Mash-Up
iOS Team

2025.03.25

# SwiftUI 레이아웃의 이모저모고모숙모 ⁎⁎(ˆᗜˆ*)

발표자

| 최혜린

# SwiftUI 레이아웃 원리(feat. UIKit)

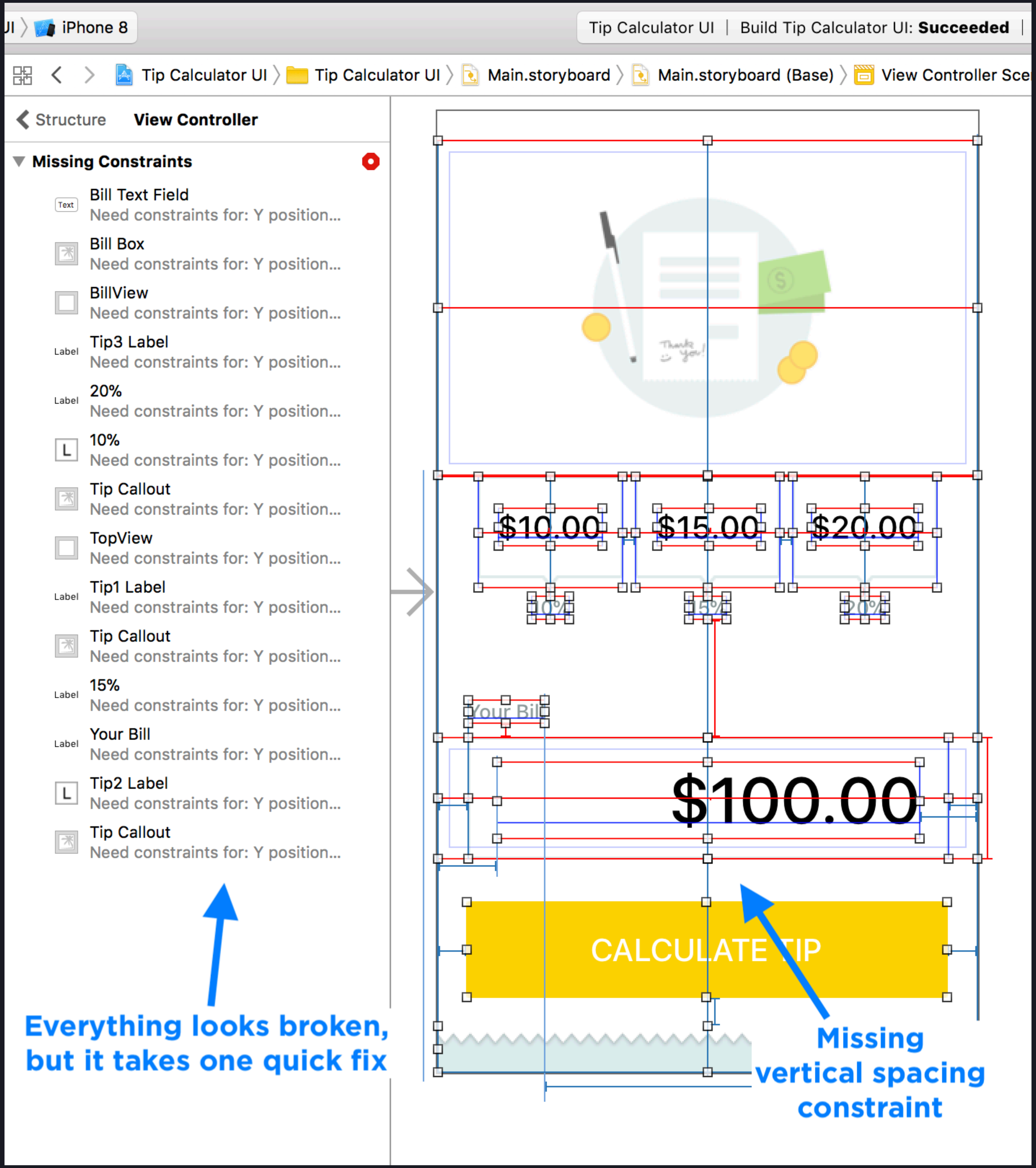
## 어라…? 스유 너 왜 이러는…?

WWDC 2019
Building Custom Views with SwiftUI
베이스입니다^^

## 원래 UIKit은 이랬어요!

**UIKit은 명령형 방식으로 레이아웃을 정의합니다.**

Auto Layout이 가장 메인이 되는 방식이었죠?


UI 〉 iPhone 8 | Tip Calculator UI | Build Tip Calculator UI: **Succeeded**

Tip Calculator UI 〉 Tip Calculator UI 〉 Main.storyboard 〉 Main.storyboard (Base) 〉 View Controller Sce

〈 Structure **View Controller**

**Missing Constraints**

Bill Text Field
Need constraints for: Y position...

Bill Box
Need constraints for: Y position...

BillView
Need constraints for: Y position...

Tip3 Label
Need constraints for: Y position...

20%
Need constraints for: Y position...

10%
Need constraints for: Y position...

Tip Callout
Need constraints for: Y position...

TopView
Need constraints for: Y position...

Tip1 Label
Need constraints for: Y position...

Tip Callout
Need constraints for: Y position...

15%
Need constraints for: Y position...

Your Bill
Need constraints for: Y position...

Tip2 Label
Need constraints for: Y position...

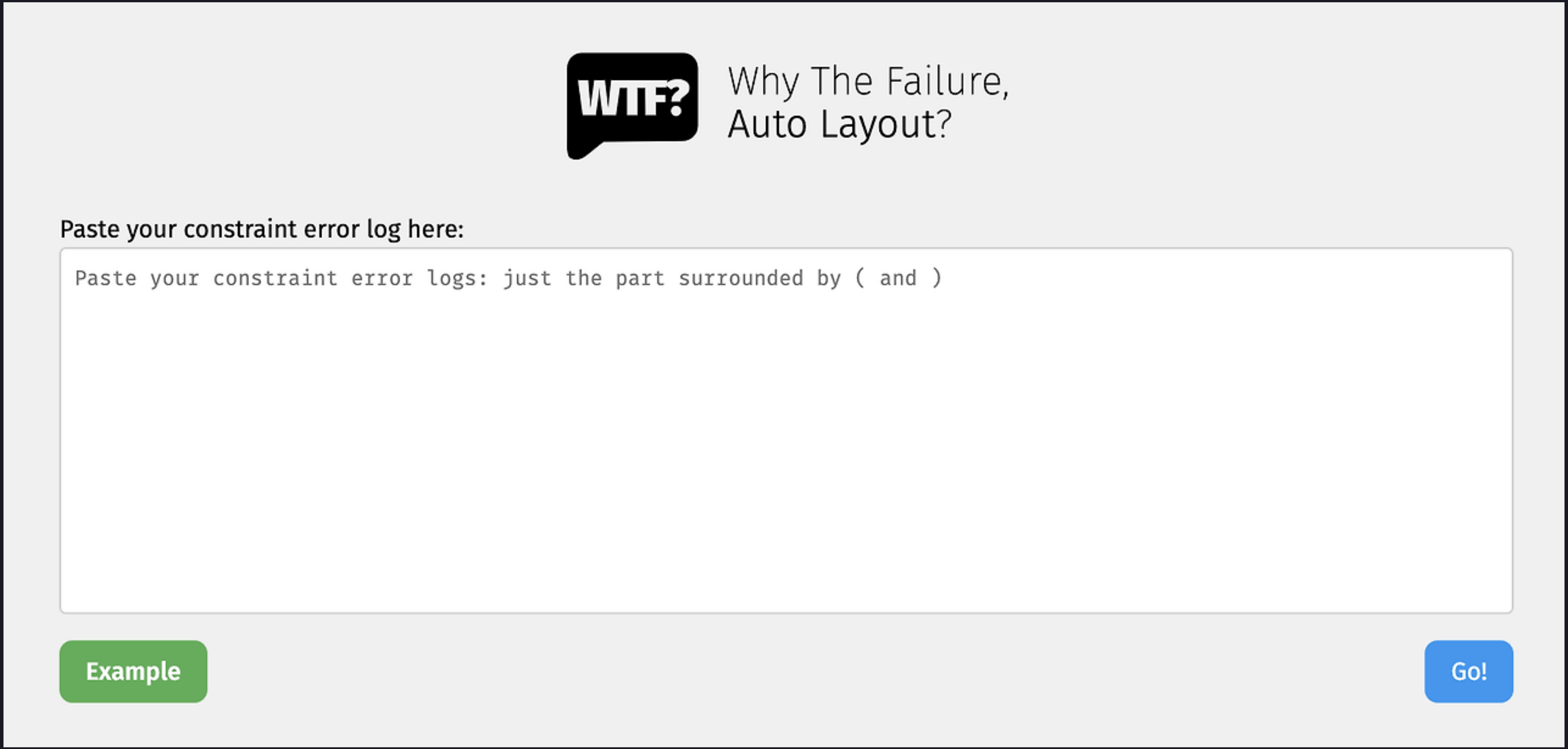
Tip Callout
Need constraints for: Y position...

**Everything looks broken, but it takes one quick fix**

**Missing vertical spacing constraint**

```
2020-06-12 15:26:05.109491+0900 CustomViewTest[27761:310387] [LayoutConstraints] Unable to simultaneously satisfy constraints.
	Probably at least one of the constraints in the following list is one you don't want.
	Try this:
		(1) look at each constraint and try to figure out which you don't expect;
		(2) find the code that added the unwanted constraint or constraints and fix it.
	(Note: If you're seeing NSAutoresizingMaskLayoutConstraints that you don't understand, refer to the documentation for the UIView
		property translatesAutoresizingMaskIntoConstraints)
(
    "<NSAutoresizingMaskLayoutConstraint:0x600001835cc0 h=-&- v=-&- UIView:0x7fab73408e50.minX == 0   (active, names:
        '|':CustomViewTest.CustomView:0x7fab73406d20 )>",
    "<NSAutoresizingMaskLayoutConstraint:0x600001835d10 h=-&- v=-&- H:[UIView:0x7fab73408e50]-(0)-|   (active, names:
        '|':CustomViewTest.CustomView:0x7fab73406d20 )>",
    "<NSAutoresizingMaskLayoutConstraint:0x600001835e50 h=--& v=--& CustomViewTest.CustomView:0x7fab73406d20.width == 0   (active)>",
    "<NSLayoutConstraint:0x60000180f6b0 UIImageView:0x7fab73409130.width == UIImageView:0x7fab73409130.height   (active)>",
    "<NSLayoutConstraint:0x60000180f390 UIImageView:0x7fab73409130.bottom == UIView:0x7fab73408fc0.bottom   (active)>",
    "<NSLayoutConstraint:0x60000180f340 V:|-(0)-[UIImageView:0x7fab73409130]   (active, names: '|':UIView:0x7fab73408fc0 )>",
    "<NSLayoutConstraint:0x60000180f2f0 H:|-(0)-[UIImageView:0x7fab73409130]   (active, names: '|':UIView:0x7fab73408fc0 )>",
    "<NSLayoutConstraint:0x60000180f570 H:[UIImageView:0x7fab73409130]-(10)-[UILabel:0x7fab73409700'furang']   (active)>",
    "<NSLayoutConstraint:0x60000180f2a0 H:[UILabel:0x7fab73409700'furang']-(0)-|   (active, names: '|':UIView:0x7fab73408fc0 )>",
    "<NSLayoutConstraint:0x60000180f070 UIView:0x7fab7340a570.width == UIView:0x7fab7340a570.height   (active)>",
    "<NSLayoutConstraint:0x6000018548c0 V:|-(0)-[UIView:0x7fab7340c1c0]   (active, names: '|':UIView:0x7fab7340c050 )>",
    "<NSLayoutConstraint:0x6000018549b0 UIView:0x7fab7340d080.bottom == UIView:0x7fab7340d3f0.top   (active)>",
    "<NSLayoutConstraint:0x600001854aa0 V:[UIView:0x7fab7340c1c0]-(0)-[UIView:0x7fab7340d080]   (active)>",
    "<NSLayoutConstraint:0x600001854af0 UIView:0x7fab7340d3f0.bottom == UIView:0x7fab7340c050.bottom   (active)>",
    "<NSLayoutConstraint:0x600001854be0 UIView:0x7fab7340d3f0.height == UIView:0x7fab7340c1c0.height   (active)>",
    "<NSLayoutConstraint:0x600001854c80 V:|-(0)-[UIView:0x7fab7340aba0]   (active, names: '|':UIView:0x7fab7340aa30 )>",
    "<NSLayoutConstraint:0x600001854d70 UIView:0x7fab7340c050.bottom == UIView:0x7fab7340aa30.bottom   (active)>",
    "<NSLayoutConstraint:0x600001854e10 V:[UIView:0x7fab7340aba0]-(0)-[UIView:0x7fab7340c050]   (active)>",
    "<NSLayoutConstraint:0x600001854eb0 V:|-(0)-[UIView:0x7fab73408fc0]   (active, names: '|':UIView:0x7fab73408e50 )>",
    "<NSLayoutConstraint:0x600001854f00 UIView:0x7fab73408fc0.height == 0.07*UIView:0x7fab73408e50.height   (active)>",
    "<NSLayoutConstraint:0x600001854f50 H:|-(10)-[UIView:0x7fab73408fc0]   (active, names: '|':UIView:0x7fab73408e50 )>",
    "<NSLayoutConstraint:0x600001854fa0 UIView:0x7fab73408fc0.trailing == UIView:0x7fab73408e50.trailing - 10   (active)>",
    "<NSLayoutConstraint:0x600001854ff0 V:[UIView:0x7fab73408fc0]-(0)-[UIView:0x7fab7340a570]   (active)>",
    "<NSLayoutConstraint:0x600001855040 UIView:0x7fab7340a570.trailing == UIView:0x7fab73408e50.trailing   (active)>",
    "<NSLayoutConstraint:0x600001855090 H:|-(0)-[UIView:0x7fab7340a570]   (active, names: '|':UIView:0x7fab73408e50 )>",
    "<NSLayoutConstraint:0x600001855130 V:[UIView:0x7fab7340a570]-(0)-[UIView:0x7fab7340aa30]   (active)>",
    "<NSLayoutConstraint:0x600001855180 UIView:0x7fab7340aa30.bottom == UIView:0x7fab73408e50.bottom   (active)>",
    "<NSLayoutConstraint:0x600001855220 UIView:0x7fab7340aba0.height == 0.07*UIView:0x7fab73408e50.height   (active)>",
    "<NSLayoutConstraint:0x600001855270 UIView:0x7fab7340c1c0.height == 0.05*UIView:0x7fab73408e50.height   (active)>"
)

Will attempt to recover by breaking constraint
<NSLayoutConstraint:0x600001854aa0 V:[UIView:0x7fab7340c1c0]-(0)-[UIView:0x7fab7340d080]   (active)>

Make a symbolic breakpoint at UIViewAlertForUnsatisfiableConstraints to catch this in the debugger.
The methods in the UIConstraintBasedLayoutDebugging category on UIView listed in <UIKitCore/UIView.h> may also be helpful.
```

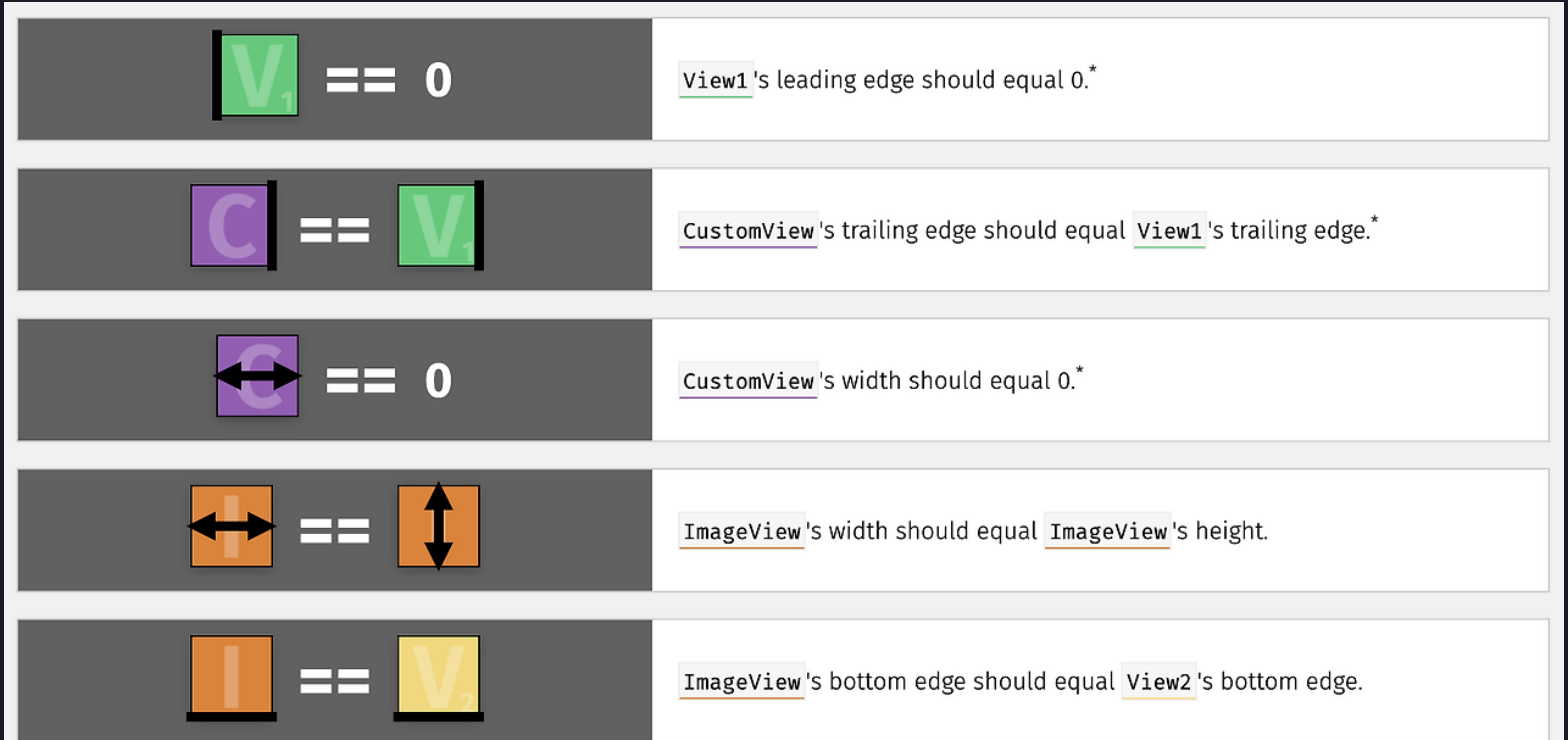
## 원래 UIKit은 이랬어요!



Paste your constraint error log here:

```
Paste your constraint error logs: just the part surrounded by ( and )
```

## 원래 UIKit은 이랬어요!



View1's leading edge should equal 0.*

CustomView's trailing edge should equal View1's trailing edge.*

CustomView's width should equal 0.*

ImageView's width should equal ImageView's height.

ImageView's bottom edge should equal View2's bottom edge.

## SwiftUI는 이렇게 해요!

**SwiftUI는 선언형 방식으로 레이아웃을 정의합니다.**

앗 스유는 프레임워크가 알아서 다 해준다!

## SwiftUI는 이렇게 해요!

**특히!**

부모 뷰와 자식 뷰 간의 상호작용이 중요합니다!

## SwiftUI는 이렇게 해요!

### >>> 3줄 요약 <<<

1. 부모 뷰는 자식 뷰에게 제공 가능한 사이즈를 전달
2. 자식 View는 원하는 크기만큼 크기를 정함
3. 부모가 자식의 위치를 정해서 배치함 (기본적으로 가운데 위치)

## SwiftUI는 이렇게 해요!

### >>> 3줄 요약 <<<

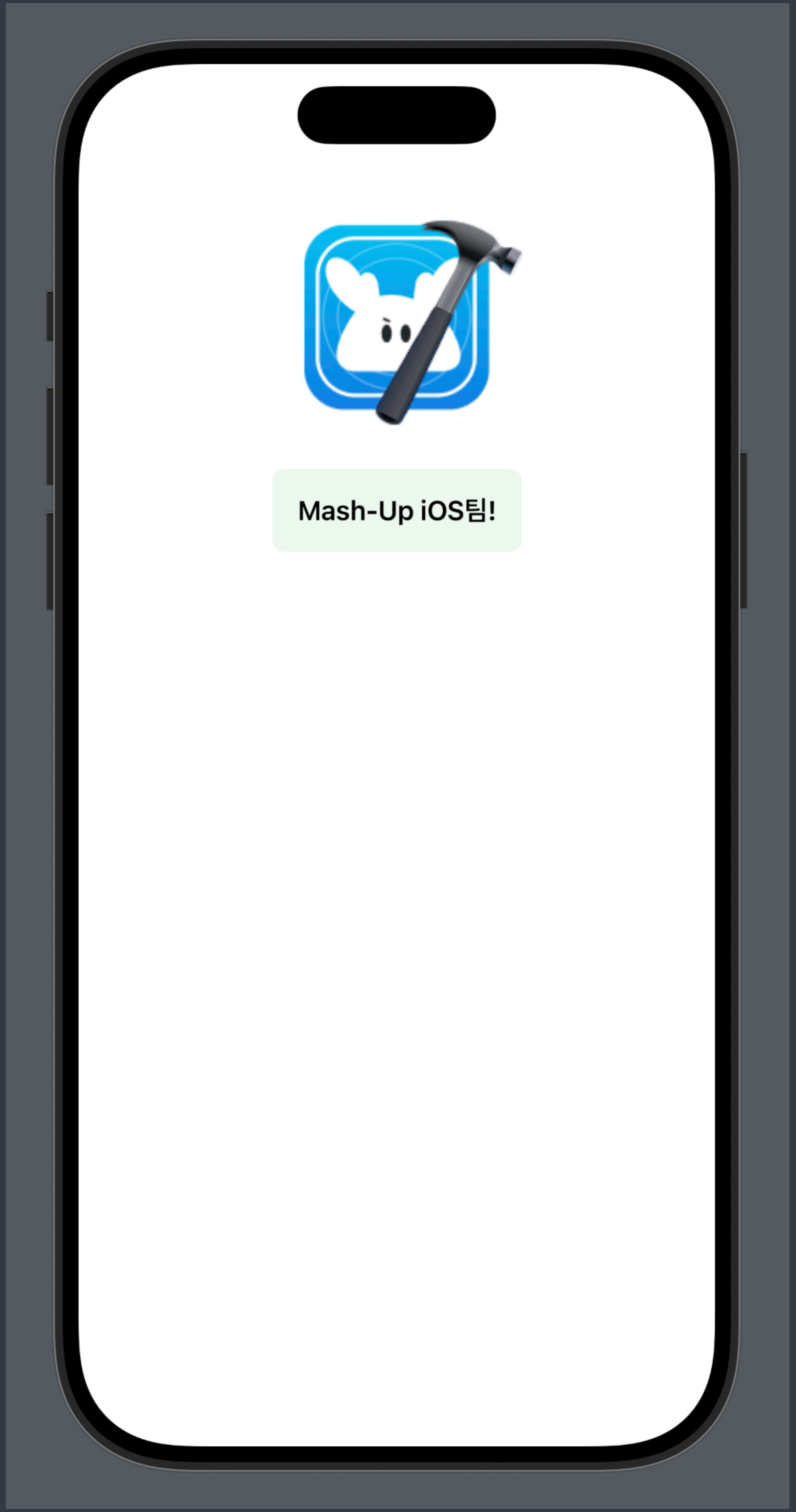
1. 부모 뷰는 자식 뷰에게 제공 가능한 사이즈를 전달
2. 자식 View는 원하는 크기만큼 크기를 정해서 부모 뷰에게 전달
3. 부모가 자식의 위치를 정해서 배치함 (기본적으로 가운데 위치)

## SwiftUI는 이렇게 해요!

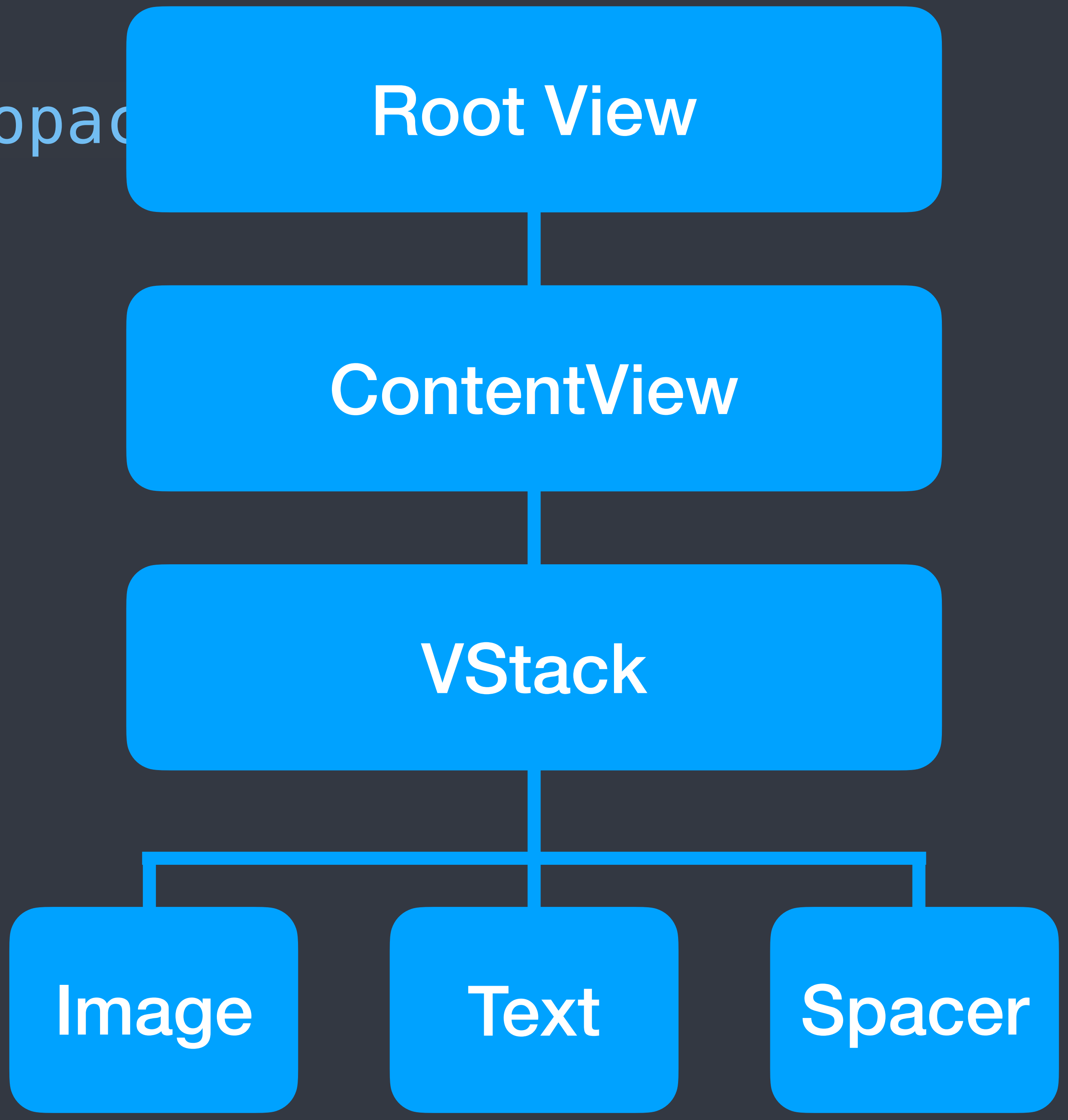
### >>> 3줄 요약 <<<

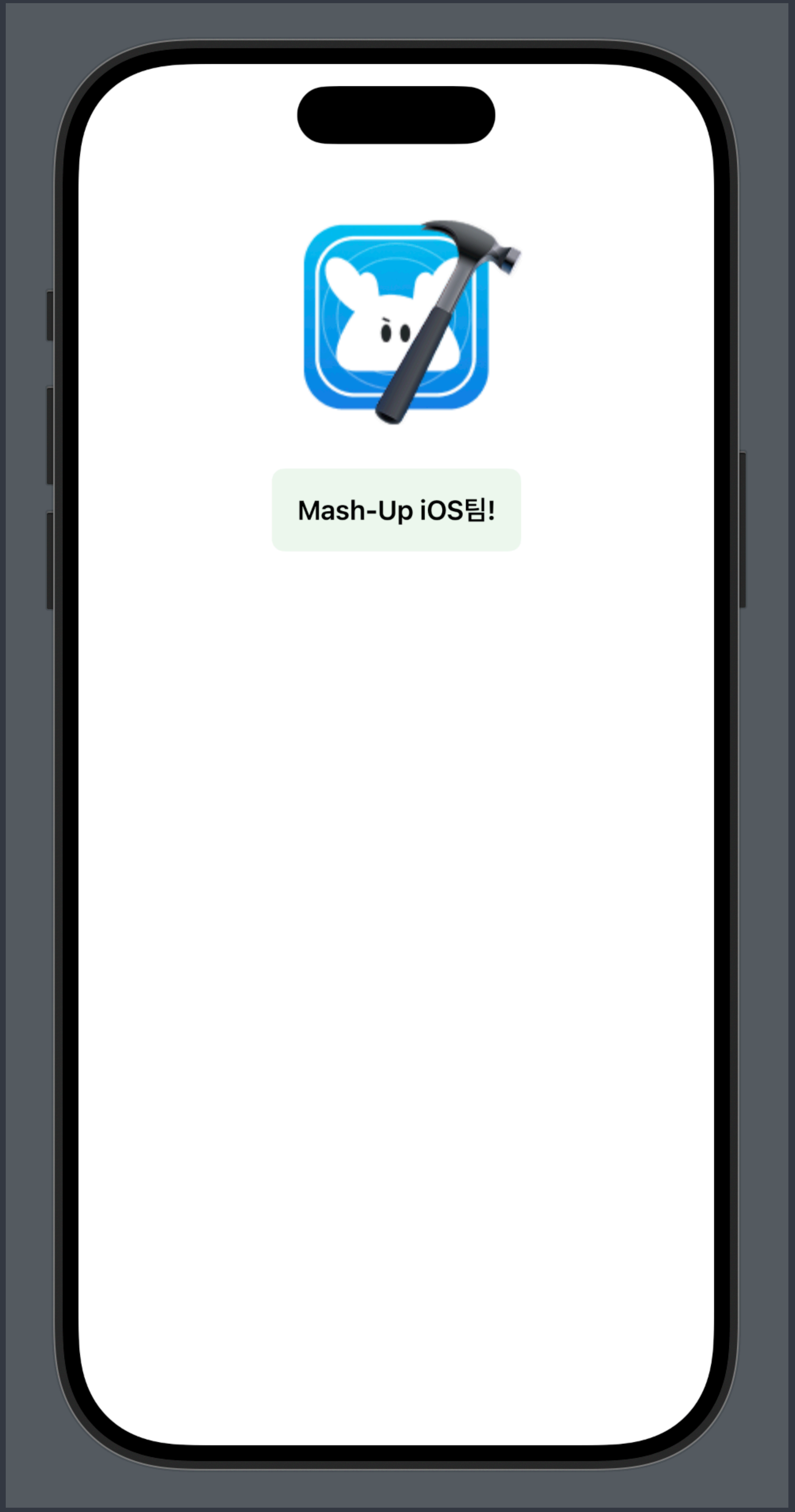
1. 부모 뷰는 자식 뷰에게 제공 가능한 사이즈를 전달
2. 자식 View는 원하는 크기만큼 크기를 정함
3. 부모가 자식의 위치를 정해서 배치함 (기본적으로 가운데 위치)

```swift
struct ContentView: View {
    var body: some View {
        VStack(spacing: 16) {
            Image(ImageResource.mashongCode)

            Text("Mash-Up iOS팀!")
                .font(.headline)
                .padding()
                .background(Color.green.opacity(0.1))
                .cornerRadius(8)

            Spacer()
        }
        .padding()
    }
}
```

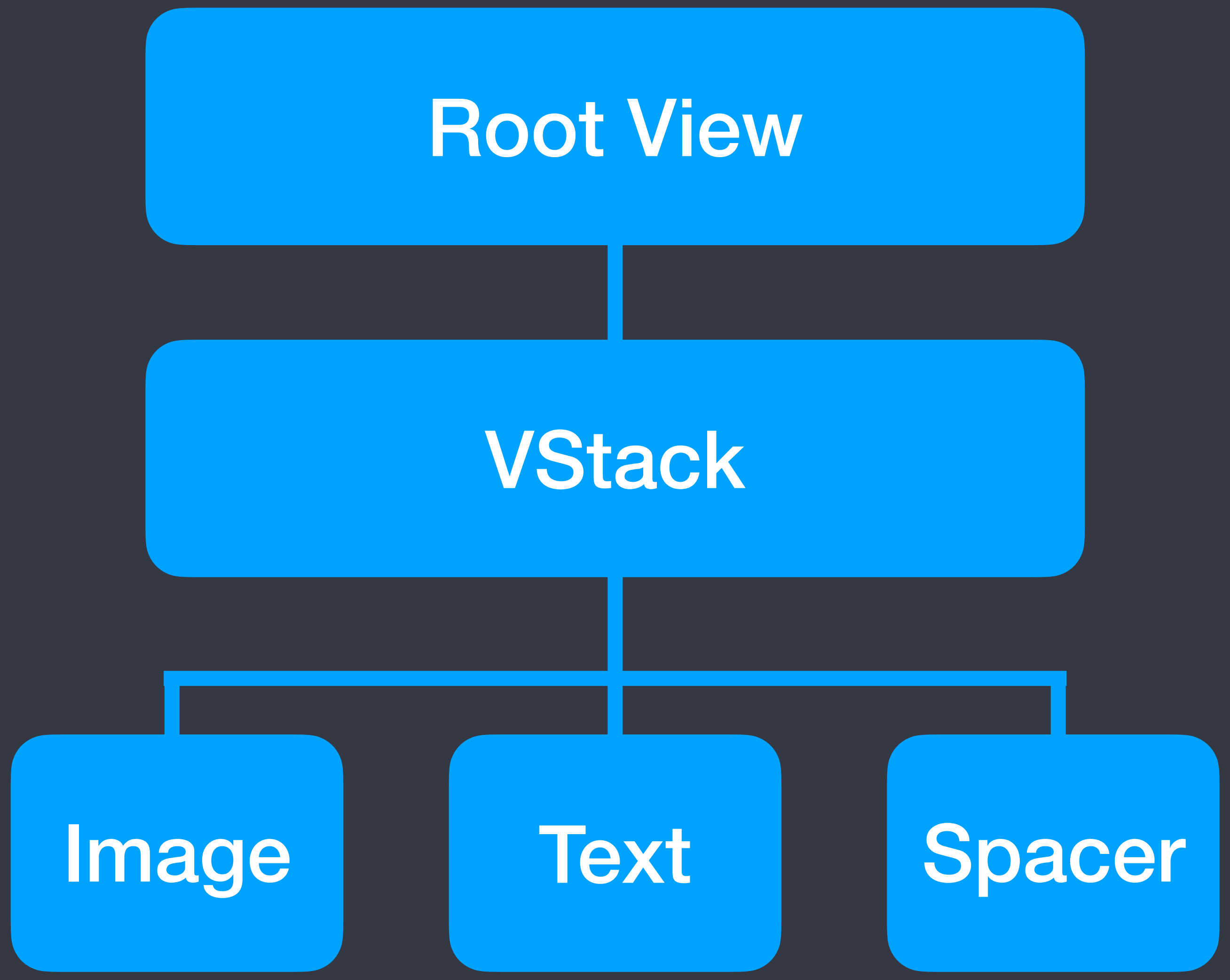


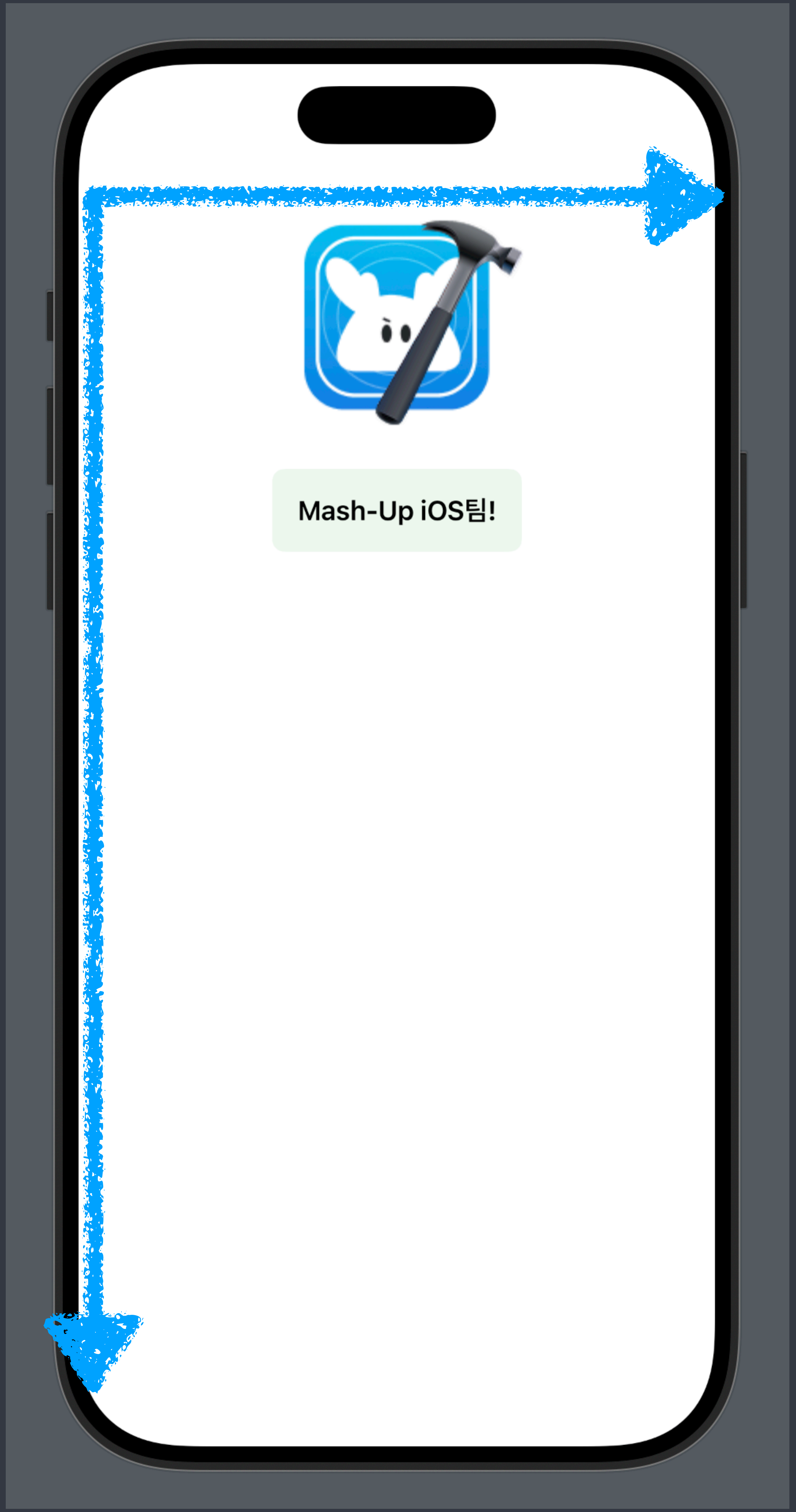
```swift
struct ContentView: View {
    var body: some View {
        VStack(spacing: 16) {
            Image(ImageResource.mashongCode)

            Text("Mash-Up iOS팀!")
                .font(.headline)
                .padding()
                .background(Color.green.opac
                .cornerRadius(8)

            Spacer()
        }
        .padding()
    }
}
```





```swift
struct ContentView: View {
    var body: some View {
        VStack(spacing: 16) {
            Image(ImageResource.mashongCode)

            Text("Mash-Up iOS팀!")
                .font(.headline)
                .padding()
                .background(Color.green.opacity(0.1))
                .cornerRadius(8)

            Spacer()
        }
        .padding()
    }
}
```

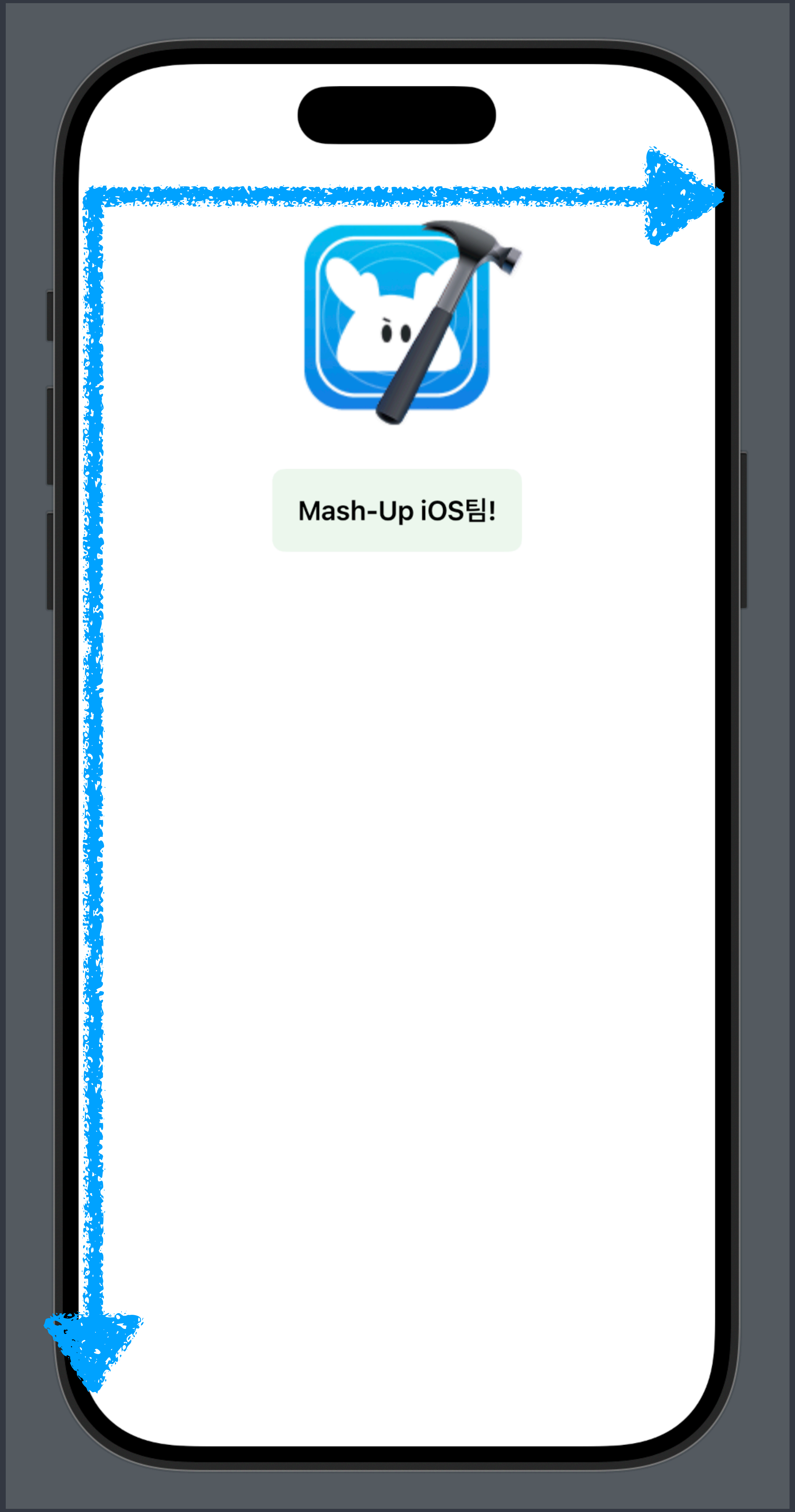




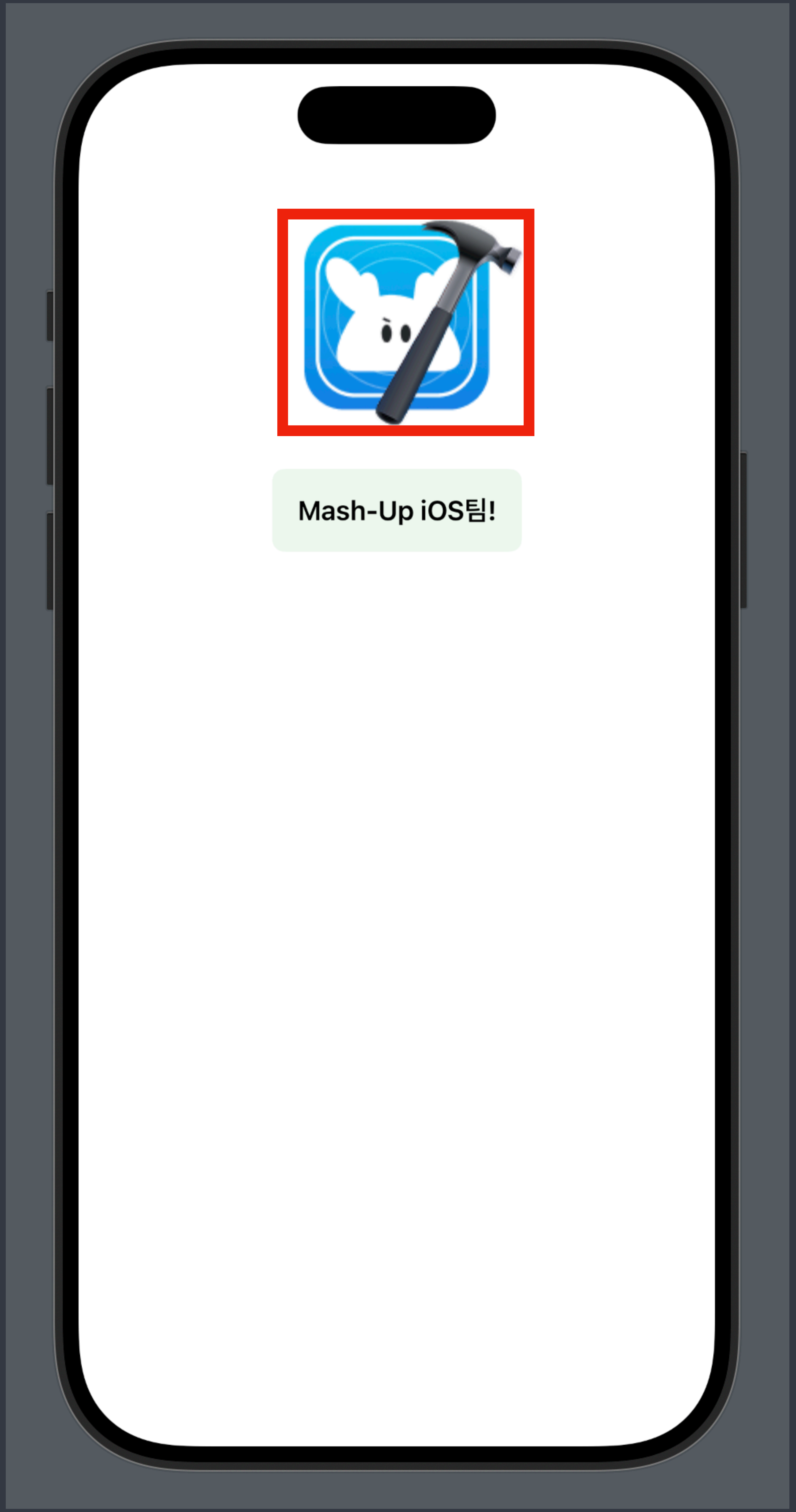
## SwiftUI는 이렇게 해요!

1. 가장 유연하지 않은 view: Image(fixed size)
2. 조금 유연한 view: Text(fit its text)
3. 매우 유연한 view: RoundedRectangle(any space offered), Color

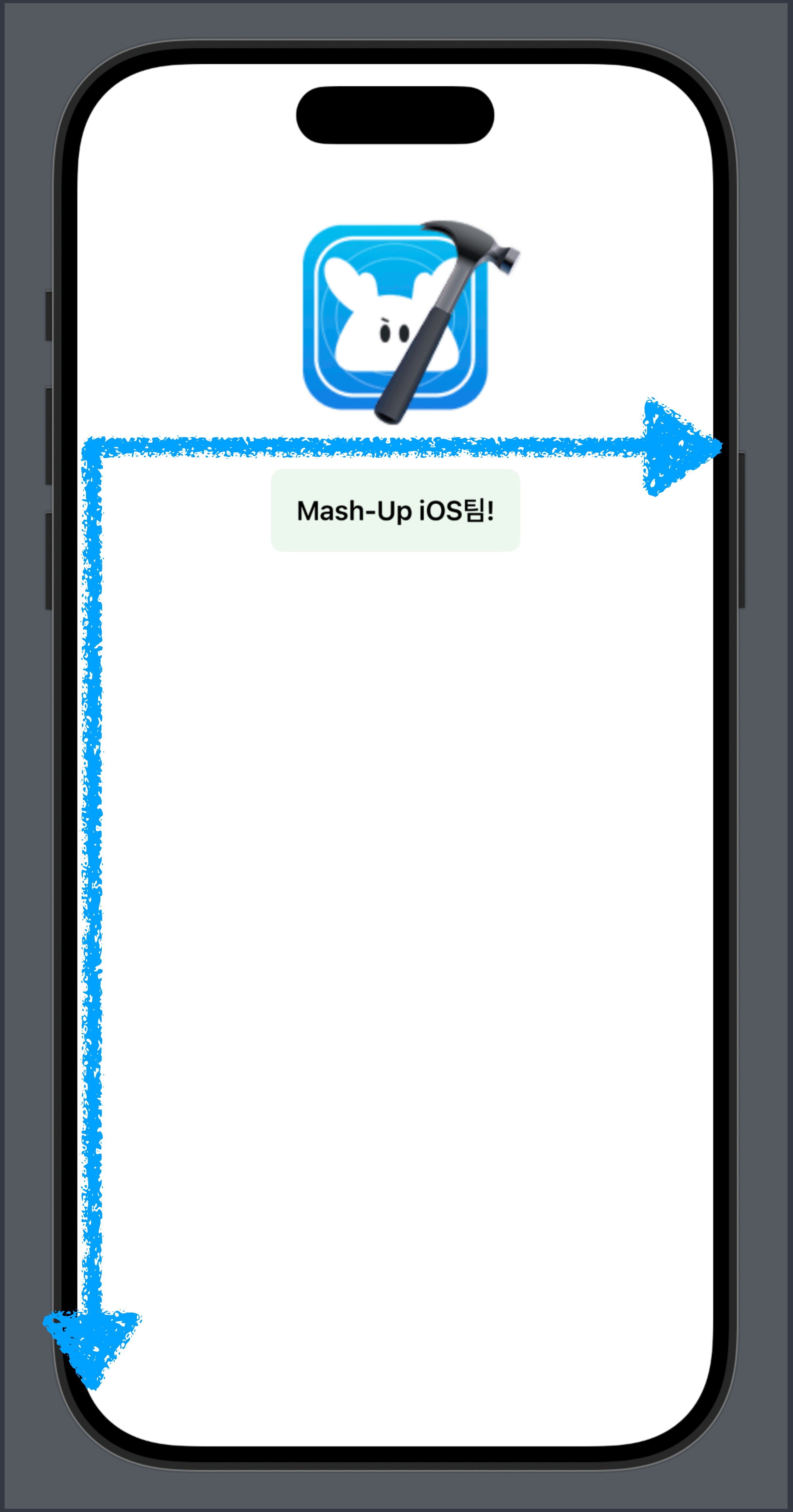
```swift
struct ContentView: View {
    var body: some View {
        VStack(spacing: 16) {
            Image(ImageResource.mashongCode)

            Text("Mash-Up iOS팀!")
                .font(.headline)
                .padding()
                .background(Color.green.opacity(0.1))
                .cornerRadius(8)

            Spacer()
        }
        .padding()
    }
}
```



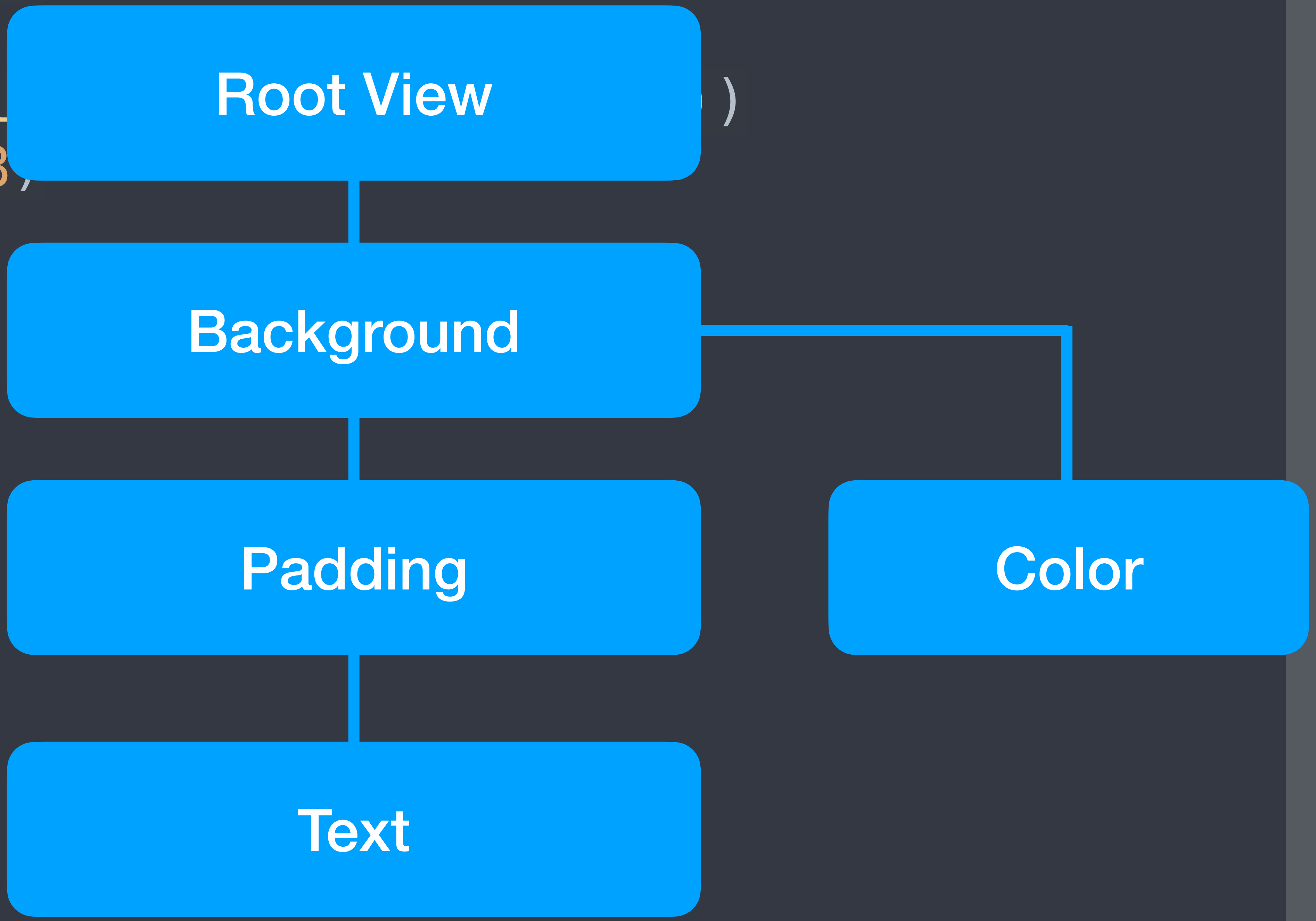
```swift
struct ContentView: View {
    var body: some View {
        VStack(spacing: 16) {
            Image(ImageResource.mashongCode)

            Text("Mash-Up iOS팀!")
                .font(.headline)
                .padding()
                .background(Color.green.opacity(0.1))
                .cornerRadius(8)

            Spacer()
        }
        .padding()
    }
}
```

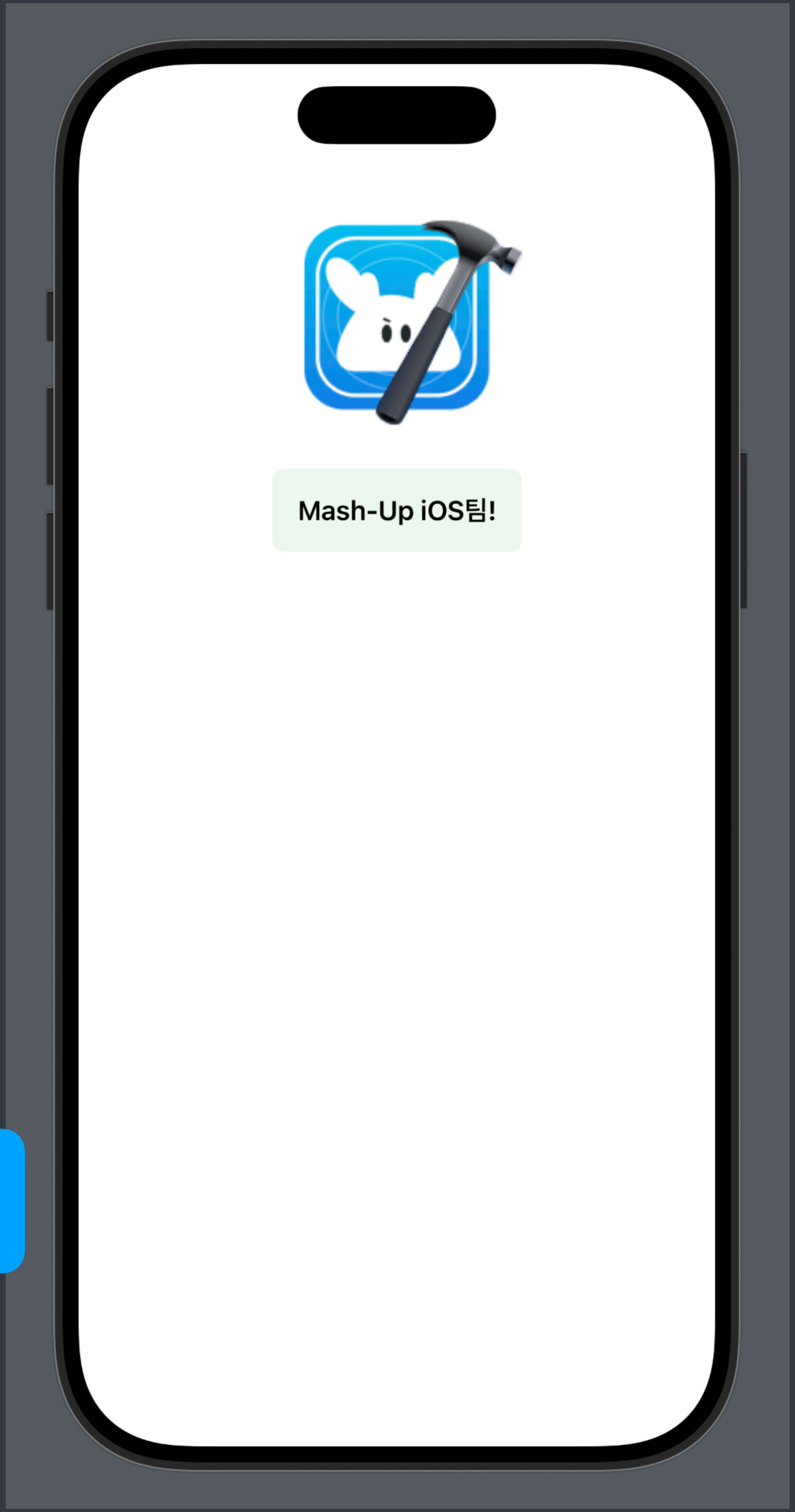


```swift
struct ContentView: View {
    var body: some View {
        VStack(spacing: 16) {
            Image(ImageResource.mashongCode)
            Text("Mash-Up iOS팀!")
                .font(.headline)
                .padding()
                .background(Color.green.opacity(0.1))
                .cornerRadius(8)
            Spacer()
        }
        .padding()
    }
}
```

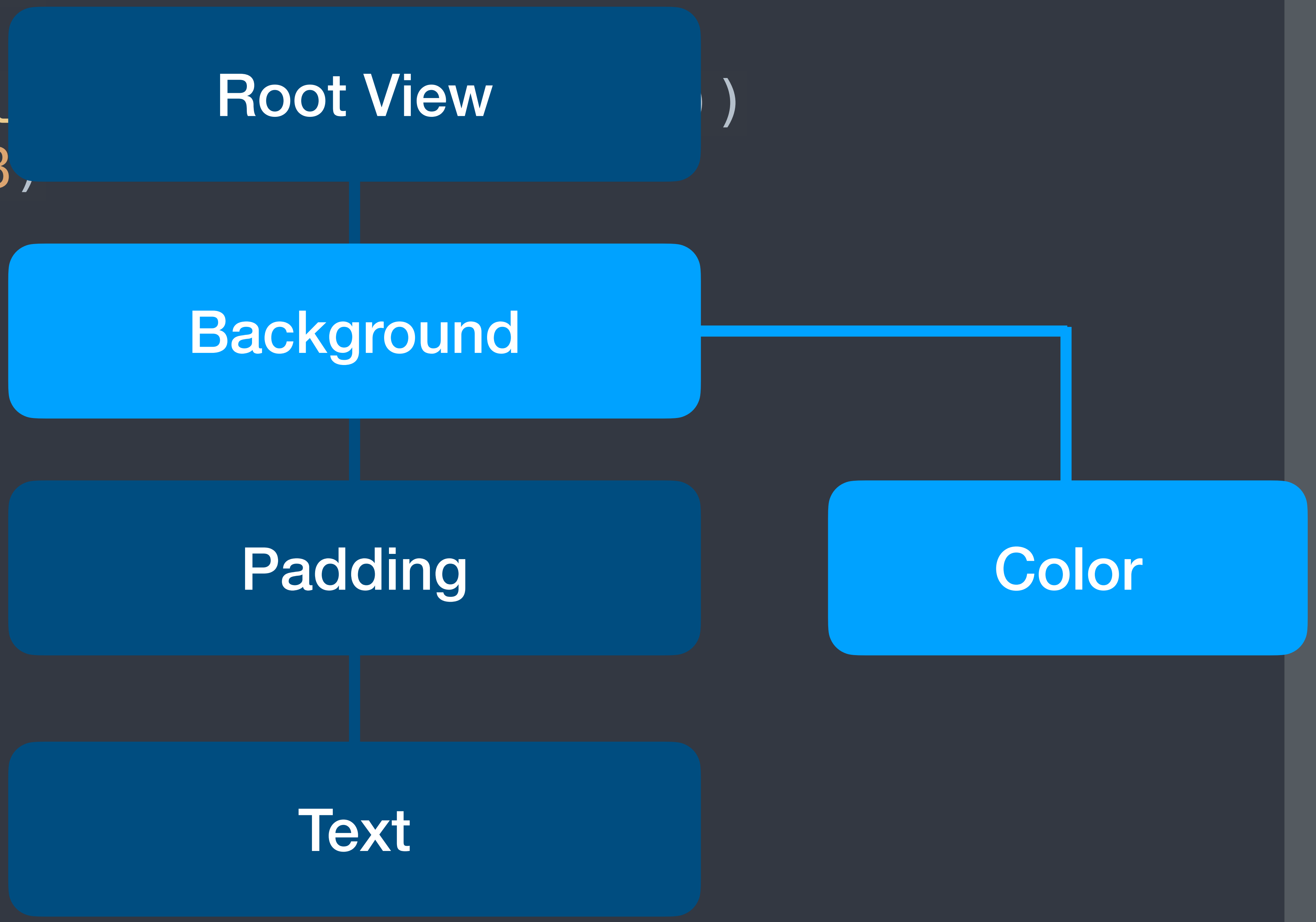


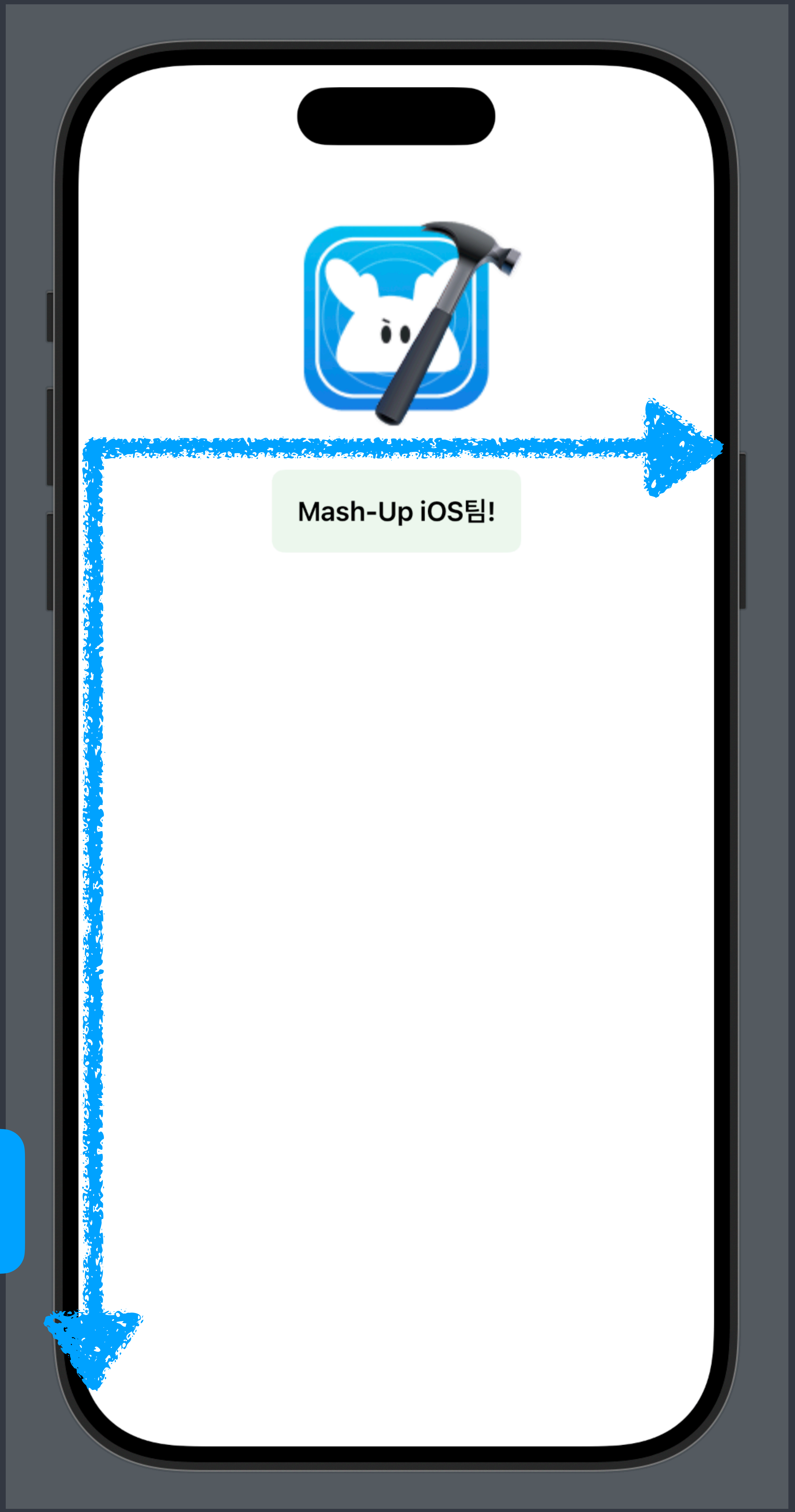
```swift
struct ContentView: View {
    var body: some View {
        VStack(spacing: 16) {
            Image(ImageResource.mashongCode)

            Text("Mash-Up iOS팀!")
                .font(.headline)
                .padding()
                .background(Col                    )
                .cornerRadius(8

            Spacer()
        }
        .padding()
    }
}
```



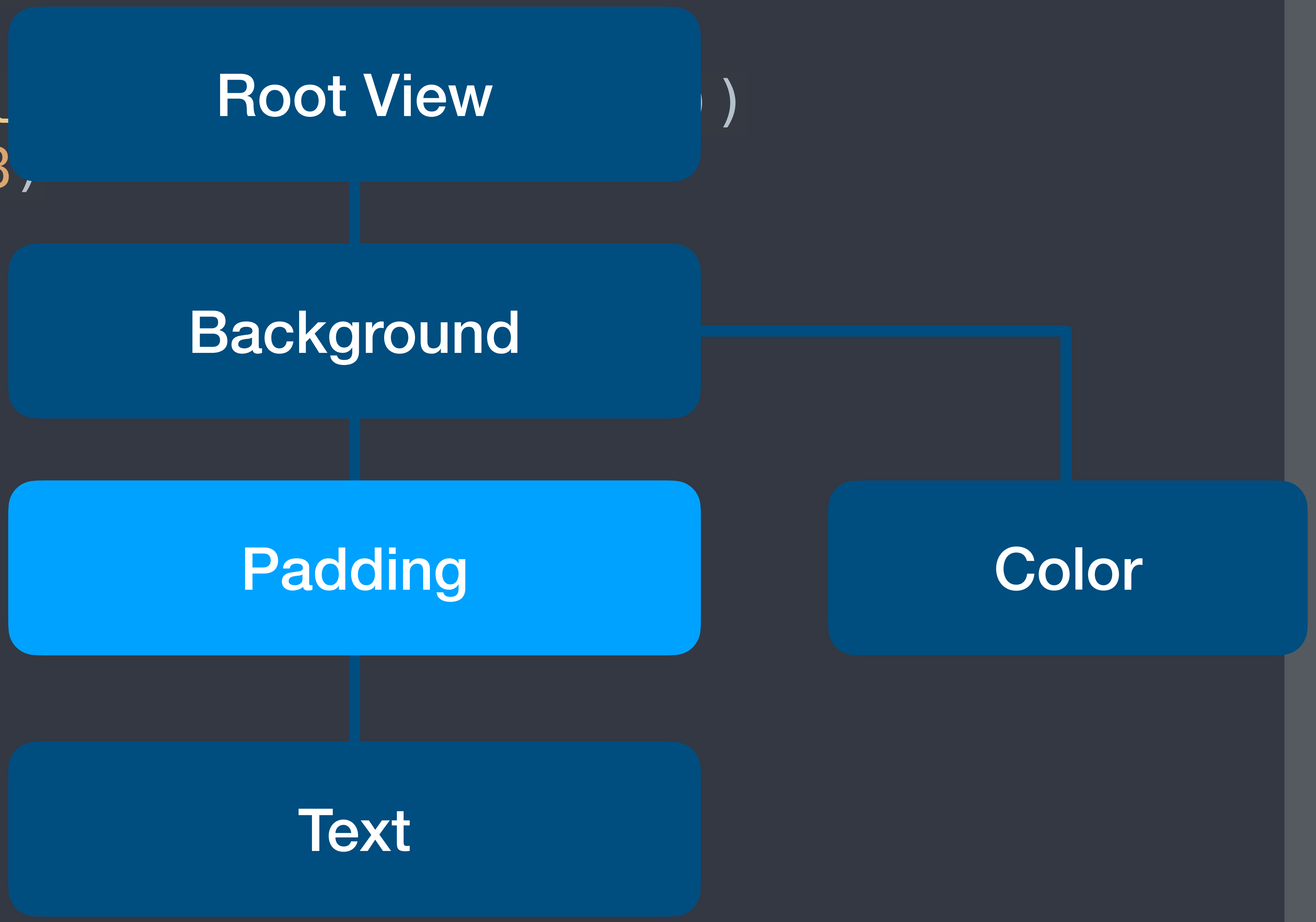


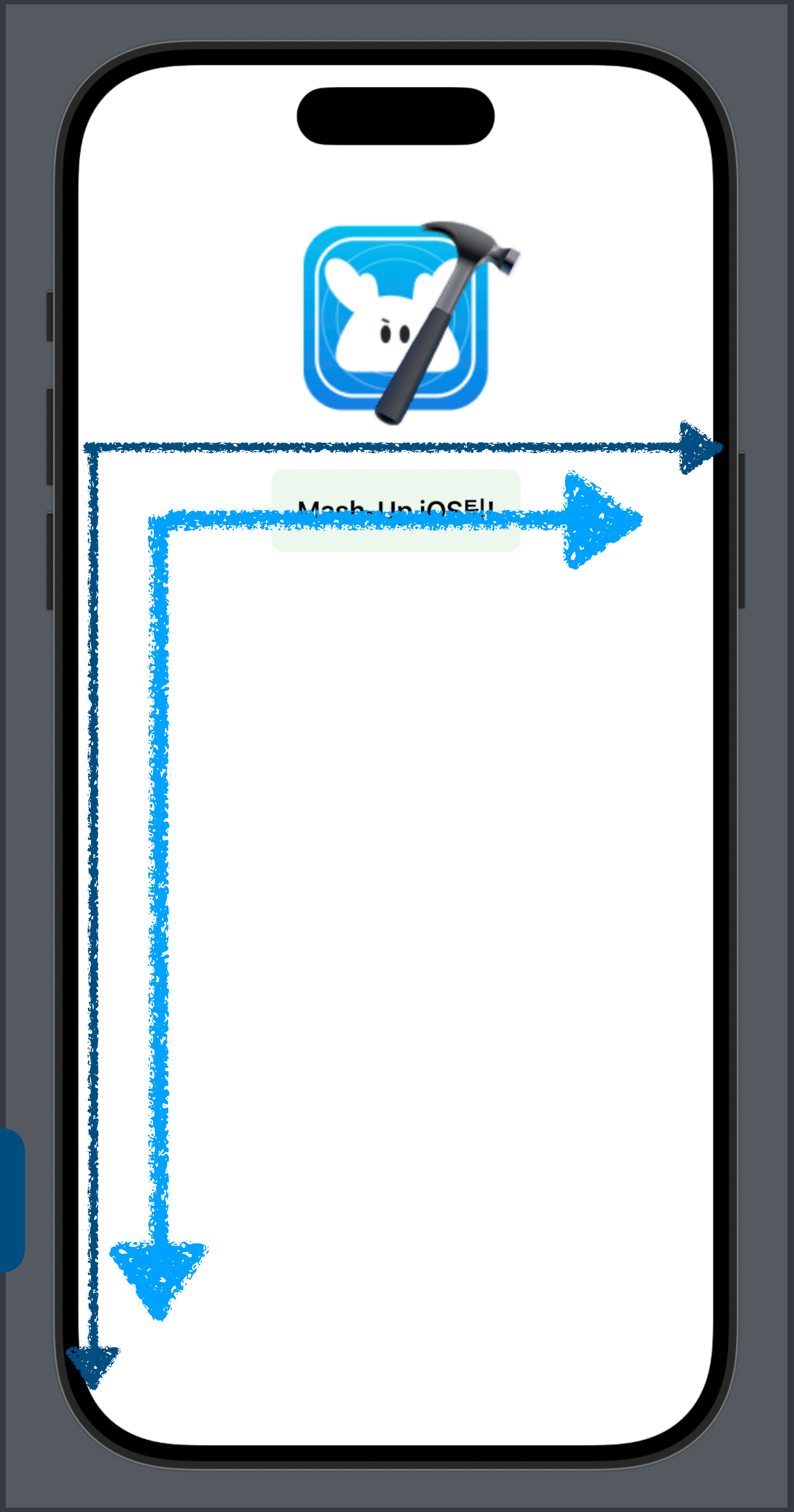
```swift
struct ContentView: View {
    var body: some View {
        VStack(spacing: 16) {
            Image(ImageResource.mashongCode)

            Text("Mash-Up iOS팀!")
                .font(.headline)
                .padding()
                .background(Col                    )
                .cornerRadius(8,

            Spacer()
        }
        .padding()
    }
}
```



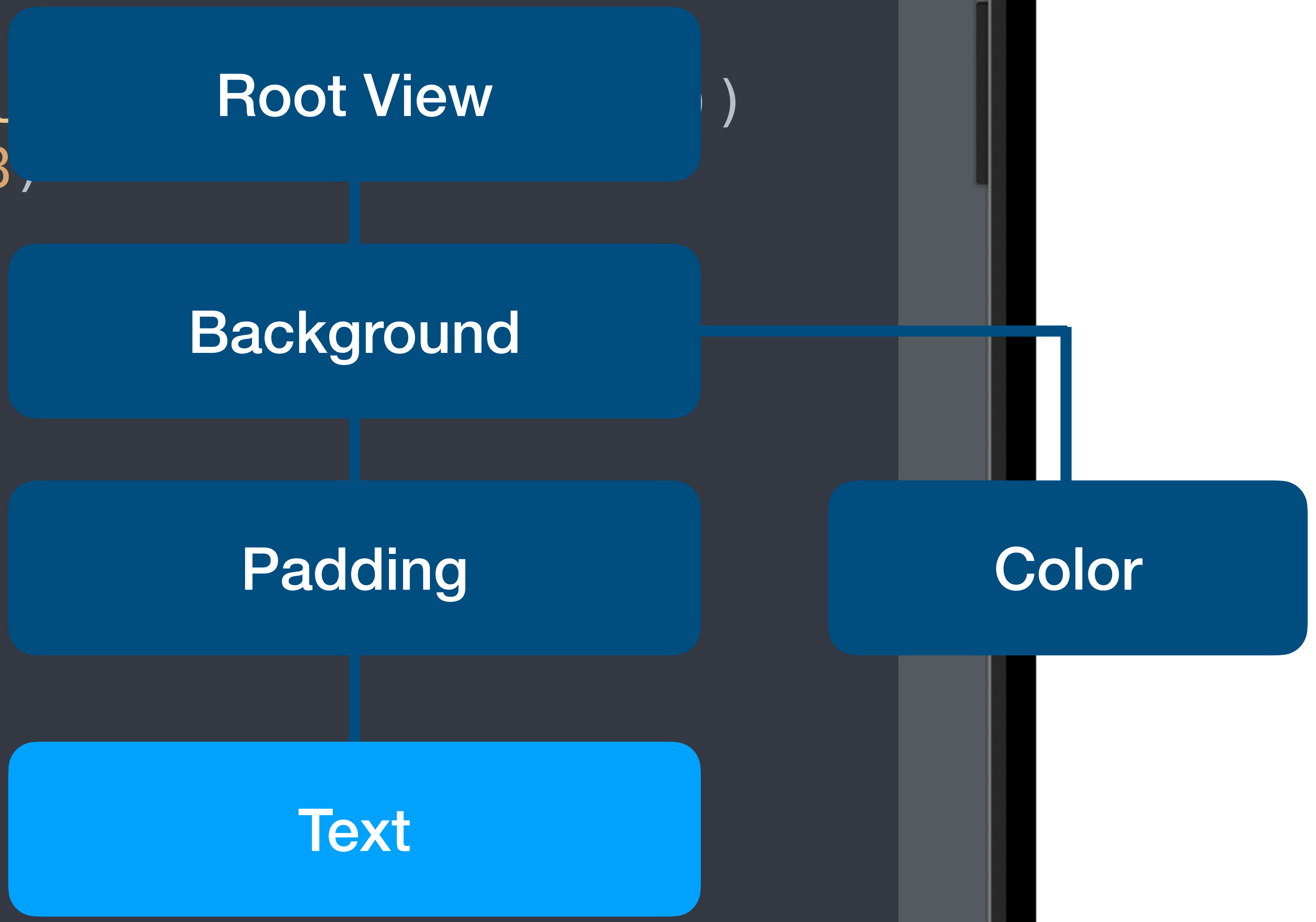


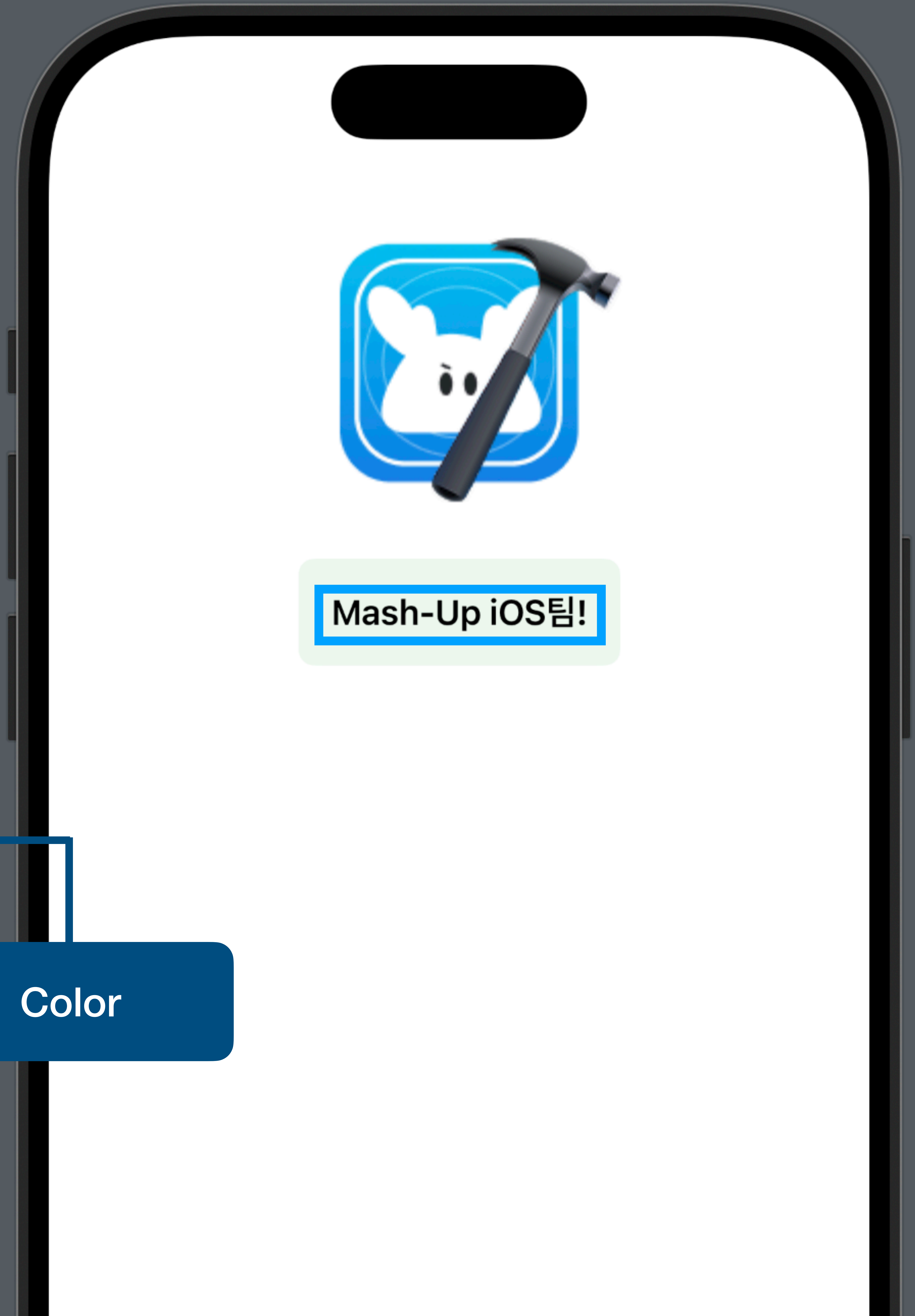
```
struct ContentView: View {
    var body: some View {
        VStack(spacing: 16) {
            Image(ImageResource.mashongCode)

            Text("Mash–Up iOS팀!")
                .font(.headline)
                .padding()
                .background(Col                    )
                .cornerRadius(8,

            Spacer()
        }
        .padding()
    }
}
```



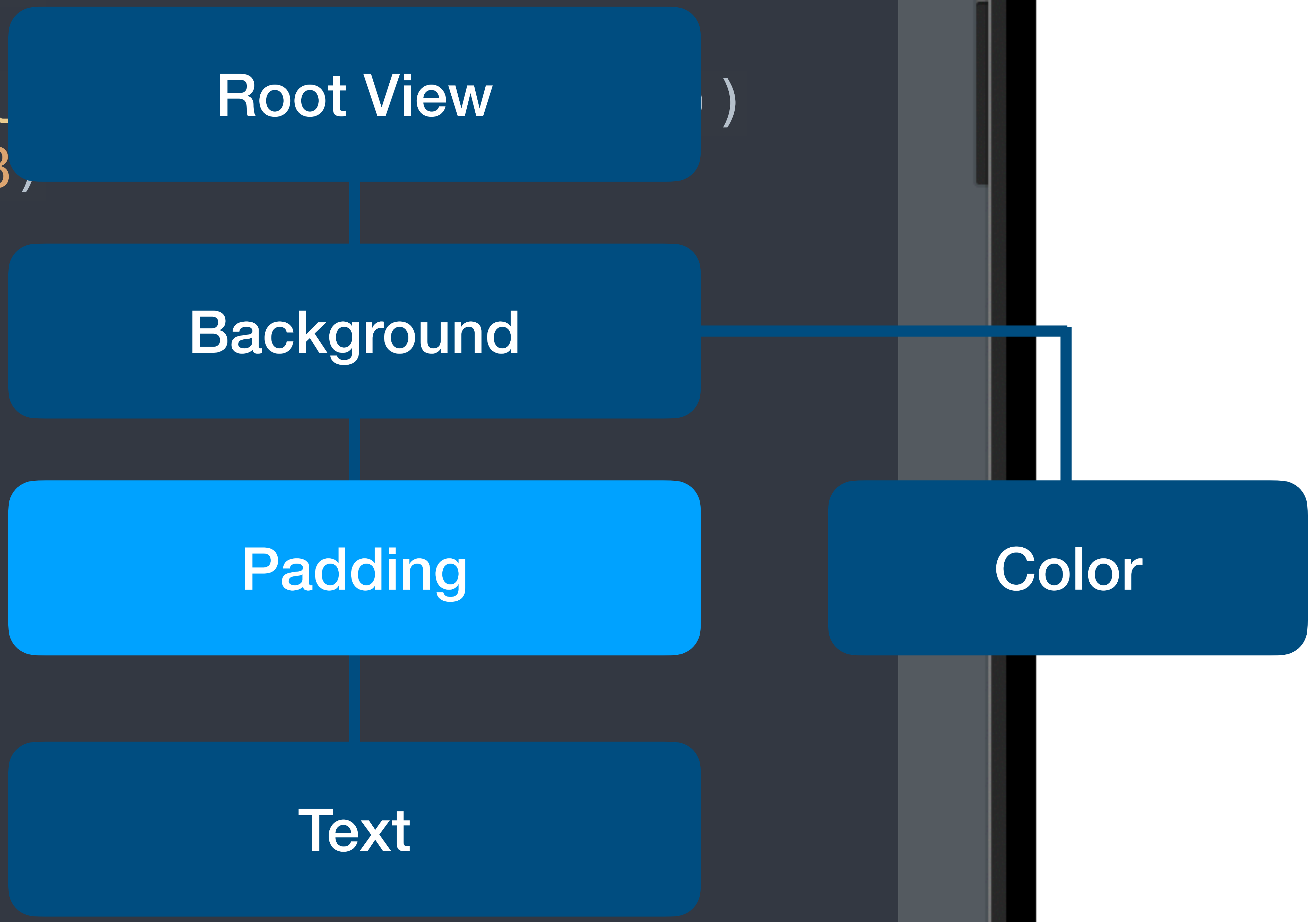


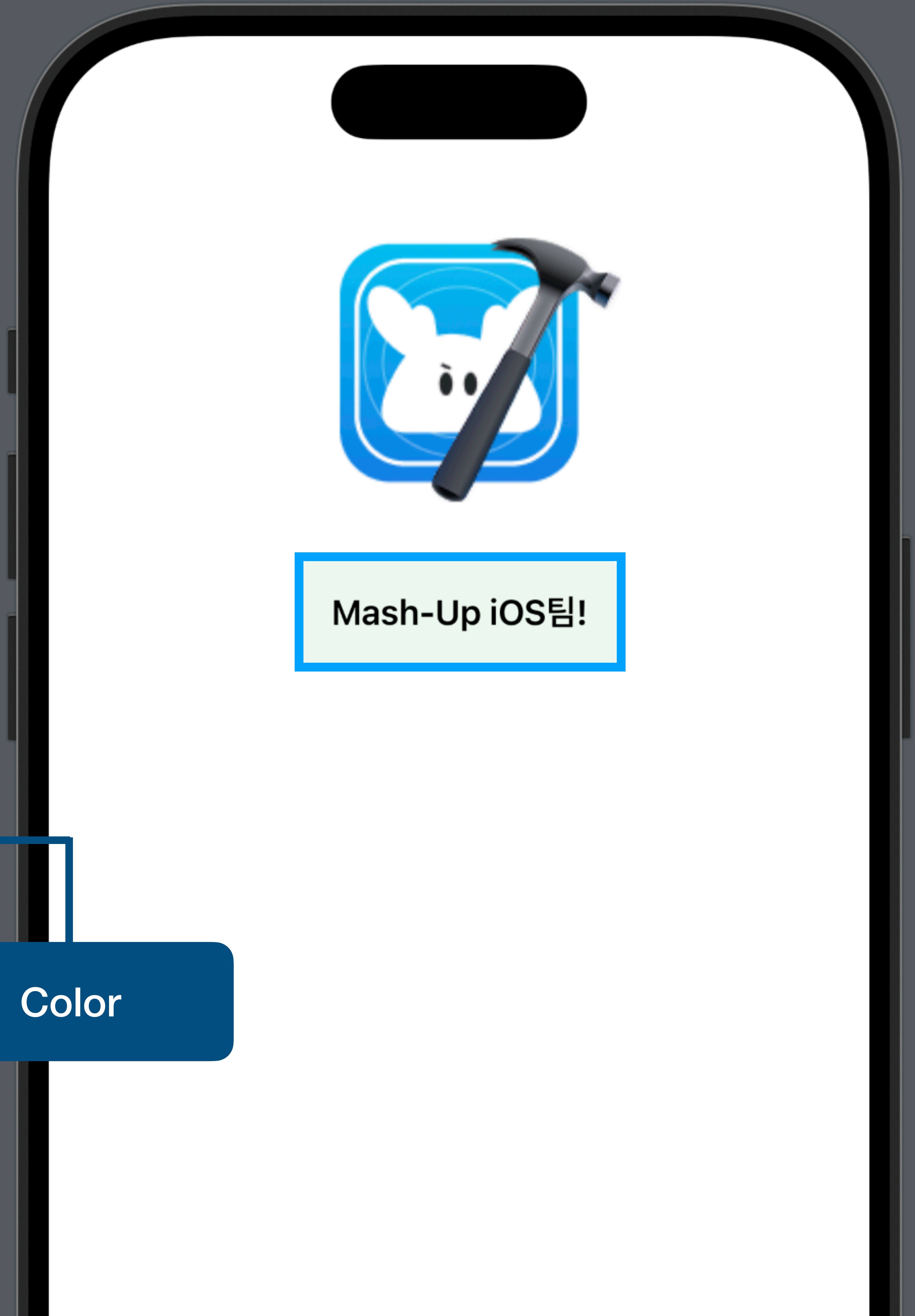
```swift
struct ContentView: View {
    var body: some View {
        VStack(spacing: 16) {
            Image(ImageResource.mashongCode)

            Text("Mash-Up iOS팀!")
                .font(.headline)
                .padding()
                .background(Col                        )
                .cornerRadius(8

            Spacer()
        }
        .padding()
    }
}
```



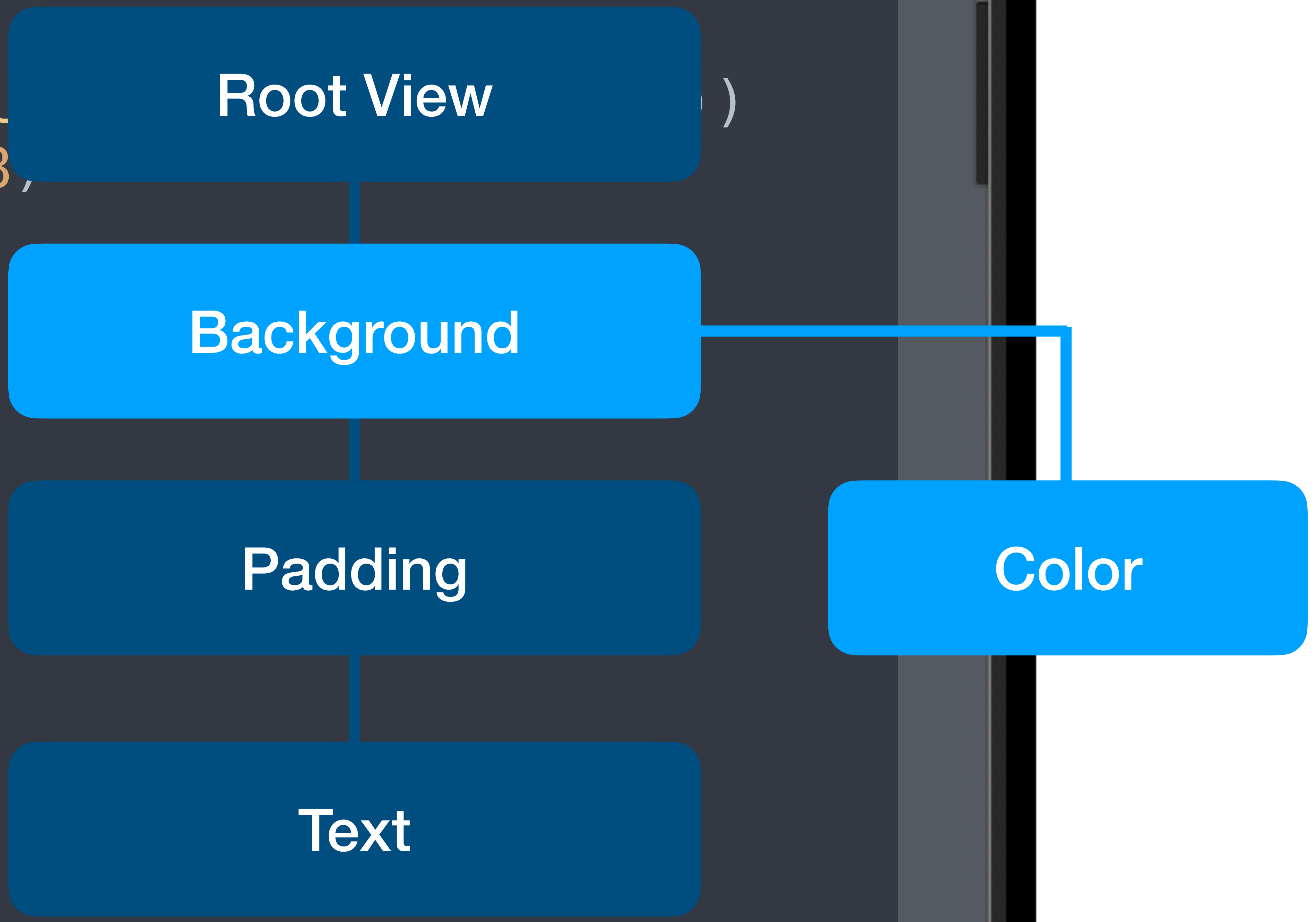


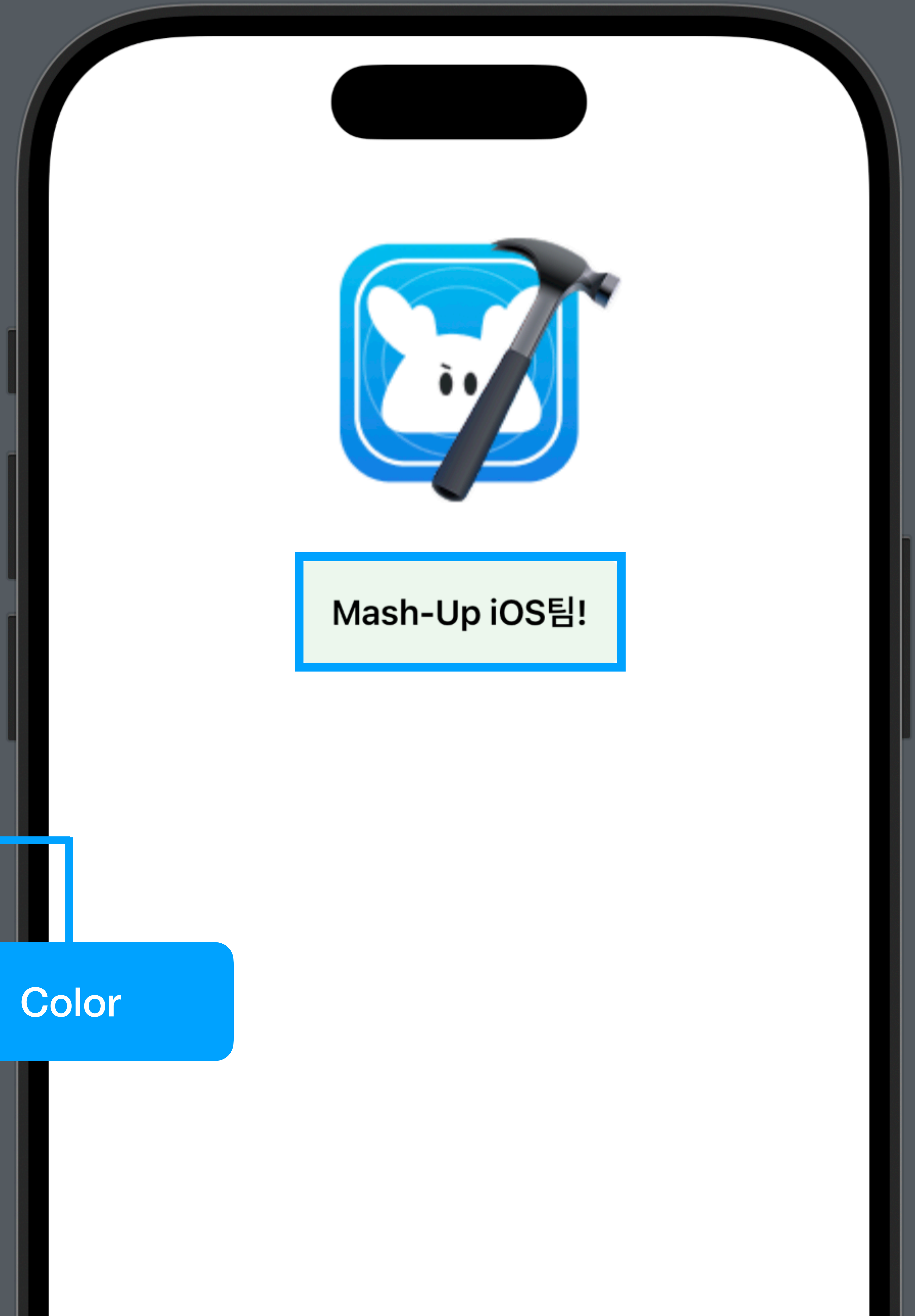
```swift
struct ContentView: View {
    var body: some View {
        VStack(spacing: 16) {
            Image(ImageResource.mashongCode)

            Text("Mash-Up iOS팀!")
                .font(.headline)
                .padding()
                .background(Col                        )
                .cornerRadius(8,

            Spacer()
        }
        .padding()
    }
}
```



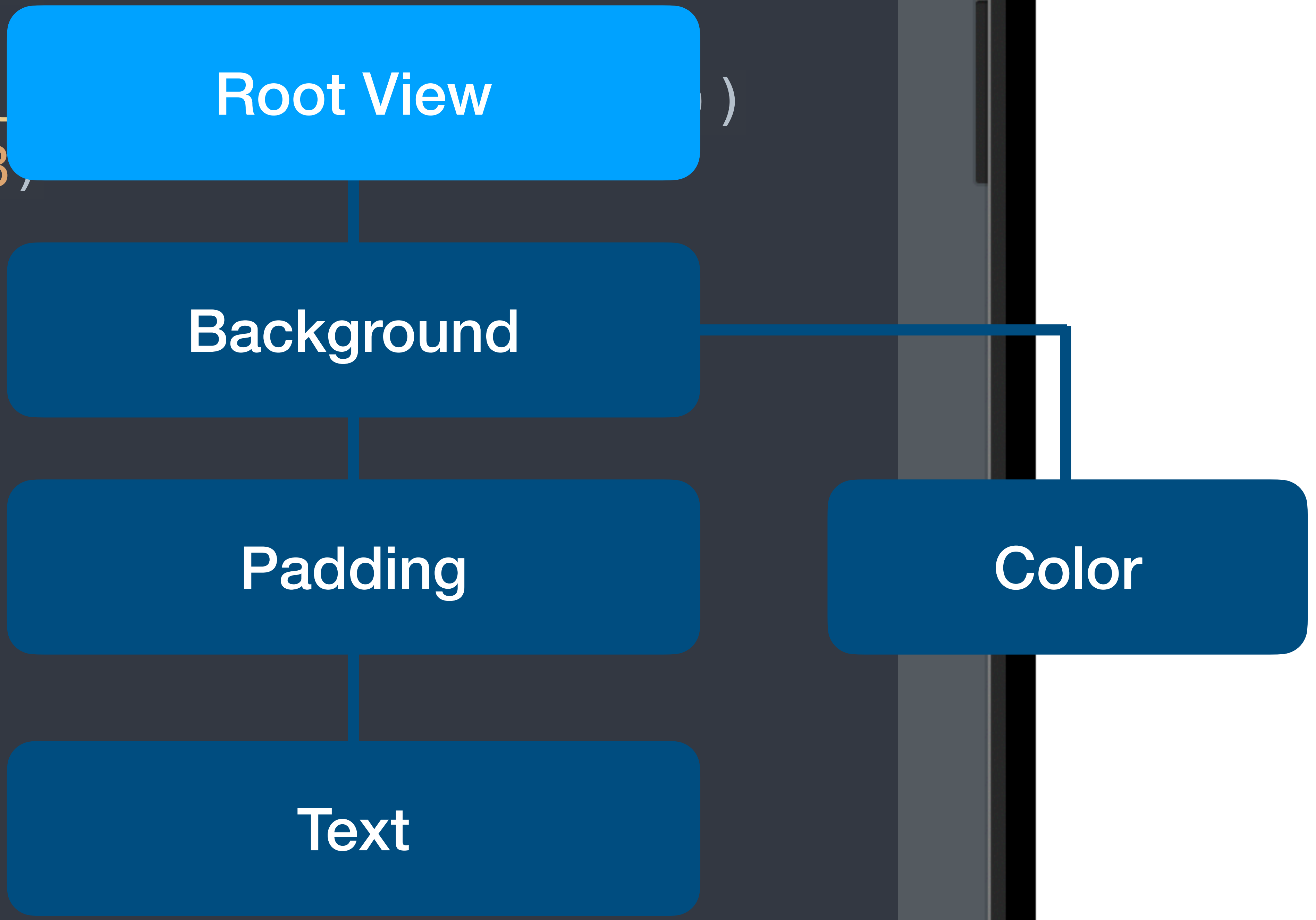


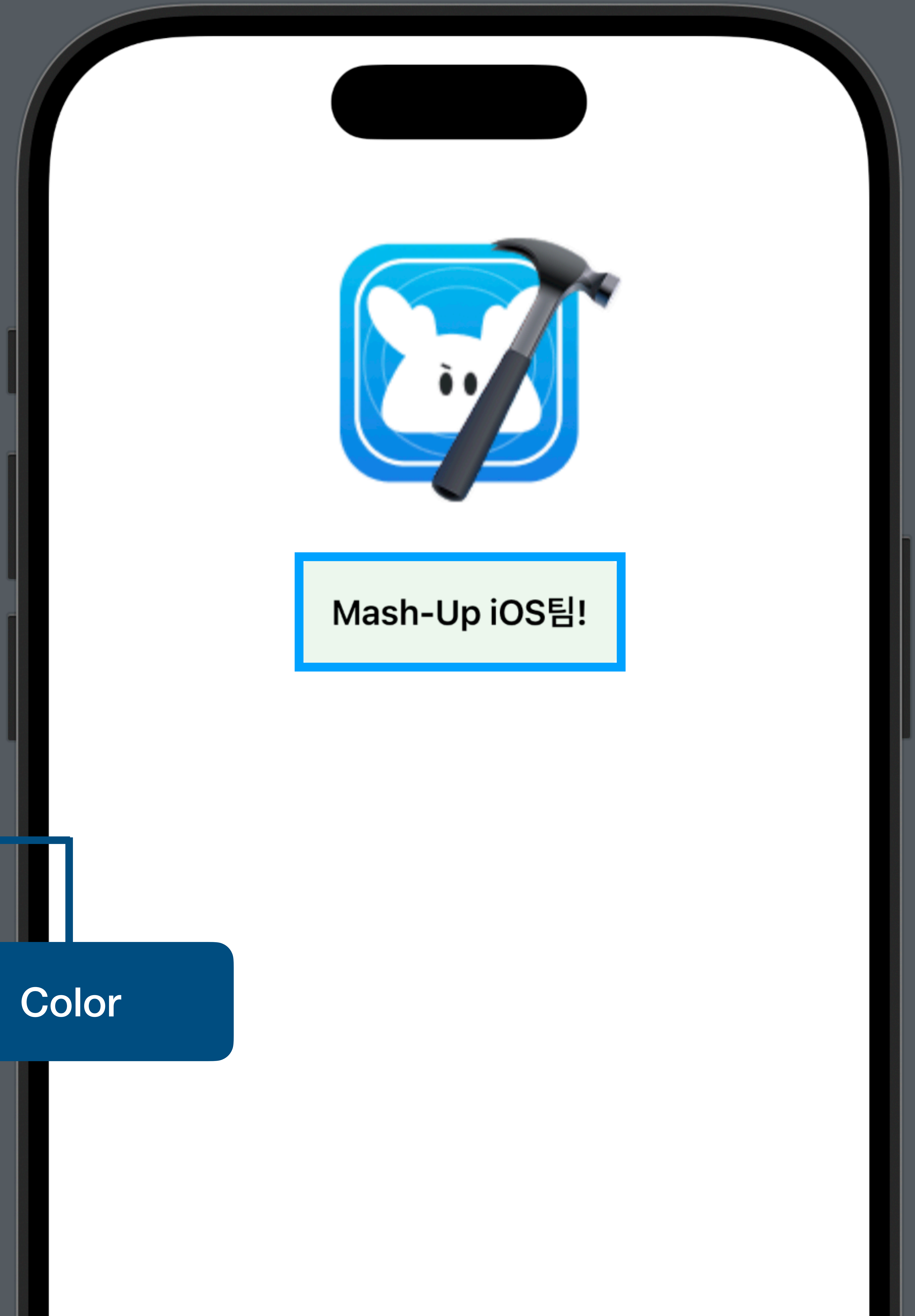
```swift
struct ContentView: View {
    var body: some View {
        VStack(spacing: 16) {
            Image(ImageResource.mashongCode)

            Text("Mash-Up iOS팀!")
                .font(.headline)
                .padding()
                .background(Col                    )
                .cornerRadius(8,

            Spacer()
        }
        .padding()
    }
}
```



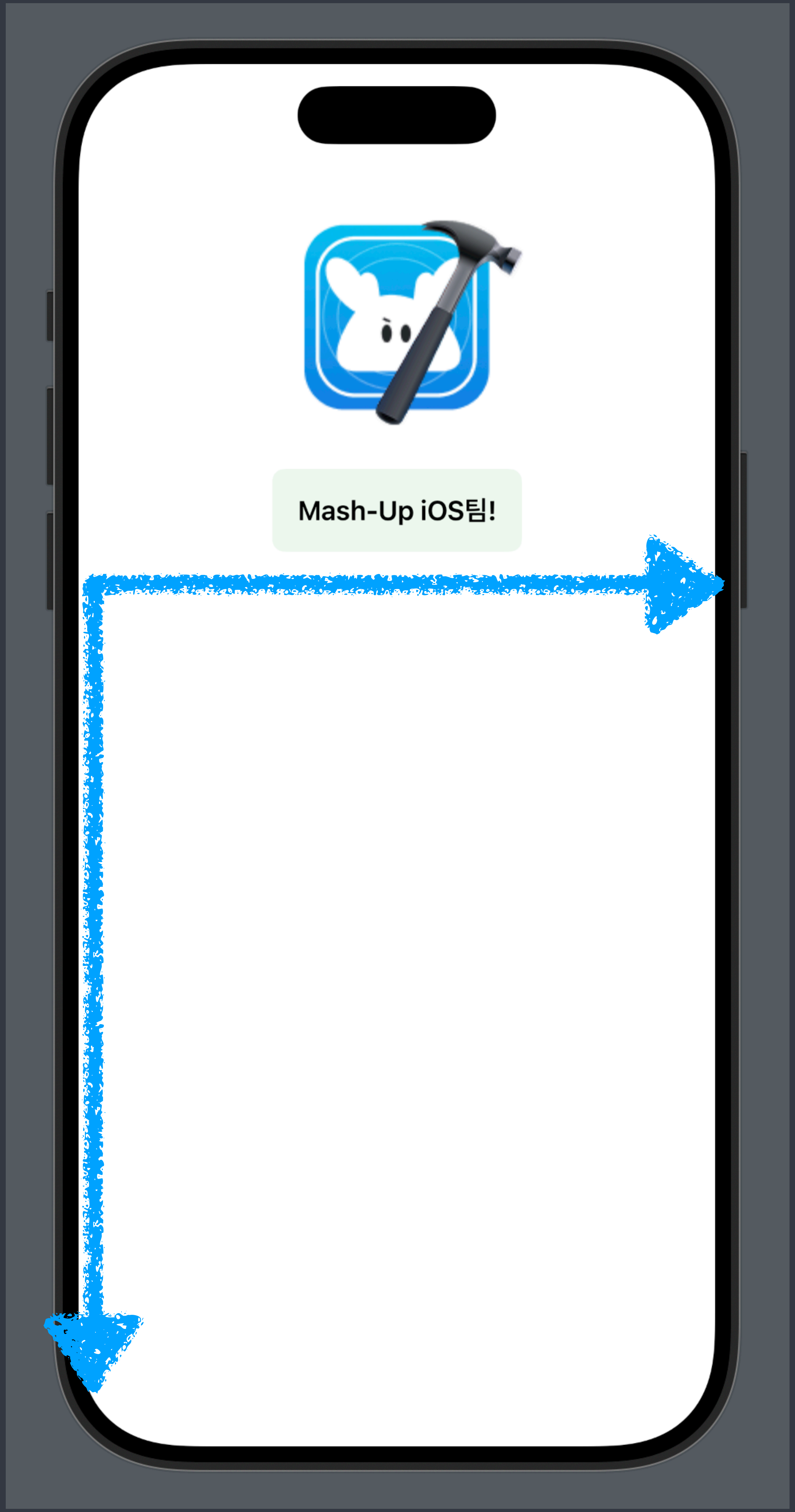


```swift
struct ContentView: View {
    var body: some View {
        VStack(spacing: 16) {
            Image(ImageResource.mashongCode)

            Text("Mash-Up iOS팀!")
                .font(.headline)
                .padding()
                .background(Col                    )
                .cornerRadius(8)

            Spacer()
        }
        .padding()
    }
}
```





```swift
struct ContentView: View {
    var body: some View {
        VStack(spacing: 16) {
            Image(ImageResource.mashongCode)

            Text("Mash-Up iOS팀!")
                .font(.headline)
                .padding()
                .background(Color.green.opacity(0.1))
                .cornerRadius(8)

            Spacer()
        }
        .padding()
    }
}
```



```swift
class ViewController: UIViewController {

    override func viewDidLoad() {
        super.viewDidLoad()
        setupViews()
    }

    private func setupViews() {
        // 배경색 설정
        view.backgroundColor = .white

        // 스택 뷰 생성 (VStack 역할)
        let stackView = UIStackView()
        stackView.axis = .vertical
        stackView.spacing = 16
        stackView.translatesAutoresizingMaskIntoConstraints = false
        view.addSubview(stackView)

        // 스택 뷰의 패딩 (padding() 역할)
        NSLayoutConstraint.activate([
            stackView.topAnchor.constraint(equalTo: view.safeAreaLayoutGuide.topAnchor, constant: 16),
            stackView.leadingAnchor.constraint(equalTo: view.safeAreaLayoutGuide.leadingAnchor, constant: 16),
            stackView.trailingAnchor.constraint(equalTo: view.safeAreaLayoutGuide.trailingAnchor, constant: -16),
            stackView.bottomAnchor.constraint(equalTo: view.safeAreaLayoutGuide.bottomAnchor, constant: -16)
        ])

        // 첫 번째 텍스트 컨테이너 생성
        let mashupContainer = createLabelContainer(withText: "Mash-Up", backgroundColor: UIColor.systemBlue.withAlphaComponent(0.1))

        // 두 번째 텍스트 컨테이너 생성
        let iOSTeamContainer = createLabelContainer(withText: "iOS팀!", backgroundColor: UIColor.systemGreen.withAlphaComponent(0.1))

        // 두 번째 레이블에 leading 정렬 적용 (SwiftUI의 .frame(maxWidth: .infinity, alignment: .leading))
        if let iOSTeamLabel = iOSTeamContainer.subviews.first as? UILabel {
            iOSTeamLabel.textAlignment = .left

            // 컨테이너의 너비를 늘리기 위한 제약조건 (SwiftUI의 maxWidth: .infinity 역할)
            iOSTeamContainer.setContentHuggingPriority(.defaultLow, for: .horizontal)
        }

        // 스페이서 생성 (Spacer() 역할)
```

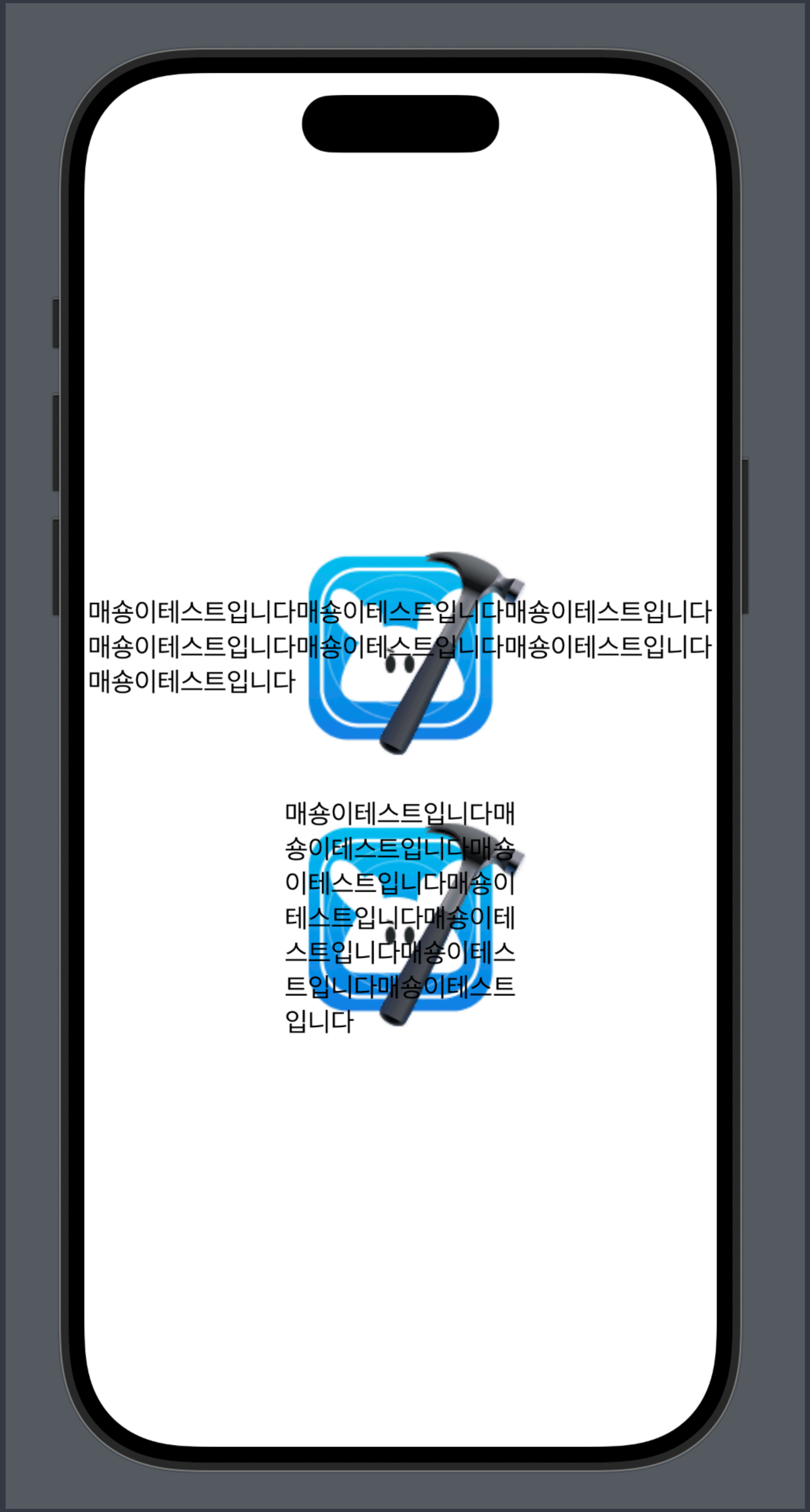
## 막간을 이용한 호기심 해소타임

막간을 이용한 호기심 해소타임

## ZStack vs overlay

가장 큰 차이점은 **뷰의 종속성!**

```swift
struct ContentView: View {
    var body: some View {
        VStack {
            ZStack {
                Image(ImageResource.mashongCode)
                Text("매숑이테스트입니다매숑이테스트입니다매숑이테스트입니다매숑이테스트입니다매숑이테스트입니다매숑이테스트입니다매숑이테스트입니다")
            }
            .clipped()

            Image(ImageResource.mashongCode)
                .overlay {
                    Text("매숑이테스트입니다매숑이테스트입니다매숑이테스트입니다매숑이테스트입니다매숑이테스트입니다매숑이테스트입니다매숑이테스트입니다")
                }
        }
    }
}
```



매숑이테스트입니다매숑이테스트입니다매숑이테스트입니다
매숑이테스트입니다매숑이테스트입니다매숑이테스트입니다
매숑이테스트입니다

매숑이테스트입니다매
숑이테스트입니다매숑
이테스트입니다매숑이
테스트입니다매숑이테
스트입니다매숑이테스
트입니다매숑이테스트
입니다

## 결론

스유 짱!

Mash-Up
iOS Team

2025.03.25

# Thank you.