



iOS Club 第二课：SwiftUI 入门 + Git 实战协作

课程日期：2025-10-12

一、开场（3分钟）

主题引入：

“上节课我们学了 Xcode 的使用；
今天，我们要一起写第一个 SwiftUI 小程序，并用 Git 把它变成一个能被所有人协作的项目。”

课程目标：

1. 掌握 SwiftUI 的基本结构与模块
 2. 学会用 Git 管理项目版本
 3. 模拟 iOS Club 实际项目协作流程
-

二、**SwiftUI** 基础模块讲解 (20分钟)

1. **SwiftUI** 项目结构

```
@main
struct DemoApp: App {
    var body: some Scene {
        WindowGroup {
            ContentView()
        }
    }
}
```

- `App` 协议：程序入口
- `ContentView`：主界面
- `body`：描述界面内容



2. 常见基础模块与示例

👉 布局容器 (Layout)

用于排列子视图：

```
struct LayoutExample: View {
    var body: some View {
        VStack {
            Text("垂直布局 VStack")
            HStack {
                Text("左")
                Spacer()
                Text("右")
            }
            ZStack {
                Color.yellow
                Text("叠放在背景上的文字")
            }
        }
        .padding()
    }
}
```

- `VStack` → 垂直排列
- `HStack` → 水平排列
- `ZStack` → 层叠排列（后写的在上面）



文本与图片 (Text & Image)

```
struct TextImageExample: View {
    var body: some View {
        VStack {
            Text("Hello, iOS Club!")
                .font(.title)
                .foregroundColor(.blue)
                .bold()
                .padding()

            Image(systemName: "star.fill")
                .resizable()
                .frame(width: 60, height: 60)
                .foregroundColor(.yellow)
        }
    }
}
```

📌 `Image(systemName:)` 可直接使用 `SF Symbols` 图标。



🔴 控件 (Controls)

```
struct ControlsExample: View {
    @State private var count = 0
    @State private var name = ""
    @State private var isDark = false

    var body: some View {
        VStack(spacing: 20) {
            TextField("请输入你的名字", text: $name)
```

```

        .textFieldStyle(.roundedBorder)
        .padding()

Button("打招呼") {
    count += 1
}
.buttonStyle(.borderedProminent)

Text("你好, \(name), 你点击了 \(count) 次")

Toggle("夜间模式", isOn: $isDark)
    .padding()
}
.padding()
.background(isDark ? Color.black : Color.white)
.foregroundColor(isDark ? .white : .black)
}
}

```

✅ 改进点:

- 用户输入的名字会实时显示在屏幕上
- 点击次数可计数
- 夜间模式可动态切换



🎨 修饰器 (Modifiers)

```

struct ModifierExample: View {
    var body: some View {
        Text("这是被修饰的文字")
            .font(.largeTitle)
            .foregroundColor(.purple)
            .padding()
            .background(Color.yellow)
            .cornerRadius(12)
            .shadow(radius: 5)
    }
}

```



导航与列表 (Navigation & List)

```
struct NavigationExample: View {
    var body: some View {
        NavigationView {
            List {
                NavigationLink("前往详情页", destination: Text("这里是详情内容"))
                NavigationLink("关于 iOS Club", destination: Text("我们是热爱苹果开发的社团！"))
            }
            .navigationTitle("导航示例")
        }
    }
}
```



状态管理解释：为什么是空字符串 ""

```
@State private var name = ""
```

说明：

- `@State`：表示这个变量会随用户操作自动刷新界面。
- `= ""`：给变量一个初始空值，否则输入框会显示 `nil` 或报错。
- 在 SwiftUI 中常见写法：
- `@State private var name = ""` → 初始空文本
- `@State private var count = 0` → 初始数字 0
- `@State private var isDark = false` → 初始开关关闭



3. 综合小项目演示

```
struct ContentView: View {
    @State private var name = ""
```

```

@State private var isDark = false

var body: some View {
    NavigationView {
        VStack(spacing: 20) {
            Text("欢迎来到 iOS Club")
                .font(.title)
                .foregroundColor(isDark ? .white : .blue)

            TextField("请输入你的名字", text: $name)
                .textFieldStyle(.roundedBorder)
                .padding()

            Button("打招呼") { }
                .buttonStyle(.borderedProminent)

            if !name.isEmpty {
                Text("你好, \(name)! 🎉")
                    .font(.headline)
            }

            Toggle("夜间模式", isOn: $isDark)
                .padding()

            Spacer()
        }
        .padding()
        .background(isDark ? .black : .white)
        .navigationTitle("SwiftUI 入门")
    }
}

```



三、Git 管理 SwiftUI 项目 (15分钟)

初始化仓库

```
git init
git add .
git commit -m "🎉 初始提交: 添加 SwiftUI 界面"

# 连接 GitHub
git remote add origin https://github.com/iosclub-ouc/swiftui-demo.git
git branch -M main
git push -u origin main
```

分支开发示例

```
git checkout -b feature-darkmode
# 添加夜间模式代码后:
git add .
git commit -m "✨ 新增夜间模式切换功能"
git push origin feature-darkmode
```

在 [GitHub](#) 上创建 [Pull Request](#)，合并到主分支。



四、iOS Club 协作标准 (8分钟)

| 阶段 | 要求 |
|----------------|----------------------------|
| 功能开发 | 新分支 (feature/xxx) |
| Bug 修复 | fix/xxx |
| 文档更新 | docs/xxx |
| Commit Message | 简洁明确, 如 feat: 新增登录页 |
| 合并流程 | PR → Review → Merge → Pull |



五、拓展与总结 (4分钟)

- SwiftUI → 快速搭建现代 iOS 界面
- Git → 管理版本与团队协作
- 结合 → iOS Club 实现真实开发流程

课后任务:

- 1. 写一个简单 SwiftUI 界面
- 2. 上传到 GitHub
- 3. 提交 PR 到社团仓库



六、时间分配

| 模块 | 内容 | 时长 |
|---------|---------------|--------------|
| SwiftUI | 讲解与演示 | 模块 + 实例 20分钟 |
| Git 实战 | 初始化 + 推送 + 协作 | 15分钟 |
| 协作标准与总结 | 讲解 + QA | 10分钟 |
| 合计 | | 45分钟 |

