

# SwiftUI vs UIKit

그래서 도대체 뭘 쓰라는 거야? 🤯

# SwiftUI

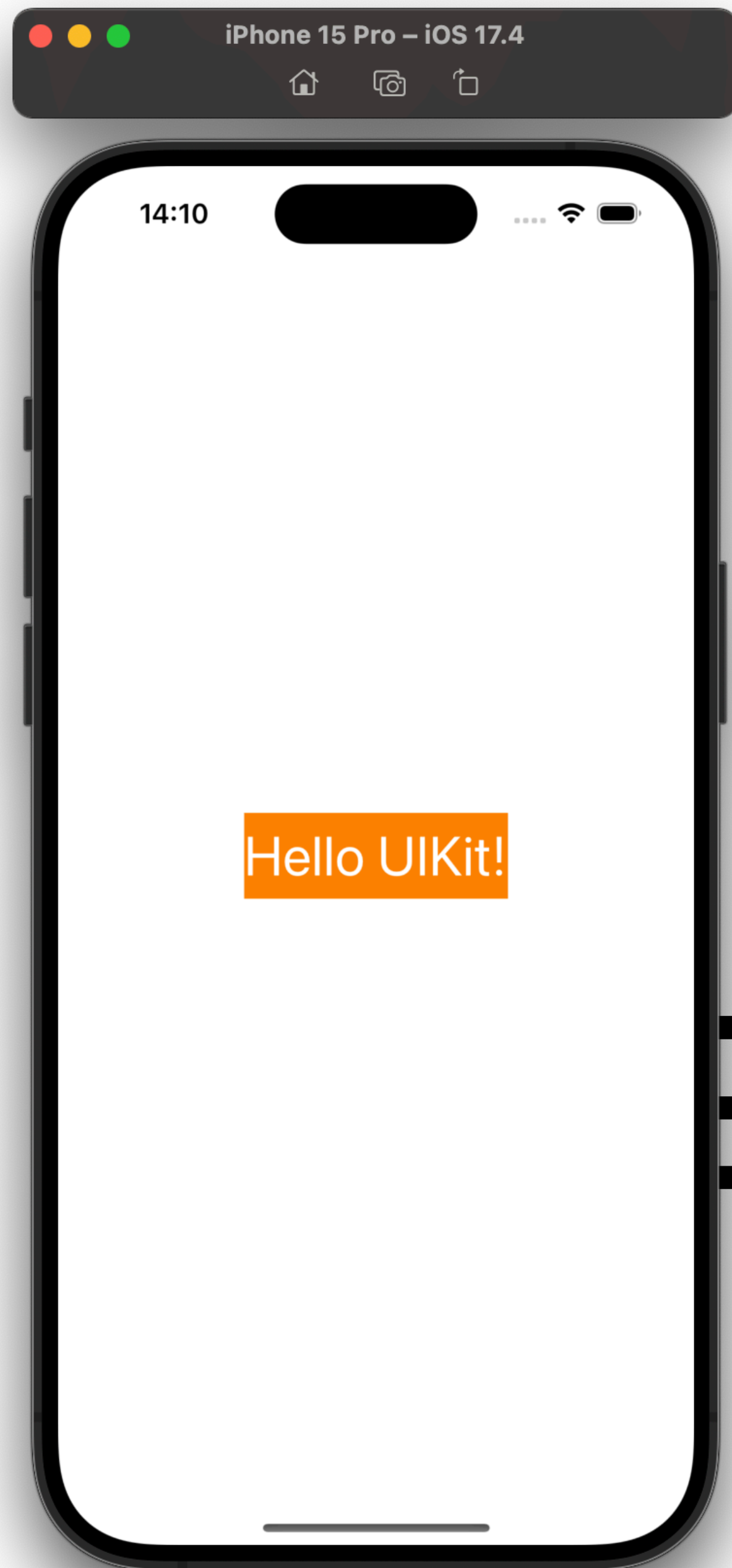
- 선언형 프로그래밍 프레임워크
- 비교적 단순한 개발 과정
- 복잡한 UI도 간결한 코드로 작성



# UIKit

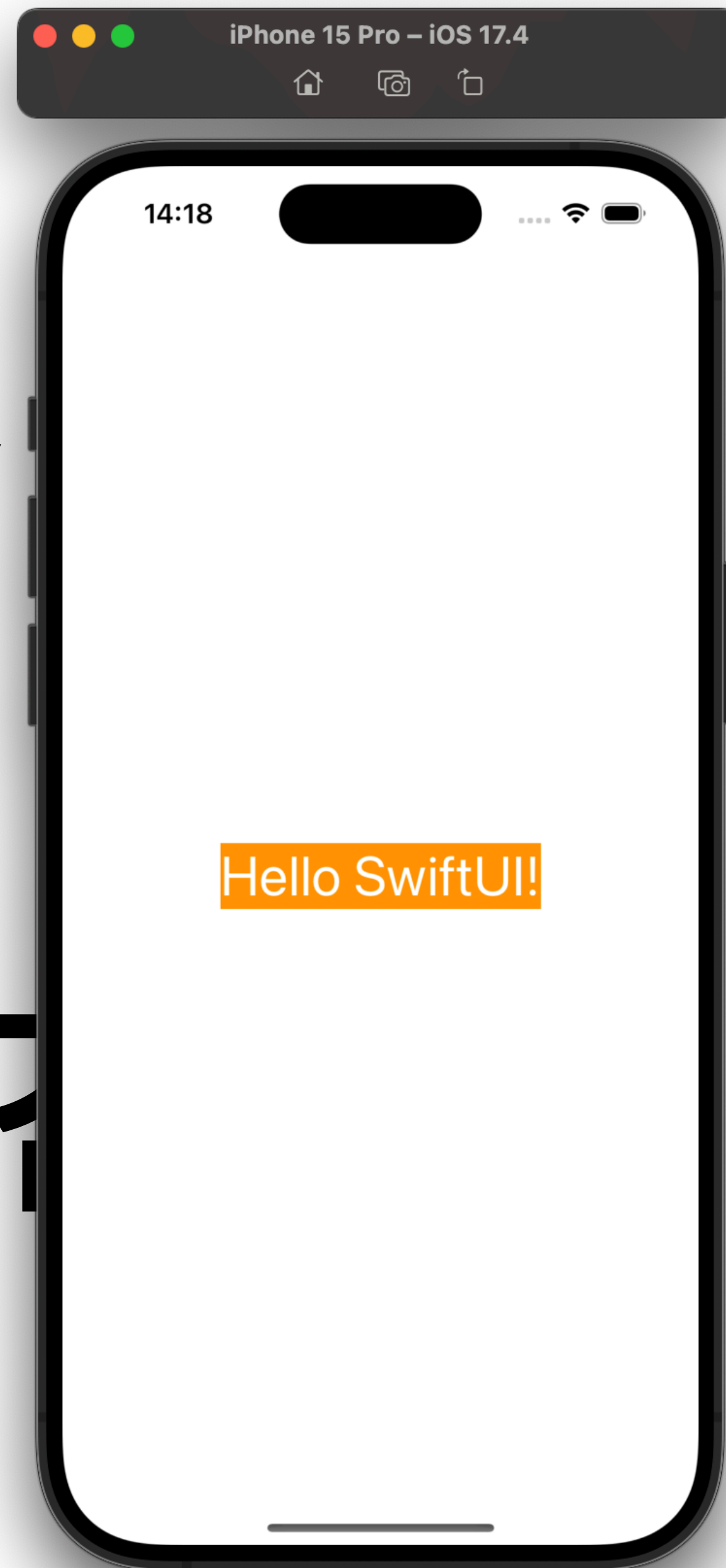
- 명령형 프로그래밍 프레임워크
- View를 직접 생성 후 제어
- 더 많은 UI 제어와 기능을 제공





행 / 의

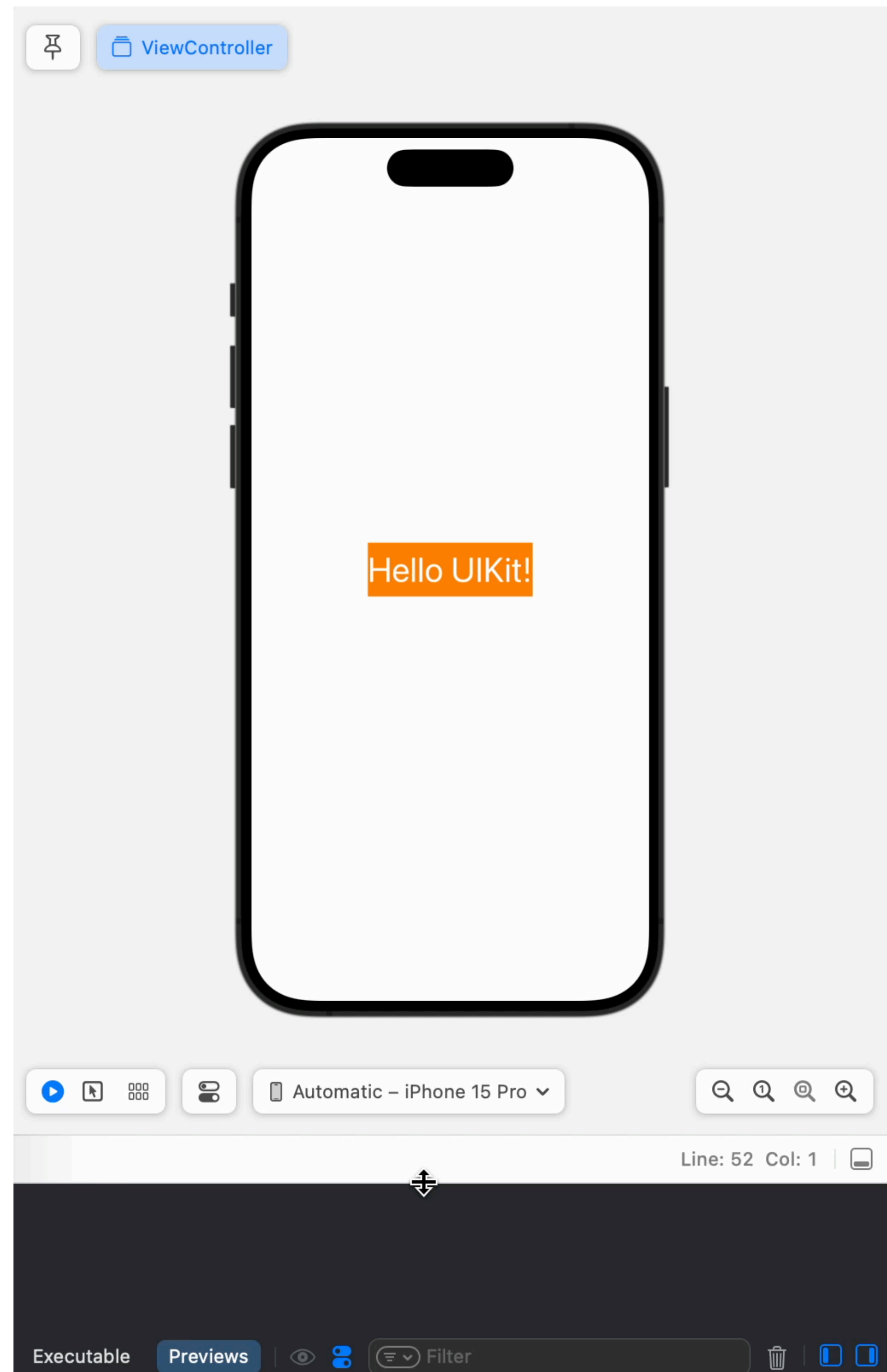
&



```

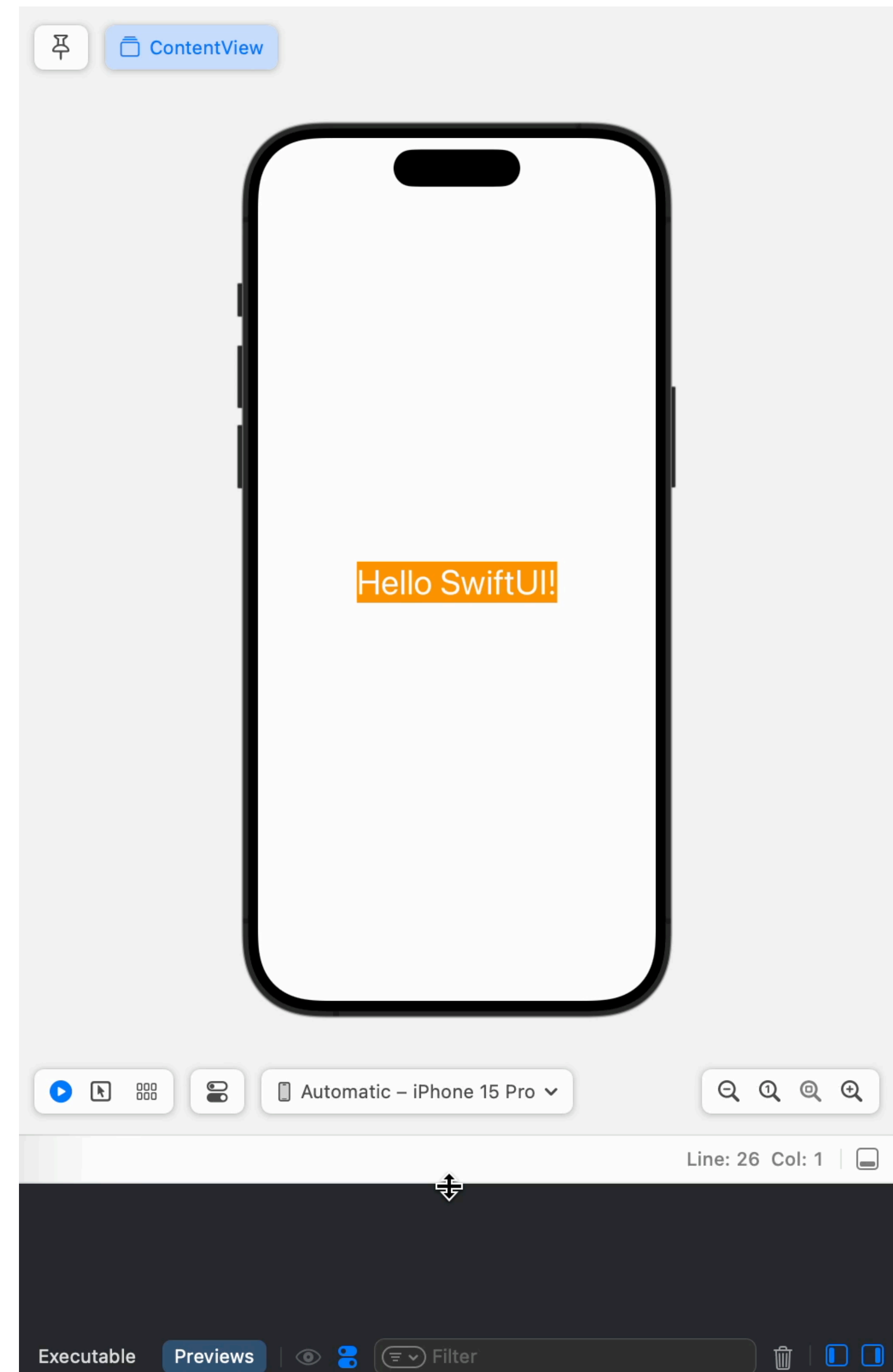
8  import UIKit
9
10 class ViewController: UIViewController {
11
12     // MARK: - Button 선언
13     var button: UIButton = {
14         let button = UIButton(type: .system)
15         button.setTitle("Hello UIKit!", for: .normal)
16         button.setTitleColor(.white, for: .normal)
17         button.titleLabel?.font = .preferredFont(forTextStyle: .largeTitle)
18         button.backgroundColor = .orange
19         return button
20     }()
21
22     override func viewDidLoad() {
23         super.viewDidLoad()
24
25         // MARK: - Button Action 할당
26         button.addTarget(self, action: #selector(buttonAction(_:)), for: .touchUpInside)
27
28         // MARK: - 화면 내에 Button 추가
29         self.view.addSubview(button)
30
31         // MARK: - Button Layout 설정
32         button.translatesAutoresizingMaskIntoConstraints = false
33
34         NSLayoutConstraint.activate([
35             button.centerXAnchor.constraint(equalTo: view.centerXAnchor),
36             button.centerYAnchor.constraint(equalTo: view.centerYAnchor)
37         ])
38     }
39
40     // MARK: - Button Action func 구현
41     @objc func buttonAction(_ sender: UIButton) {
42         print("Clicked UIKit Button!")
43     }
44
45 }

```

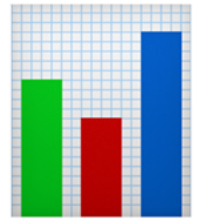


# SwiftUI Example

```
8  import SwiftUI
9
10 struct ContentView: View {
11     var body: some View {
12         Button {
13             // MARK: - Button의 Action 구성
14             print("Clicked SwiftUI Button!")
15         } label: {
16             // MARK: - Button의 Label(UI) 구성
17             Text("Hello SwiftUI!")
18                 .font(.largeTitle)
19                 .foregroundColor(Color.white)
20                 .background(Color.orange)
21         }
22     }
23 }
```







# Trend

## 요즘은 어떤 흐름인 거 같나요?

### 주요업무

- No.1 수능 공부 앱인 오르조 수능의 새로운 기능 개발 및 기존 기능 개선
- 개발 과정에서 발생하는 부정적 가치를 제거하고, 가치 생산을 위한 노력
- SwiftUI, Combine, TCA를 기반으로 프로젝트를 개발합니다

### 💡 우리는 이런 도구를 사용해요

강남언니 팀 iOS 개발자는 다음의 도구를 주로 사용해요. 그 밖에도 주어진 문제에 따라 다양한 도구를 도입합니다.

- SwiftUI, Concurrency, TCA, Tuist, Combine



Sponsorship session

제조업에서 **SwiftUI** + TCA로  
앱 개발하기

강성규 / 현대자동차

### 우대사항

- 7년이상 유경험자
- 태블릿 전용 애플리케이션을 개발해본 경험이 있으신 분
- SwiftUI를 통한 선언형 UI 개발에 익숙하신 분
- Combine, RxSwift 같이 스트림기반의 비동기 처리 프레임워크가 익숙하신 분
- TCA를 비롯한 다양한 데이터 플로우 아키텍처를 사용해보신 분

### 우대사항

< 이런 점이 있다면 더욱 좋아요 >

- 오픈소스등 다양한 코드 리딩 경험과 리팩토링 경험이 있으신 분
- 다양한 디자인 요구사항과 UX에 관심이 많고 프로토타이핑을 좋아하시는 분
- SwiftUI 에 대한 높은 이해도를 보유하신 분
- 타 직군과의 원활한 커뮤니케이션 능력을 보유하신 분

# **My Experience**

**실무자로서 느낀 게 있다면?**

- **Storyboard -> UIKit Code Base -> SwiftUI**
- **작년 7월 Design System 도입! 근데 SwiftUI로 구성된.**
- **Currently, SwiftUI**
- **But..... SwiftUI 100%는 아직 이르다.**
- **Why? UIKit != SwiftUI 니까!**



**“The Shortest Path to a Great App”**  
**“Less code, Better code, Everywhere”**

**WWDC 2019, Apple**



# Future

향후 몇 년 간 어떻게 변할까요?

- 최종적으로 Apple은 SwiftUI를 통한 Less Code를 원한다.
- SwiftUI 코드 작성 한 번으로 iOS, macOS, watchOS 지원
- 점차 UIKit의 기능을 보완하는 모습을 보이는 중 -> 앞으로도 그럴 것
- 그렇지만 우린 한 우물만 팔 수 없다.



**What's your opinion?**



**Anything else?**

**Finish ! Thanks you 🙌**