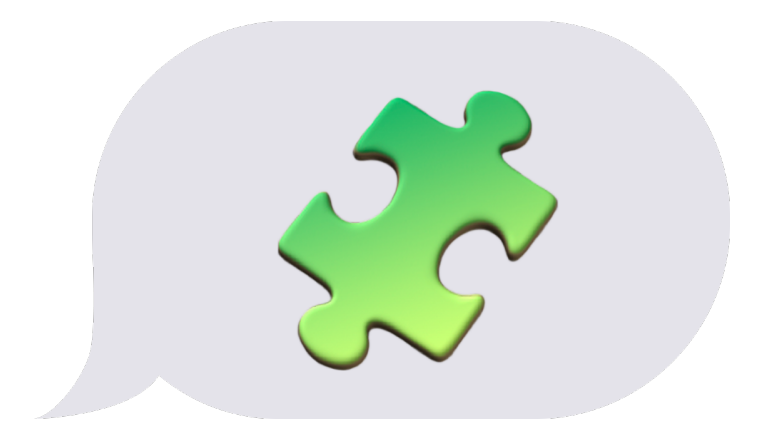
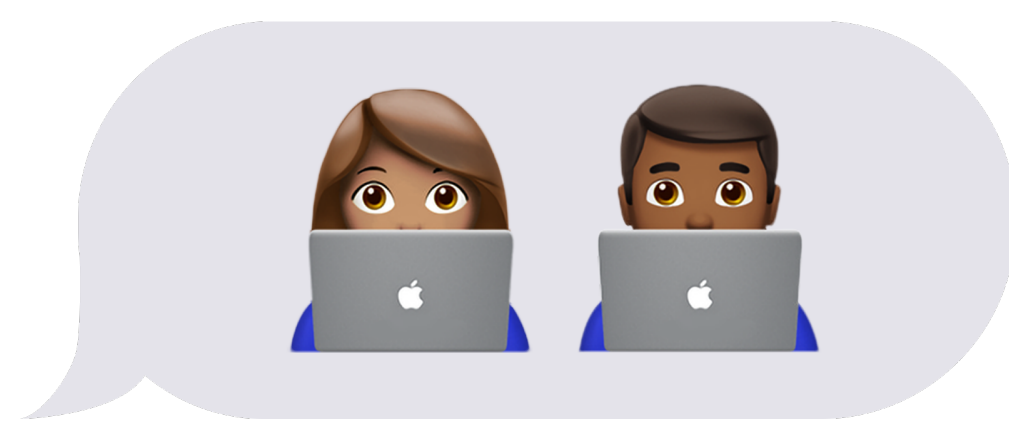
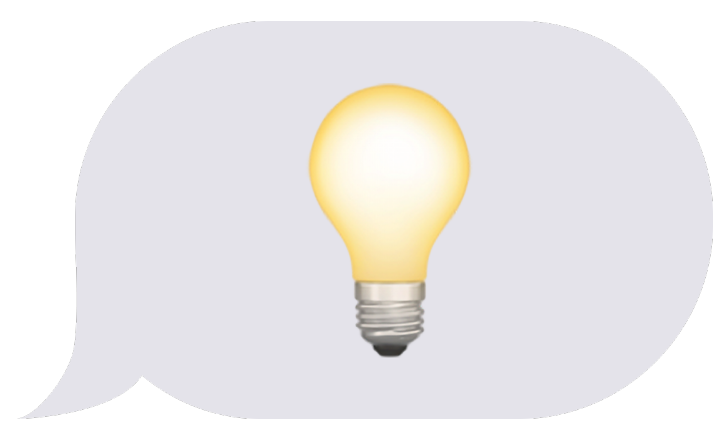
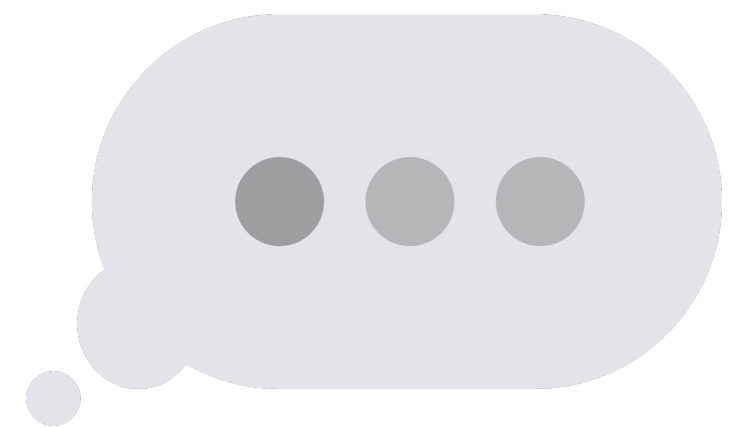


移动平台开发

实验课四：数据模型设计



2026春夏学期

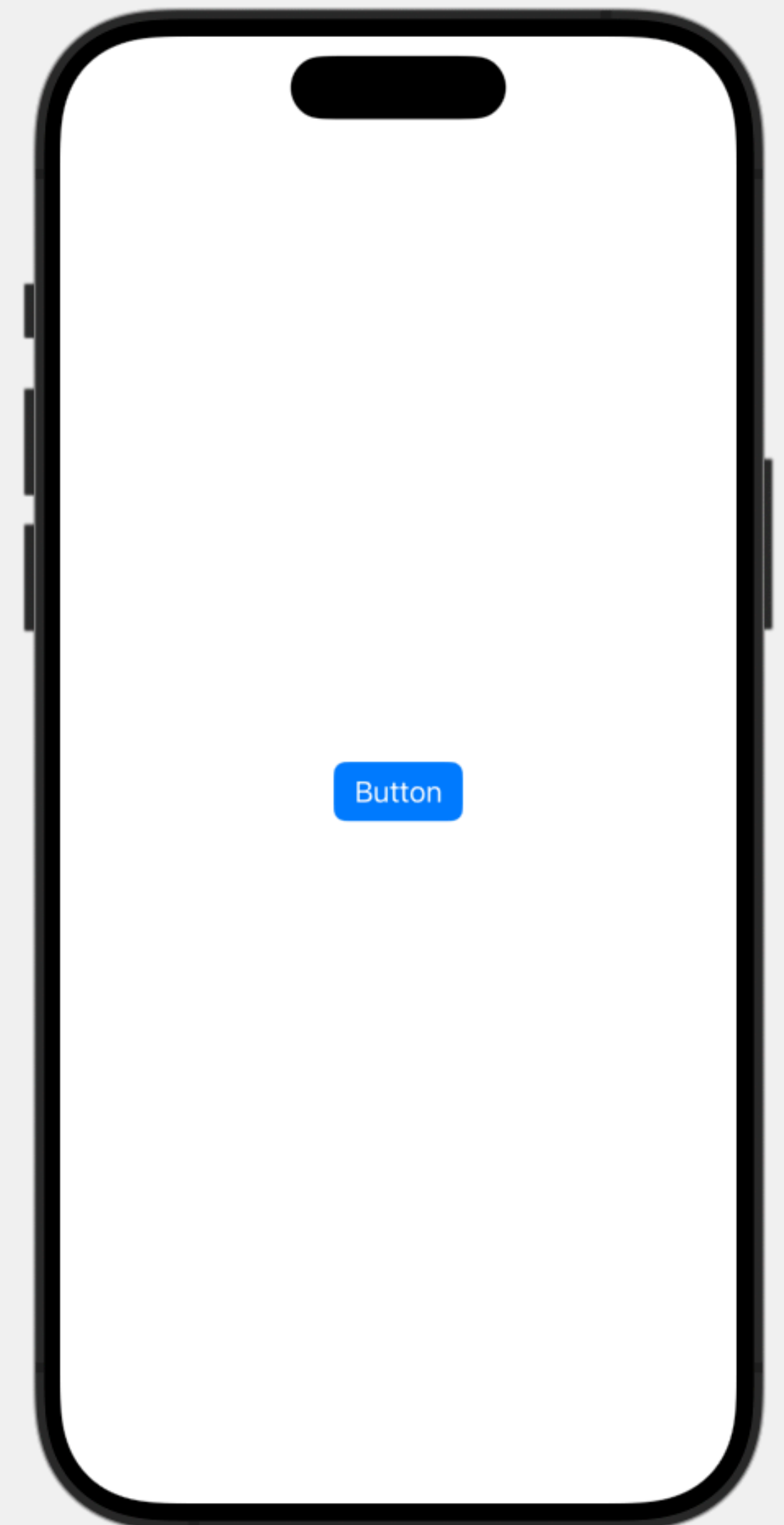
SwiftUI中的数据交互

Interacting data in SwiftUI



浙江大学
ZHEJIANG UNIVERSITY

```
func printOK()  
{  
    var a = 0  
    for i in 0..  
{  
        a += i  
    }  
    print("OK")  
    print("a的数值是: " + String(a))  
}
```



SwiftUI中的数据交互

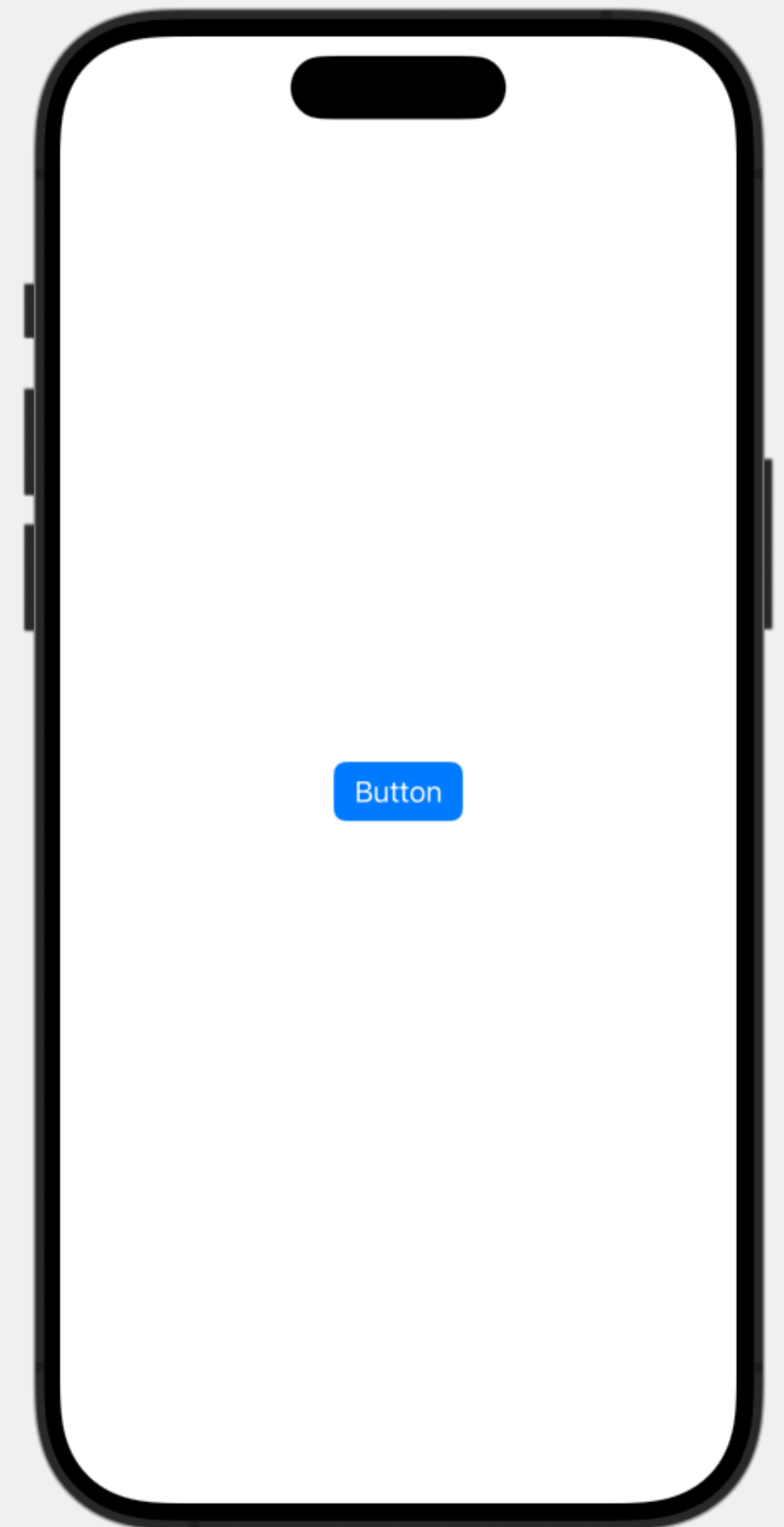
Interacting data in SwiftUI



浙江大学
ZHEJIANG UNIVERSITY

```
import SwiftUI
```

```
Button(action: {  
    printOK()  
})
```



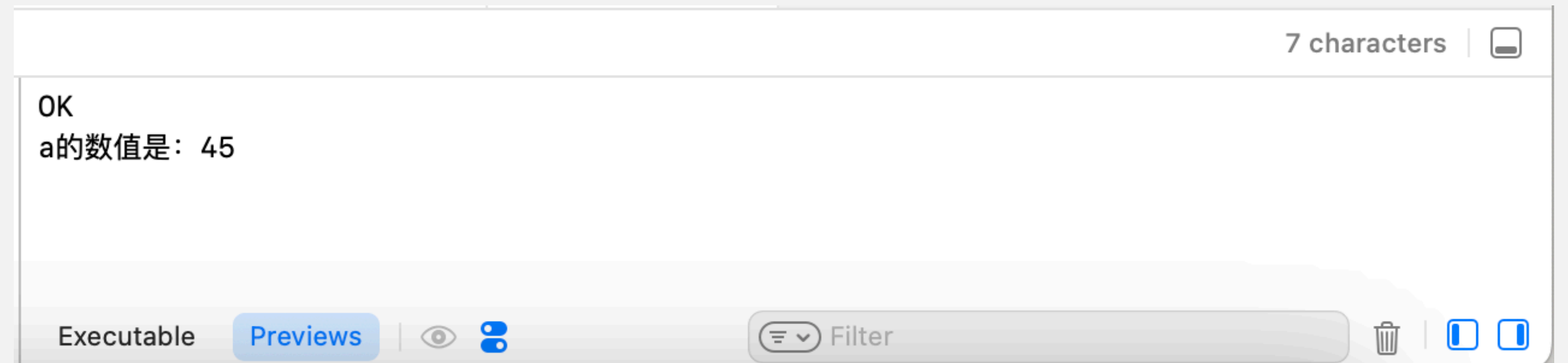
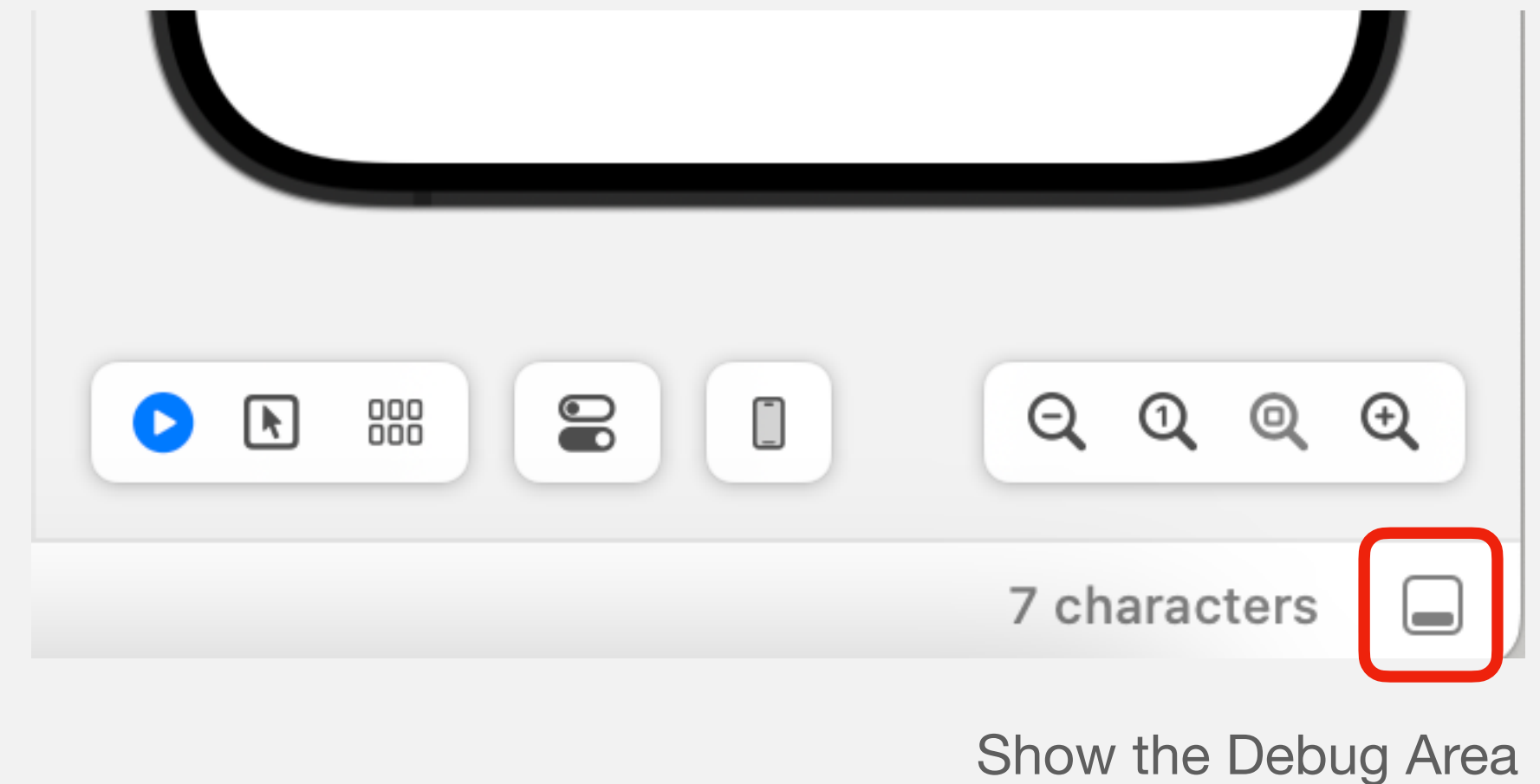
SwiftUI中的数据交互

Interacting data in SwiftUI



```
import SwiftUI
```

```
Button(action: {  
    printOK()  
})
```



SwiftUI中的数据交互

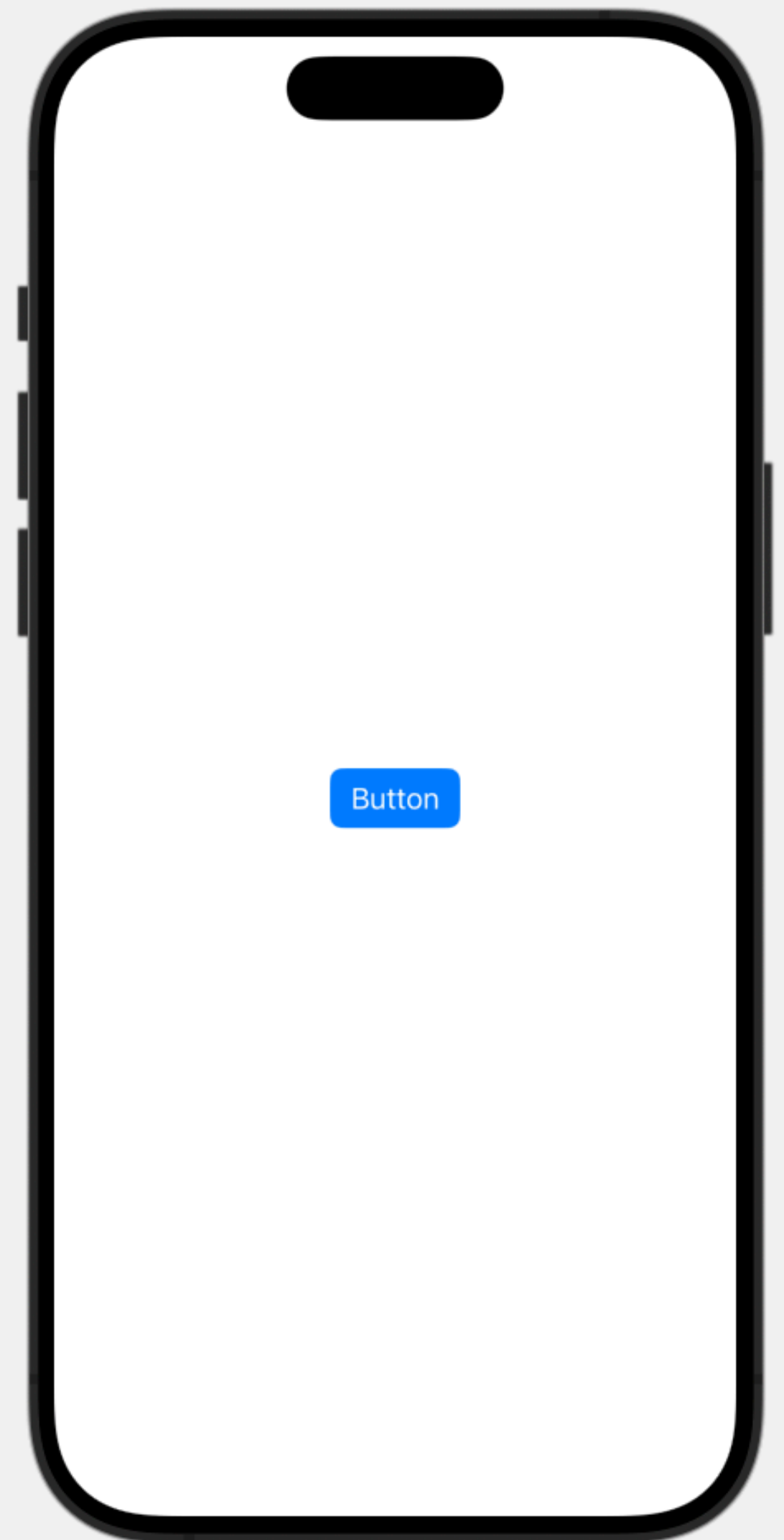
Interacting data in SwiftUI



浙江大学
ZHEJIANG UNIVERSITY

```
import SwiftUI
```

```
Button(action: {  
    printOK()  
})
```

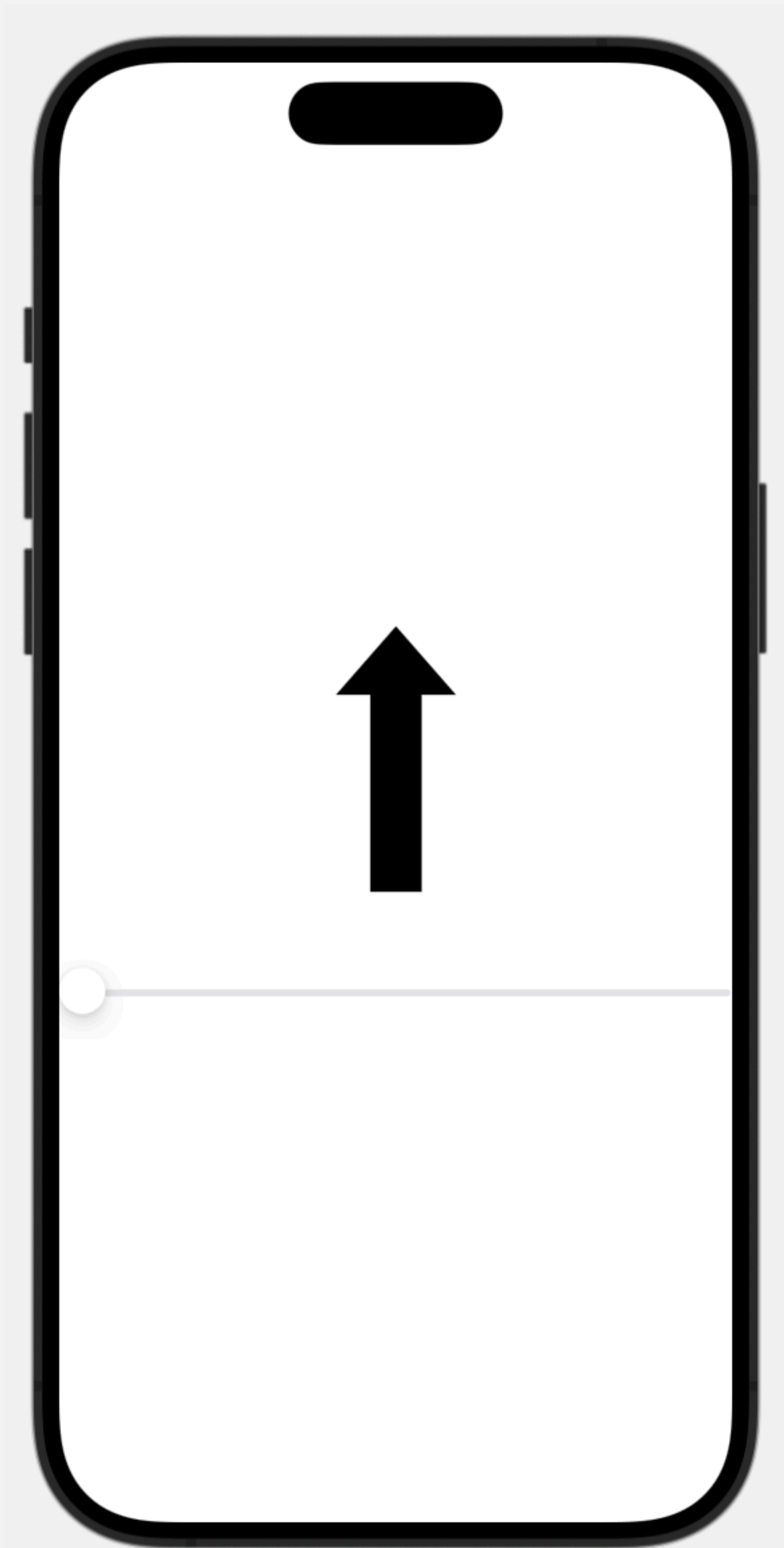


SwiftUI中的数据交互



Interacting data in SwiftUI

```
struct RotatableArrow: View {  
    var angle: Angle  
  
    var body: some View {  
        ZStack {  
            Rectangle()  
                .frame(width: 30, height: 130)  
            Triangle()  
                .frame(width: 70, height: 40)  
                .offset(x: 0, y: -70)  
        }  
        .frame(height: 200)  
        .rotationEffect(angle)  
    }  
}
```



SwiftUI中的数据交互

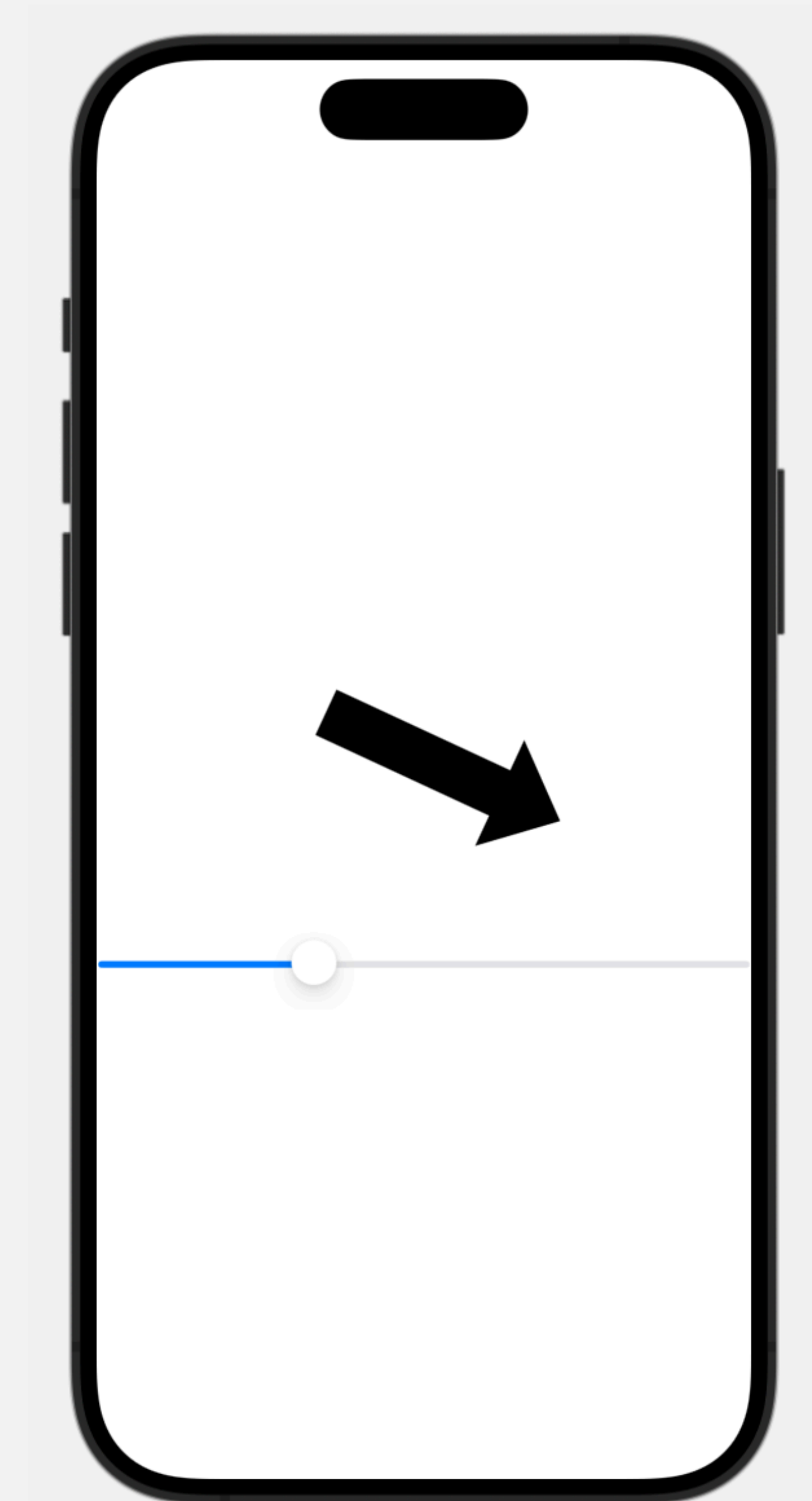
Interacting data in SwiftUI



浙江大学
ZHEJIANG UNIVERSITY

```
@State var angle: Double = 0.0
```

```
var body: some View {  
    VStack {  
        RotatableArrow(angle: Angle(degrees: angle))  
        Slider(value: $angle, in: 0...360, step: 0.1)  
    }  
}
```

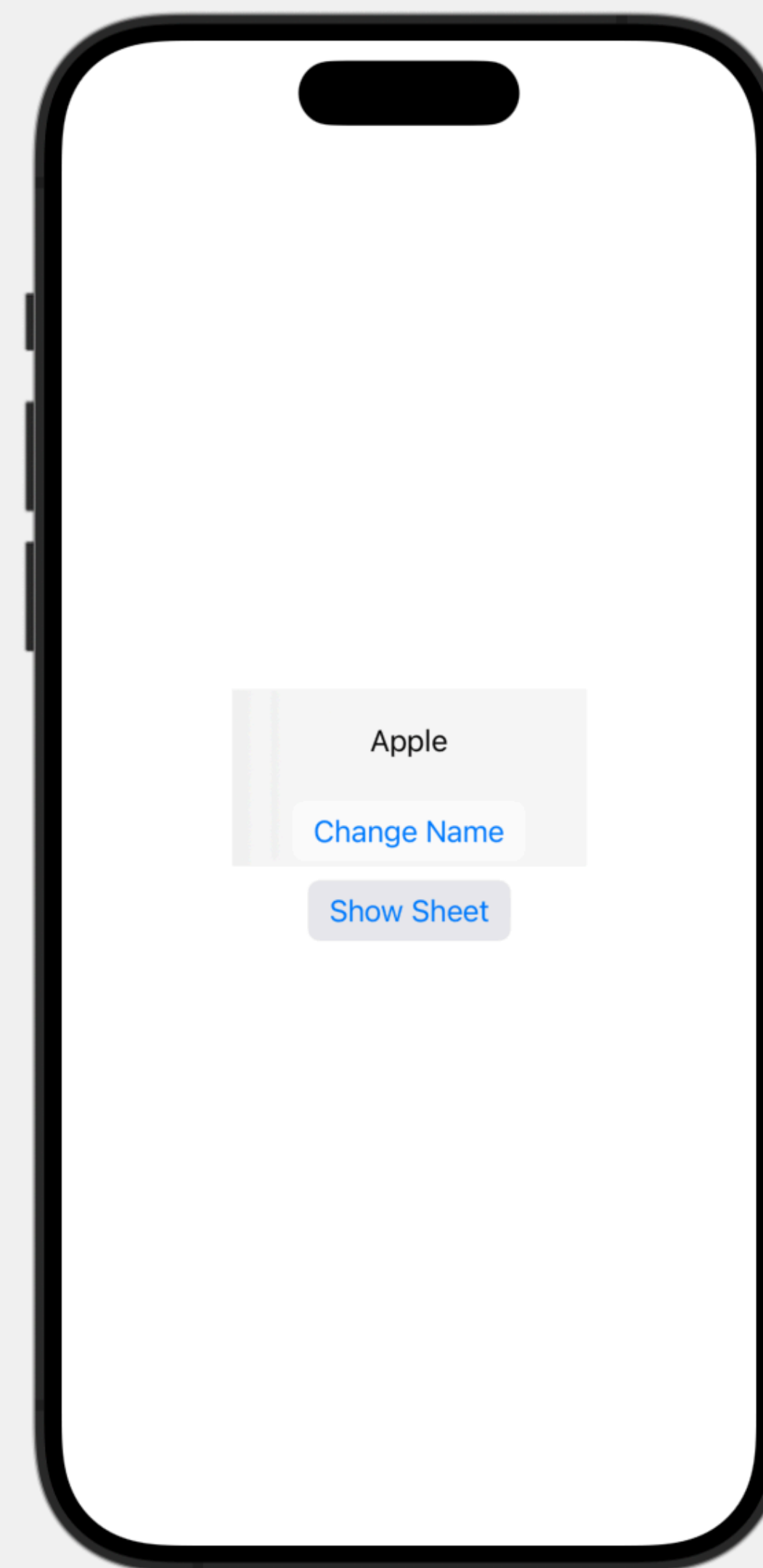




SwiftUI中的数据交互

Interacting data in SwiftUI

```
var name_list = ["Apple", "HWawei", "XIAOMI", "OPPO"]
@State var name_index = 0
var body: some View
{
    VStack
    {
        Text(name_list[name_index])
            .padding()
        Button("Change Name", action:
            {
                name_index += 1
                name_index = name_index % 4
            })
    }
}
```



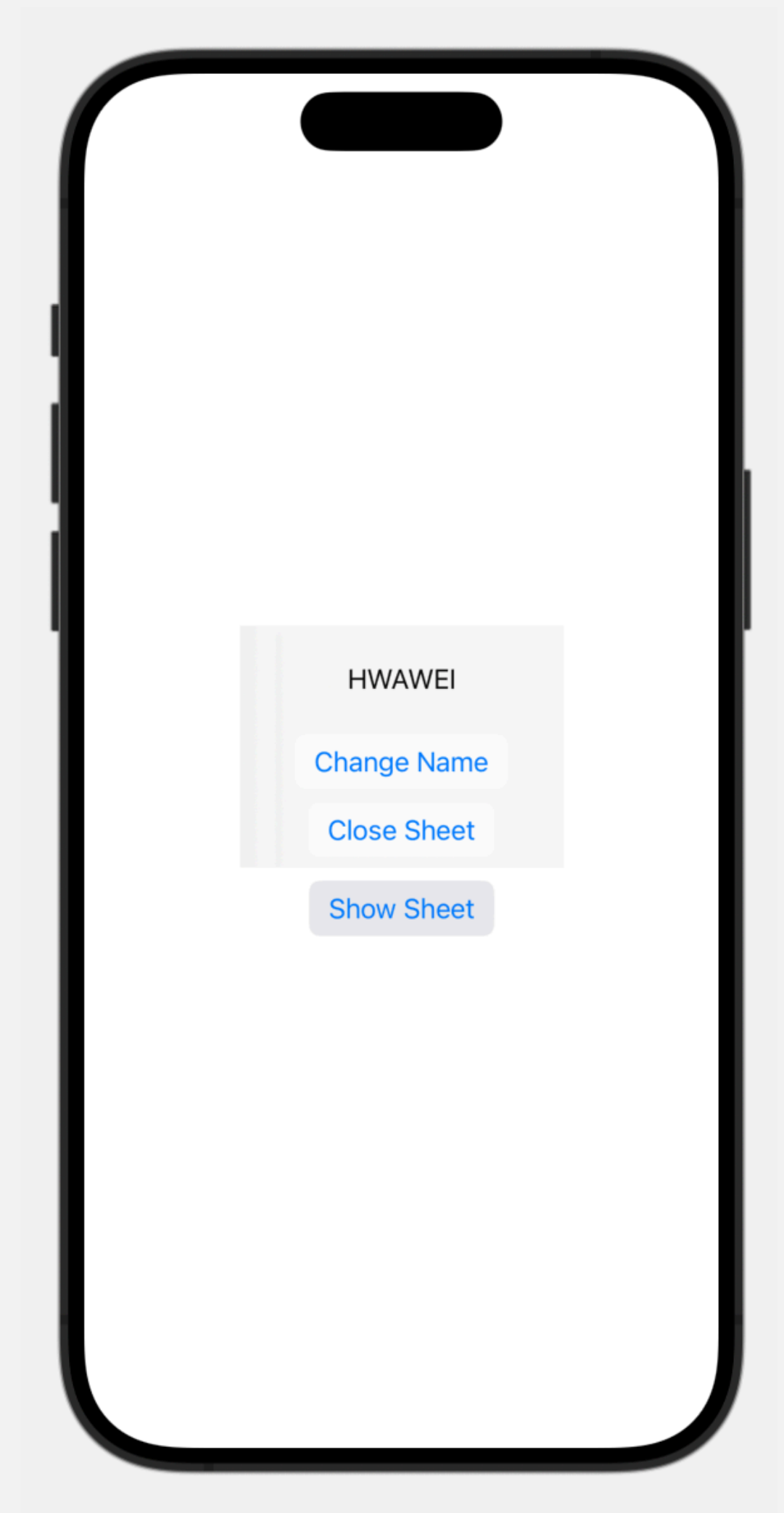
SwiftUI中的数据交互

Interacting data in SwiftUI



```
struct ShareDataPhoneNameView: View
{
    @Binding var show_sheet: Bool
    var name_list: [String]
    @Binding var name_index: Int
}
```

```
ShareDataPhoneNameView(
    show_sheet: $show_sheet,
    name_list: name_list,
    name_index: $name_index
)
```



SwiftUI中的数据交互

Interacting data in SwiftUI



浙江大学
ZHEJIANG UNIVERSITY

```
1  
2 {  
3     "name": "Apple",  
4     "times": "0"  
5 }  
6
```

data.json 文件

- 应用包 (Bundle) 中的 **只读** JSON 文件
(程序内置资源, 无法修改)
- 沙盒 (Sandbox) 中的 **可读可写** JSON 文件
(应用运行时可以创建、修改和读取)

SwiftUI中的数据交互

Interacting data in SwiftUI



浙江大学
ZHEJIANG UNIVERSITY

```
static func JsonRead(filename: String) -> Dictionary<String, String>
{
    ..
    // 优先从沙盒（可读写区）读取 JSON 文件，如果不存在，就从 Bundle 复制一份
}
```

```
static func JsonWrite(filename: String, data: Dictionary<String, String>)
{
    ..
    // 将 JSON 数据写入到 Documents 目录，使其在下次应用启动时可用
}
```

SwiftUI中的数据交互



Interacting data in SwiftUI

```
import Foundation
```

```
...
```

```
// 读取只读文件
```

```
static func JsonReadOrigin(filename: String) -> Dictionary<String, String>
```

```
if let path = Bundle.main.path(forResource: String(res[0]), ofType: String(res[1])) {
```

```
    do {
```

```
        let data = try Data(contentsOf: URL(fileURLWithPath: path), options: .mappedIfSafe)
```

```
        let orgJsonResult = try JSONSerialization.jsonObject(with: data, options: .mutableLeaves)
```

```
        if let newJsonResult = orgJsonResult as? Dictionary<String, String> {
```

```
            // 在此处处理JSON数据
```

```
            jsonResult = newJsonResult
```

```
        }
```

```
    } catch {
```

```
        // 错误处理
```

```
    }
```

```
}
```

```
1
2 {
3     "name": "Apple",
4     "times": "0"
5 }
6
```

data.json 文件

SwiftUI中的数据交互

Interacting data in SwiftUI

```
import Foundation
```

```
...
```

```
// 读取初始只读文件，并且将其写入App的可读可写文件夹内
```

```
static func JsonStartFromOrigin(filename: String) -> Dictionary<String, String>
if let path = Bundle.main.path(forResource: String(res[0]), ofType: String(res[1])) {
    do {
        let data = try Data(contentsOf: URL(fileURLWithPath: path), options: .mappedIfSafe)
        let orgJsonResult = try JSONSerialization.jsonObject(with: data, options: .mutableLeaves)
        if let newJsonResult = orgJsonResult as? Dictionary<String, String> {
            jsonResult = newJsonResult
            iOS_JSON.JsonWrite(filename: filename, data: jsonResult)
        }
    } catch {
        // 错误处理
    }
}
```

```
1
2 {
3     "name": "Apple",
4     "times": "0"
5 }
6
```

data.json 文件

SwiftUI中的UIKit

UIKit in SwiftUI

```
import UIKit
...

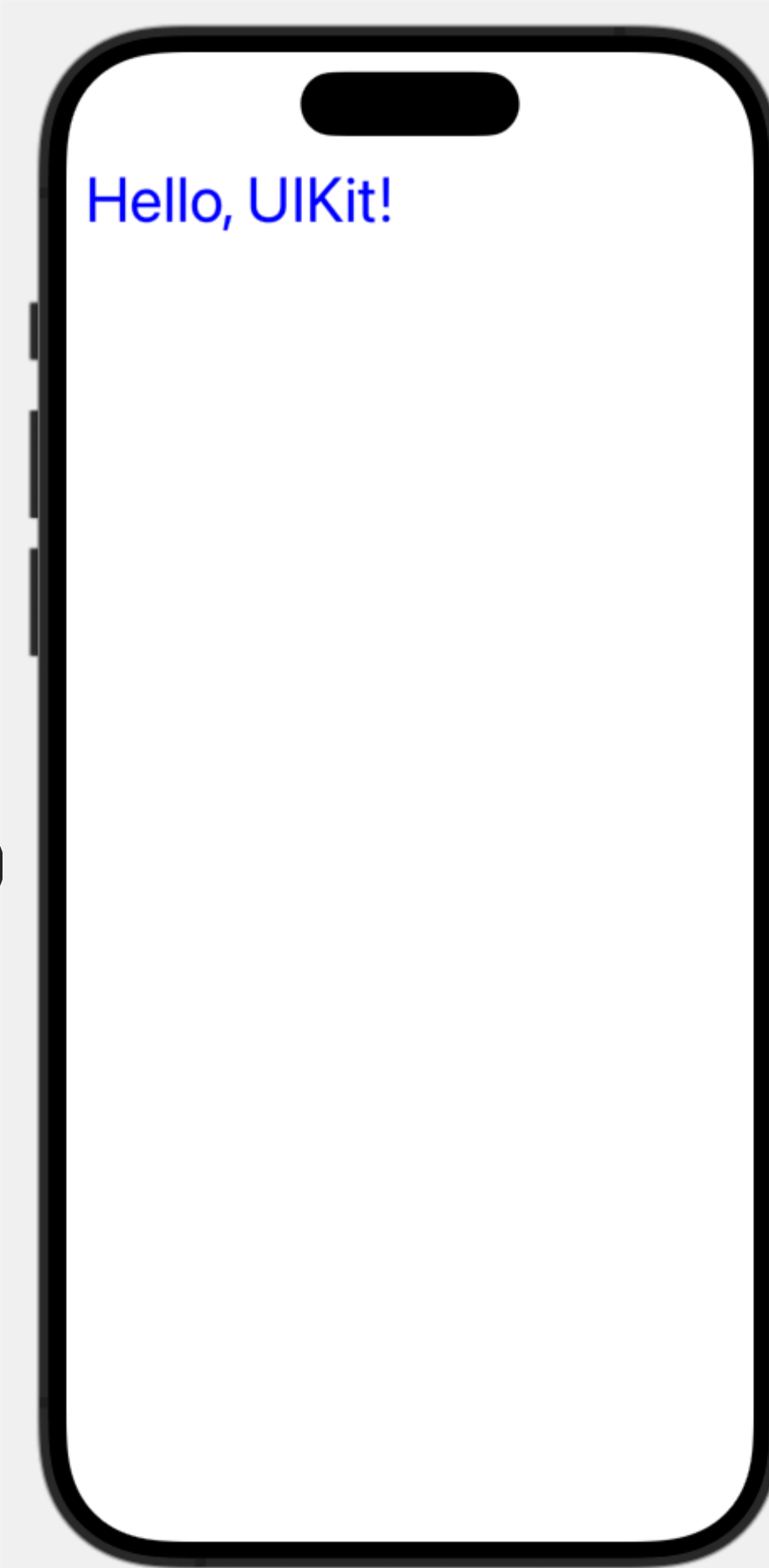
class CustomUIKitView: UIView {

    override init(frame: CGRect) {
        super.init(frame: frame)

        let button = UIButton(frame: CGRect(x: 0, y: 0, width: 200, height: 50))

        button.setTitle("Hello, UIKit!", for: .normal)
        button.titleLabel?.font = UIFont.systemFont(ofSize: 36)
        button.setTitleColor(.blue, for: .normal)

        addSubview(button)
    }
}
```



SwiftUI中的UIKit

UIKit in SwiftUI

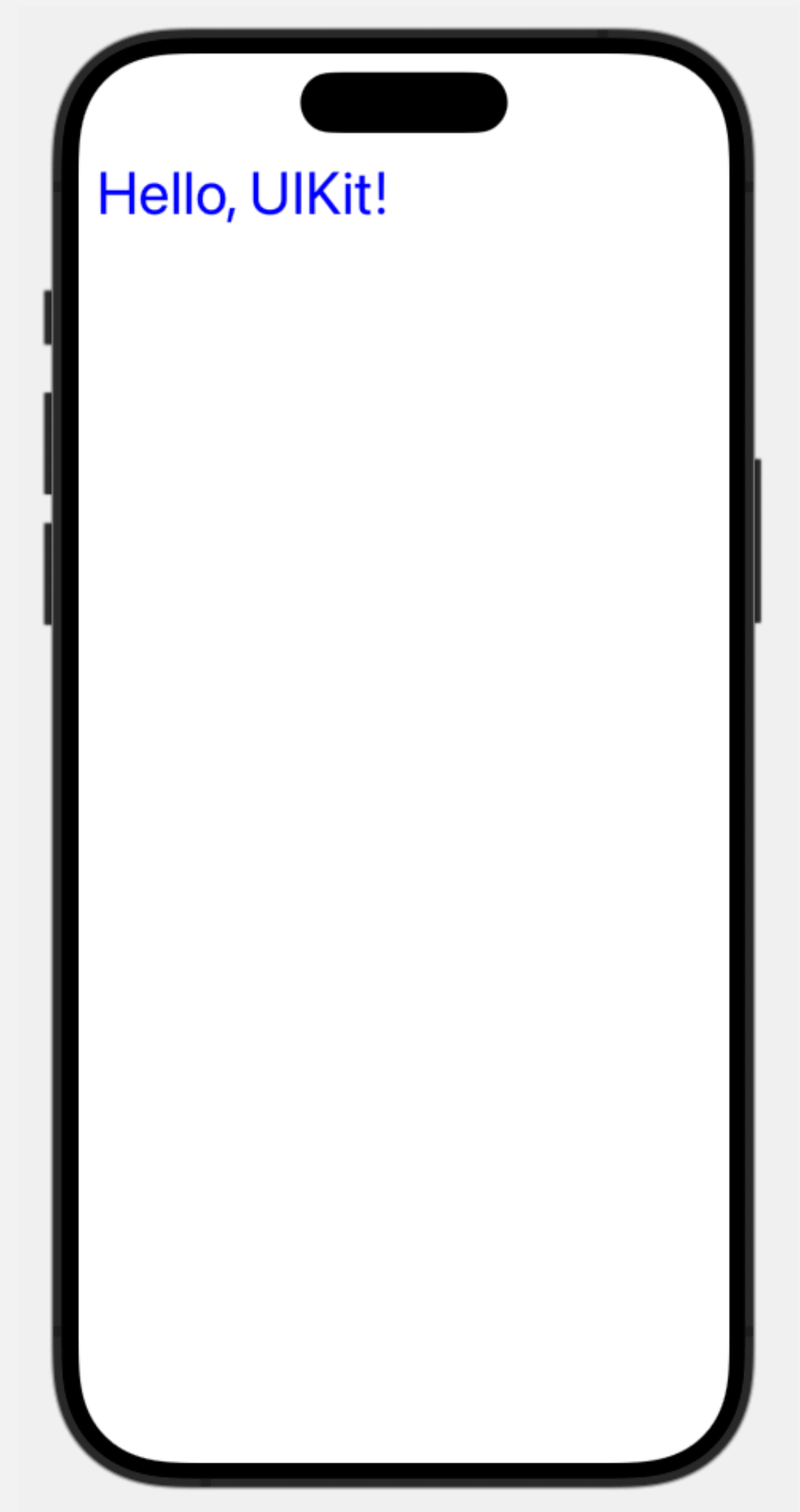
```
import UIKit
...
import SwiftUI

struct UIKitRepresentable: UIViewRepresentable {
    func makeUIView(context: Context) -> CustomUIKitView {
        return CustomUIKitView()
    }
}

struct SwiftUIView: View {
    var body: some View {
        UIKitRepresentable()
    }
}
```



浙江大学
ZHEJIANG UNIVERSITY



MVVM模式实例

MVVM in SwiftUI

```
struct SharedDataModel
{
    // data..
}

class SharedDataViewModel : ObservableObject
{
    @Published private var data = SharedDataModel()
    // add func..
}

struct ShareDataMainView: View {
    @StateObject var shared_view_model = SharedDataViewModel()
    var body: some View
    {
        // add UI..
    }
}
```

