  
        }  
    }  
}
```

Liste statique

```
struct Lists: View {  
    var body: some View {  
        List {  
            Text("Come Together")  
            Text("Something")  
            Text("Maxwell's Silver Hammer")  
            Text("Oh! Darling")  
            Text("Octopus's Garden")  
        }  
    }  
}
```



Listes dynamiques



Listes dynamiques

```
struct Album: Identifiable {  
    var id = UUID()  
    var name: String  
}
```

```
struct AlbumRow: View {  
    var album: Album  
  
    var body: some View {  
        Text(album.name)  
    }  
}
```

Listes dynamiques

```
struct ContentView: View {  
    let albums = [  
        Album(name: "1999"),  
        Album(name: "Desire"),  
        Album(name: "Hello Sunshine")  
    ]  
  
    var body: some View {  
        List(albums) { album in  
            AlbumRow(album: album)  
        }  
    }  
}
```


Listes dynamiques

```
struct ContentView: View {  
    private var albums = ["Emancipation",  
                           "Hotel California",  
                           "3121"]  
  
    var body: some View {  
        List {  
            ForEach(albums, id: \.self) { album in  
                Text(album)  
            }  
        }  
    }  
}
```

List & Form

```
struct ContentView: View {  
    ...  
  
    var body: some View {  
        List {  
            Picker(selection: $selectedColor,  
                    label: Text("Select a color")) {  
                ForEach(0 ..< colors.count) {  
                    Text(self.colors[$0])  
                }  
            }.pickerStyle(SegmentedPickerStyle())  
  
            Toggle(isOn: $enableLogging) {  
                Text("Enable Logging")  
            }  
  
            Button("Save changes") {  
                // activate theme!  
            }  
        }  
    }  
}
```

Settings

Red

Green

Blue

Enable Logging



Save changes

List & Form

```
struct ContentView: View {  
    ...  
  
    var body: some View {  
        Form {  
            Picker(selection: $selectedColor,  
                    label: Text("Select a color")) {  
                ForEach(0 ..< colors.count) {  
                    Text(self.colors[$0])  
                }  
            }.pickerStyle(SegmentedPickerStyle())  
  
            Toggle(isOn: $enableLogging) {  
                Text("Enable Logging")  
            }  
  
            Button("Save changes") {  
                // activate theme!  
            }  
        }  
    }  
}
```

