



APP SCHOOL : iOS

SwiftUI 기초

다국어와 현지화

다국어와 현지화

SwiftUI

국제화와 현지화

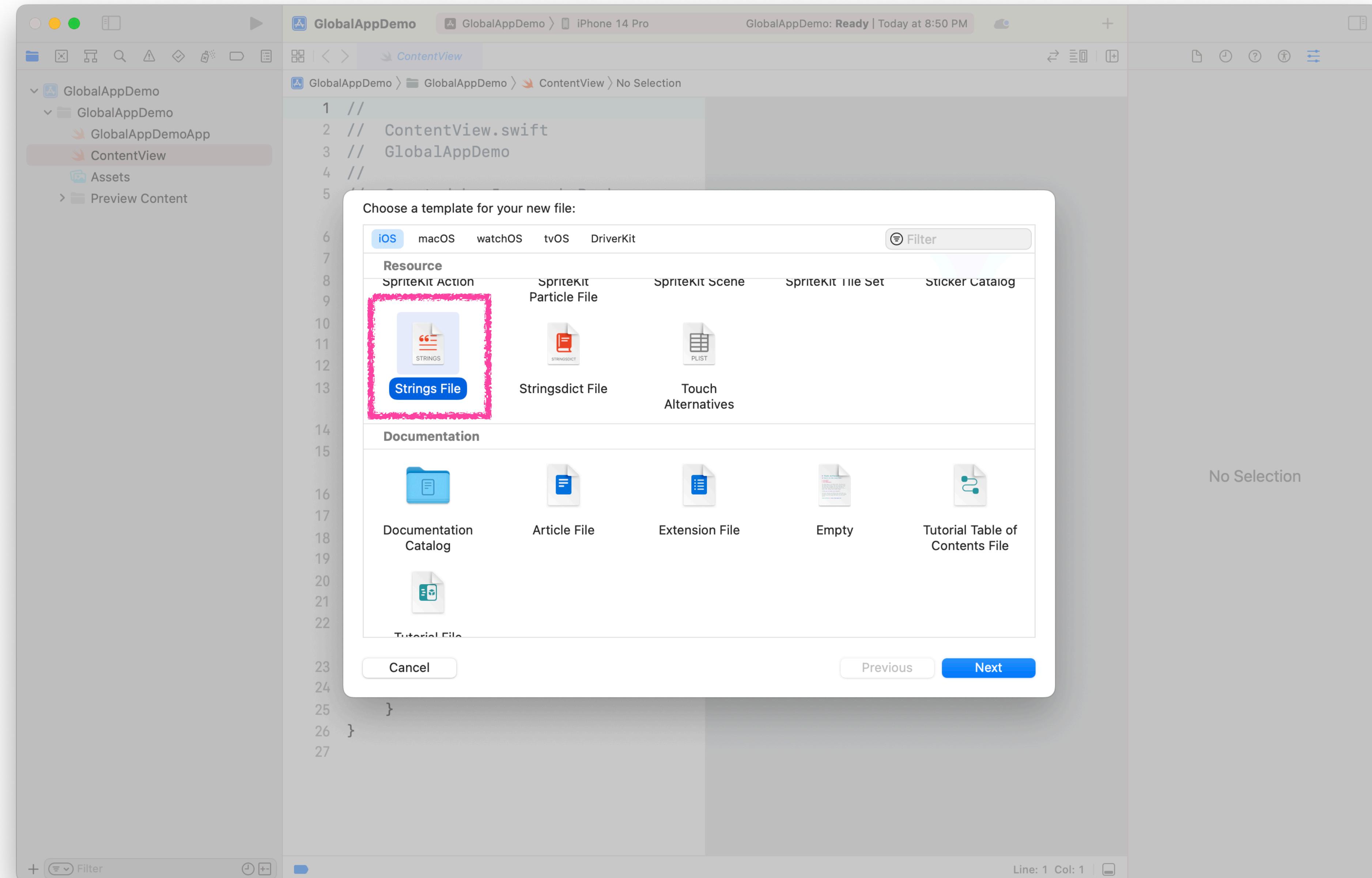
- 앱을 여러 나라에서 사용할 수 있게 하는 것
- 앱을 전 세계의 기기에서 작동되기 준비시킨다는 뜻
- 한 나라에서도 여러 언어를 지원해야 할 수 있다
 - 미국에서 스페인어 사용자
 - 수많은 언어의 인도 시장
- 현지화를 통해 해당 지역의 규칙에 맞춰줄 수 있다
 - 시간, 도량, 통화 단위

전통적 방식의 다국어 지원

SwiftUI

다국어 문자열 준비

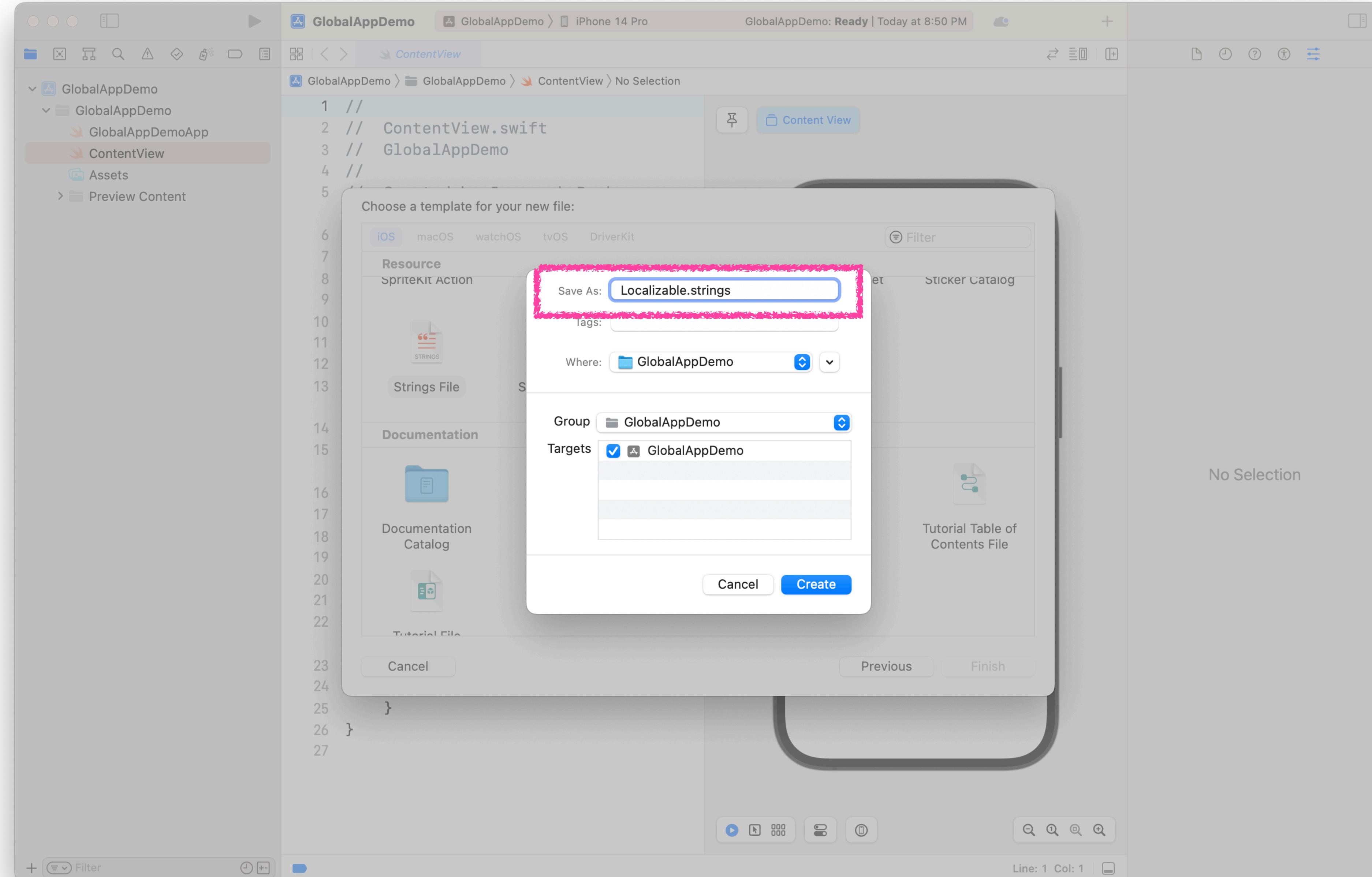
- String File 만들기



SwiftUI

다국어 문자열 준비

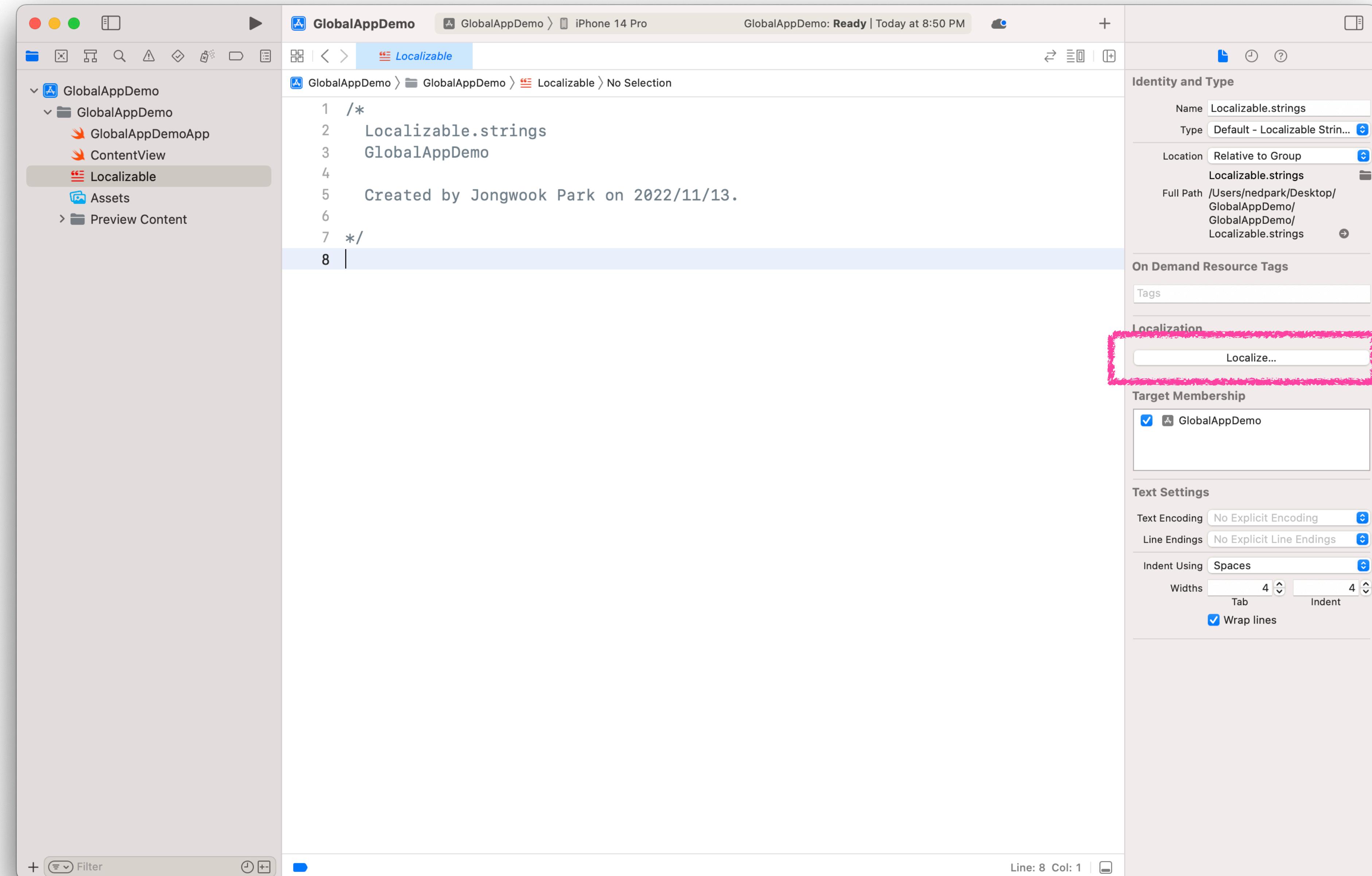
- String File 만들기
 - Localizable.strings



SwiftUI

다국어 문자열 준비

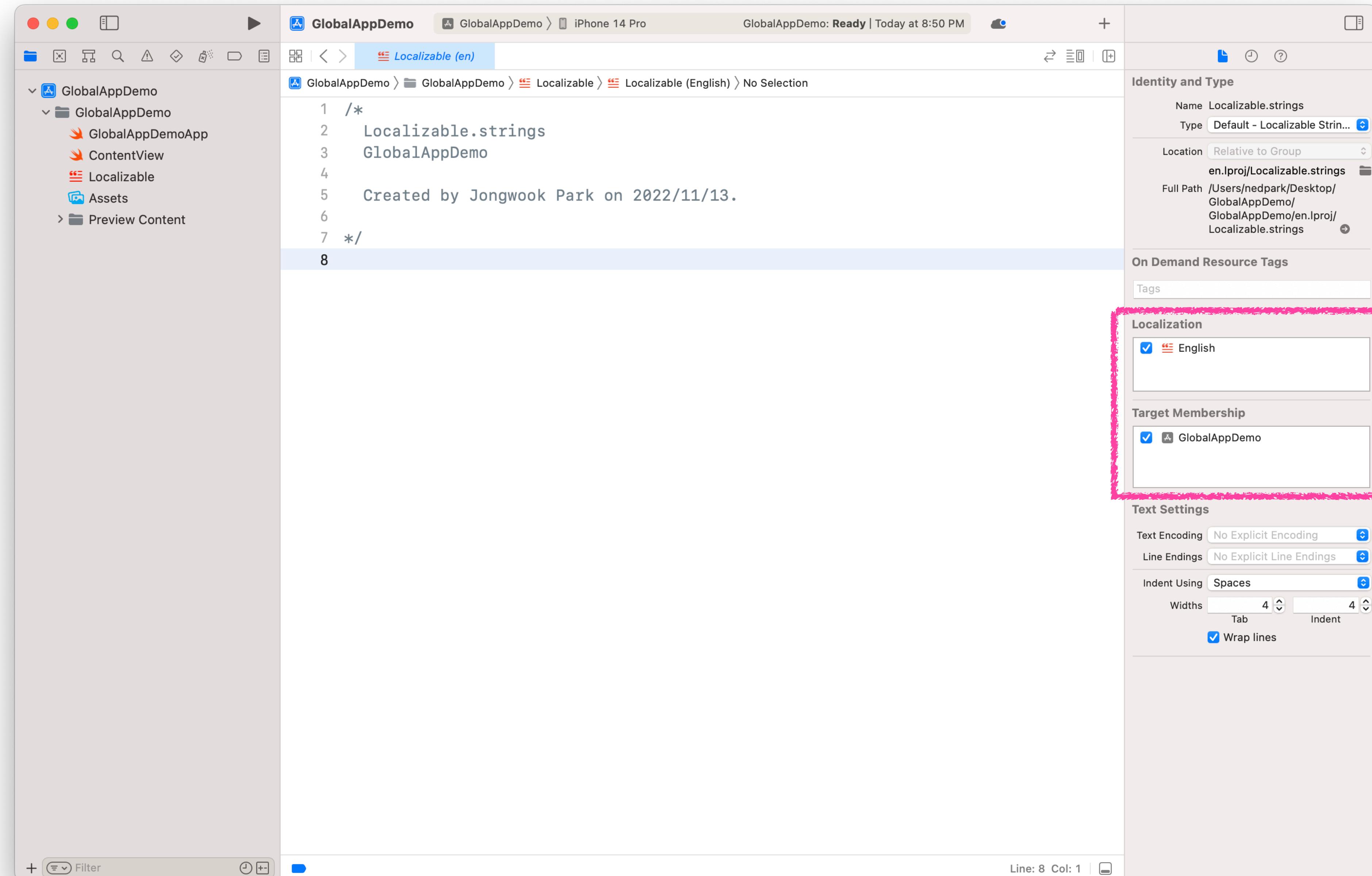
- String File 만들기
 - Localizable.strings
 - Localization 처리



SwiftUI

다국어 문자열 준비

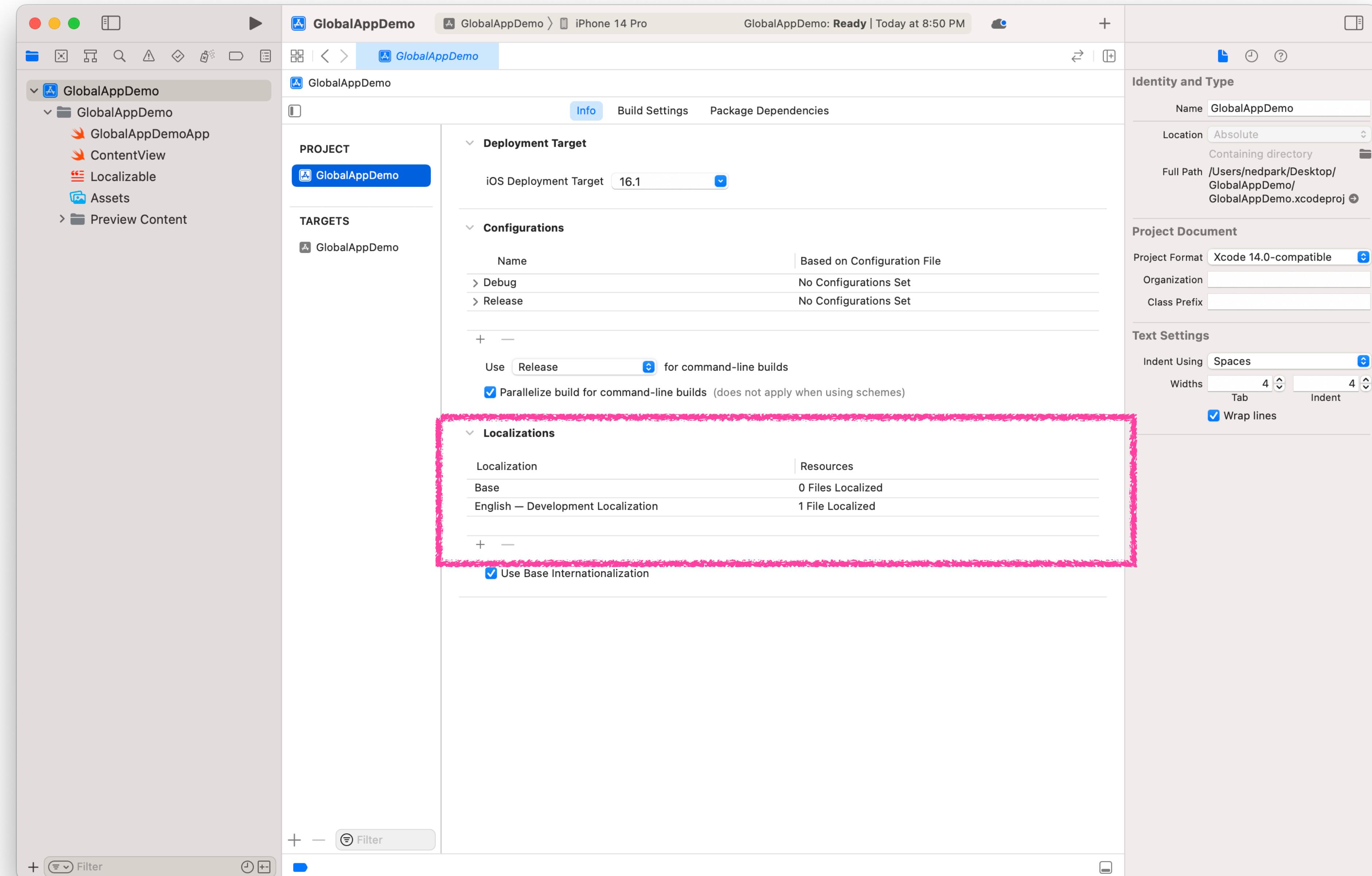
- String File 만들기
 - Localizable.strings
 - Localization 처리
 - 처리결과 확인



SwiftUI

다국어 문자열 준비

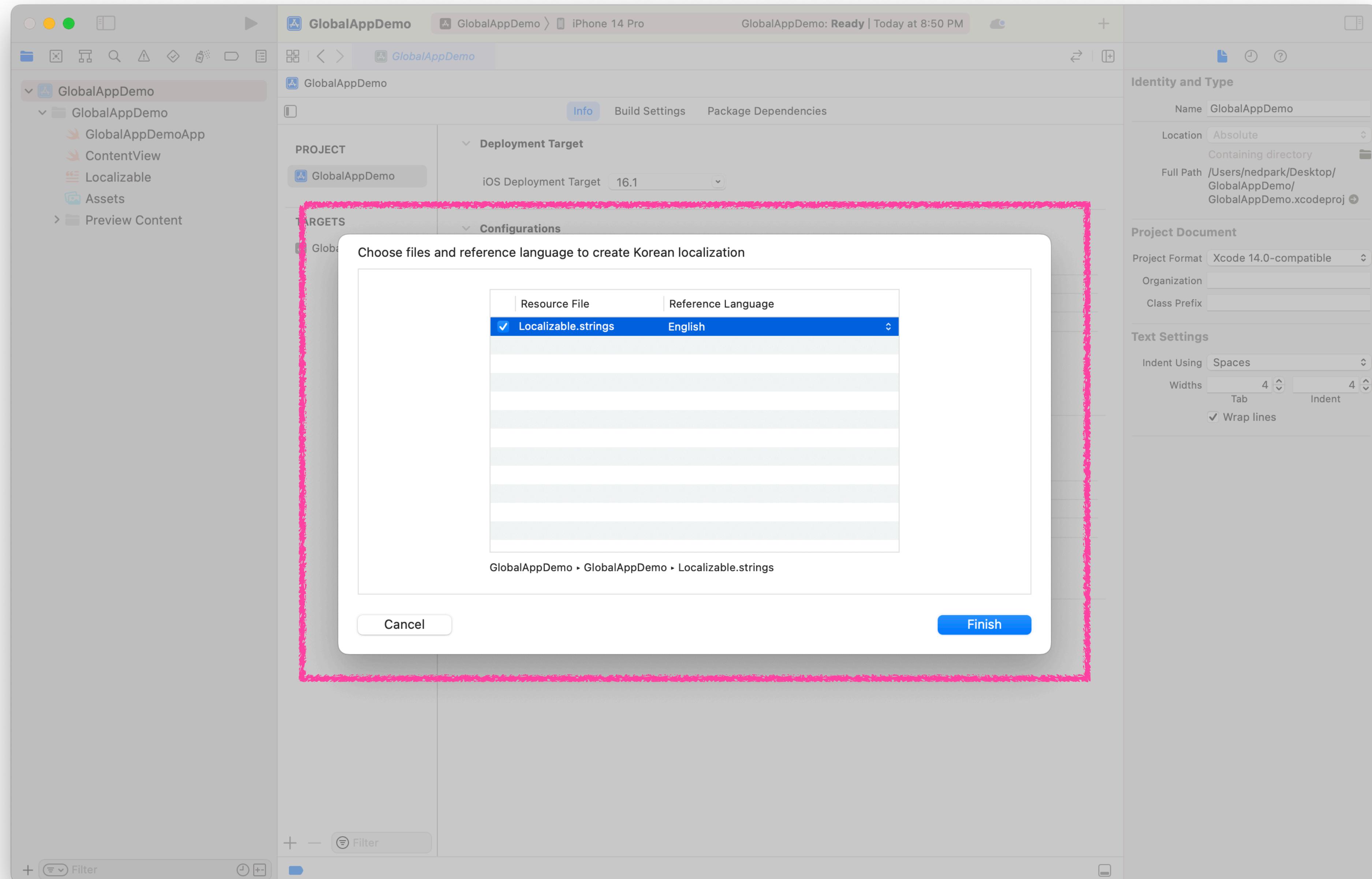
- String File 만들기
 - Localizable.strings
 - Localization 처리
 - 처리결과 확인



SwiftUI

다국어 문자열 준비

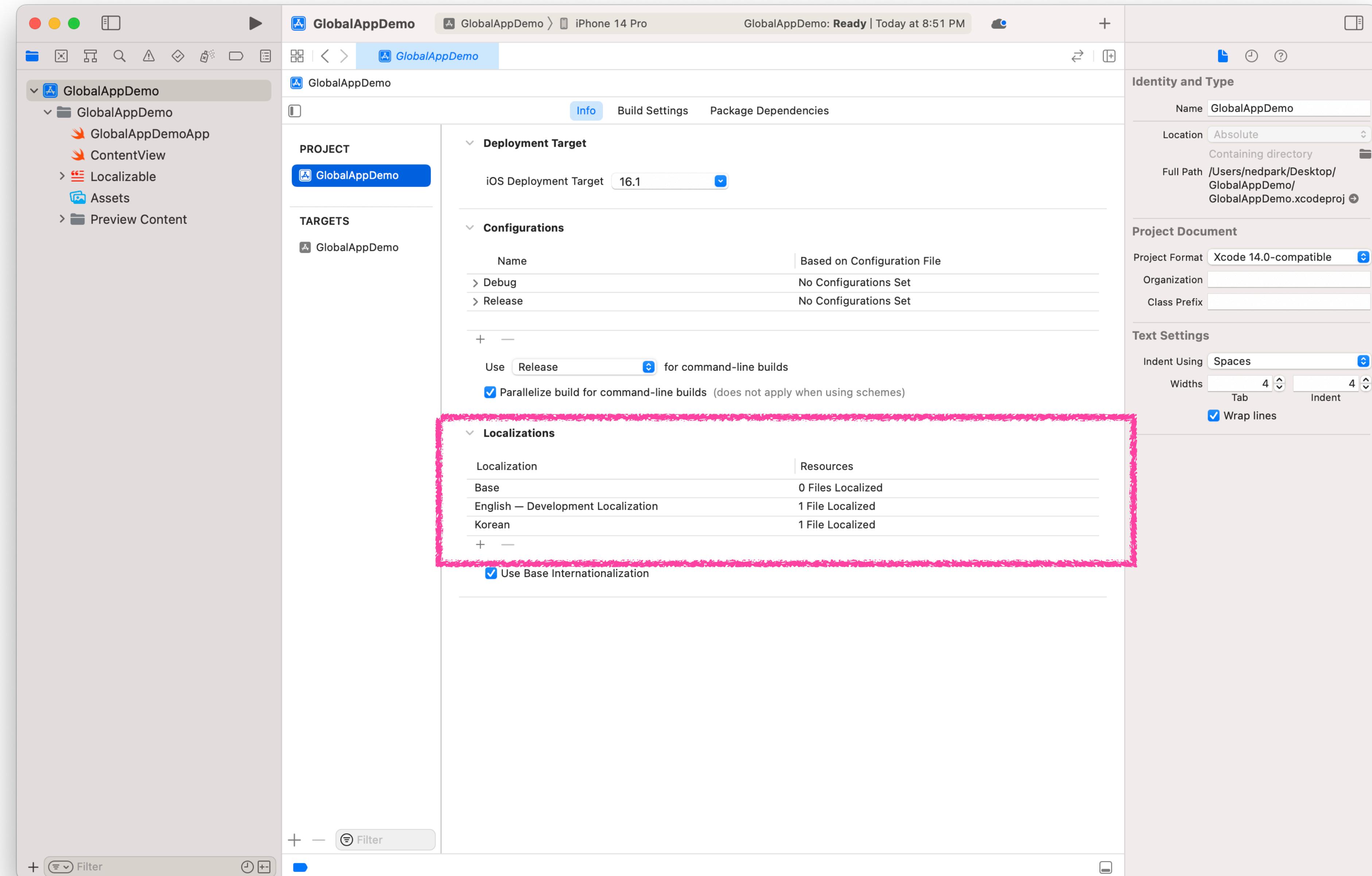
- String File 만들기
 - Localizable.strings
 - Localization 처리
 - 처리결과 확인
 - 영어 외 언어 추가



SwiftUI

다국어 문자열 준비

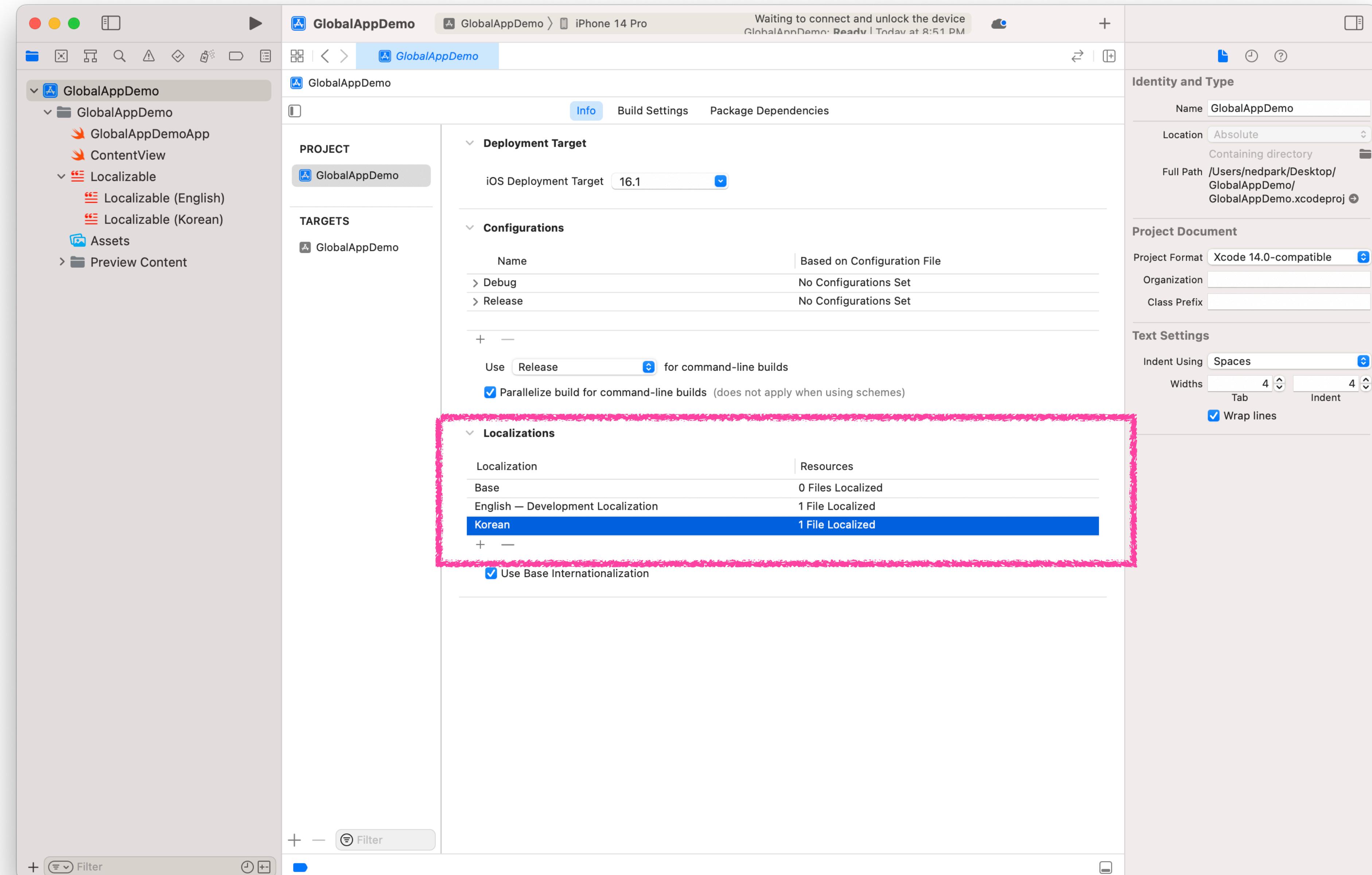
- String File 만들기
 - Localizable.strings
 - Localization 처리
 - 처리결과 확인
 - 영어 외 언어 추가



SwiftUI

다국어 문자열 준비

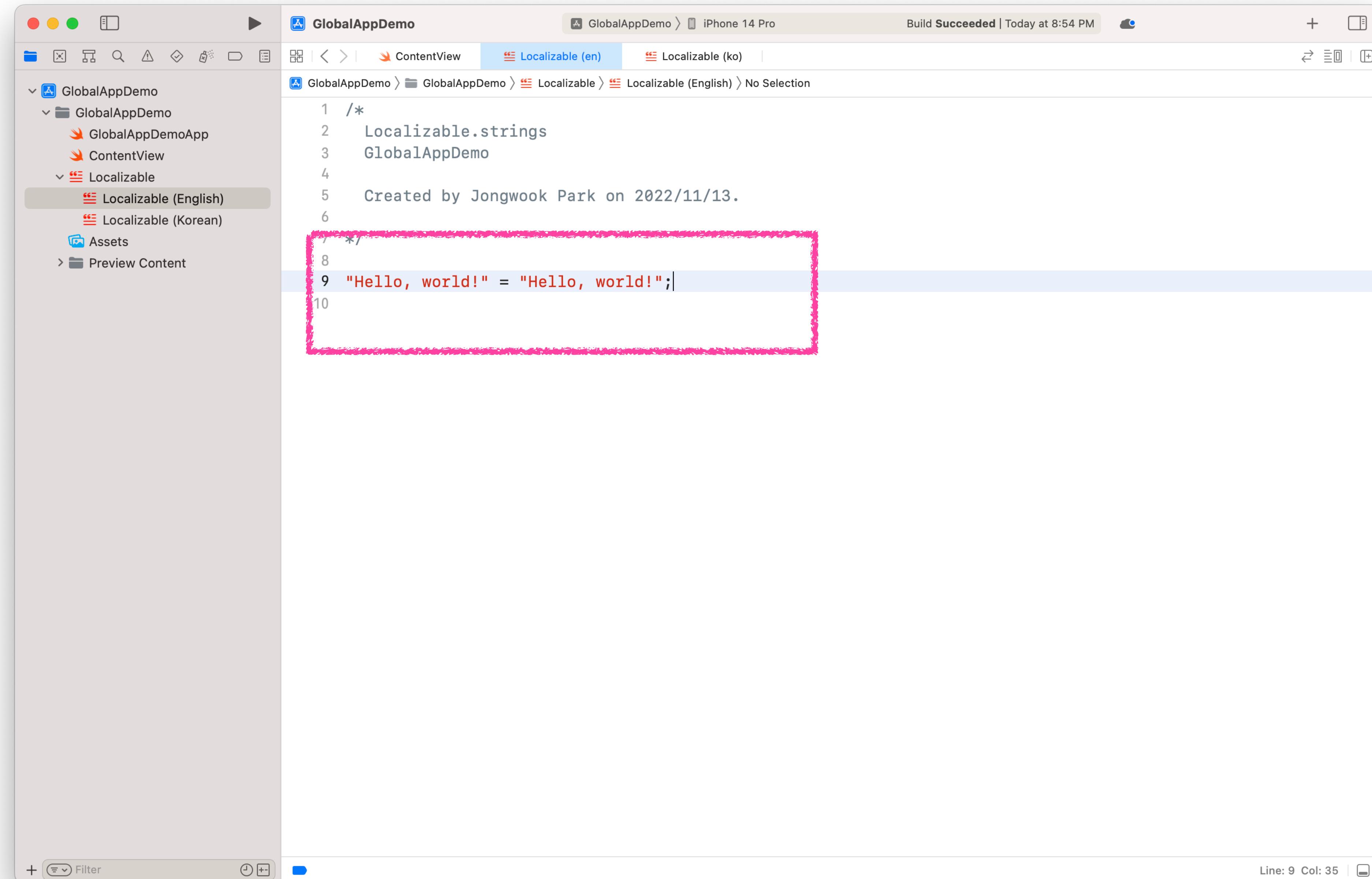
- String File 만들기
 - Localizable.strings
 - Localization 처리
 - 처리결과 확인
 - 영어 외 언어 추가



SwiftUI

다국어 문자열 준비

- String File 만들기
 - Localizable.strings
 - Localization 처리
 - 처리결과 확인
 - 영어 외 언어 추가
 - 다국어 문자열 정의



The screenshot shows the Xcode interface with the project "GlobalAppDemo" selected. The "ContentView" tab is active in the top navigation bar. The left sidebar shows the project structure with "Localizable" expanded, revealing "Localizable (English)" and "Localizable (Korean)". The main editor area displays the "Localizable (en)" file content:

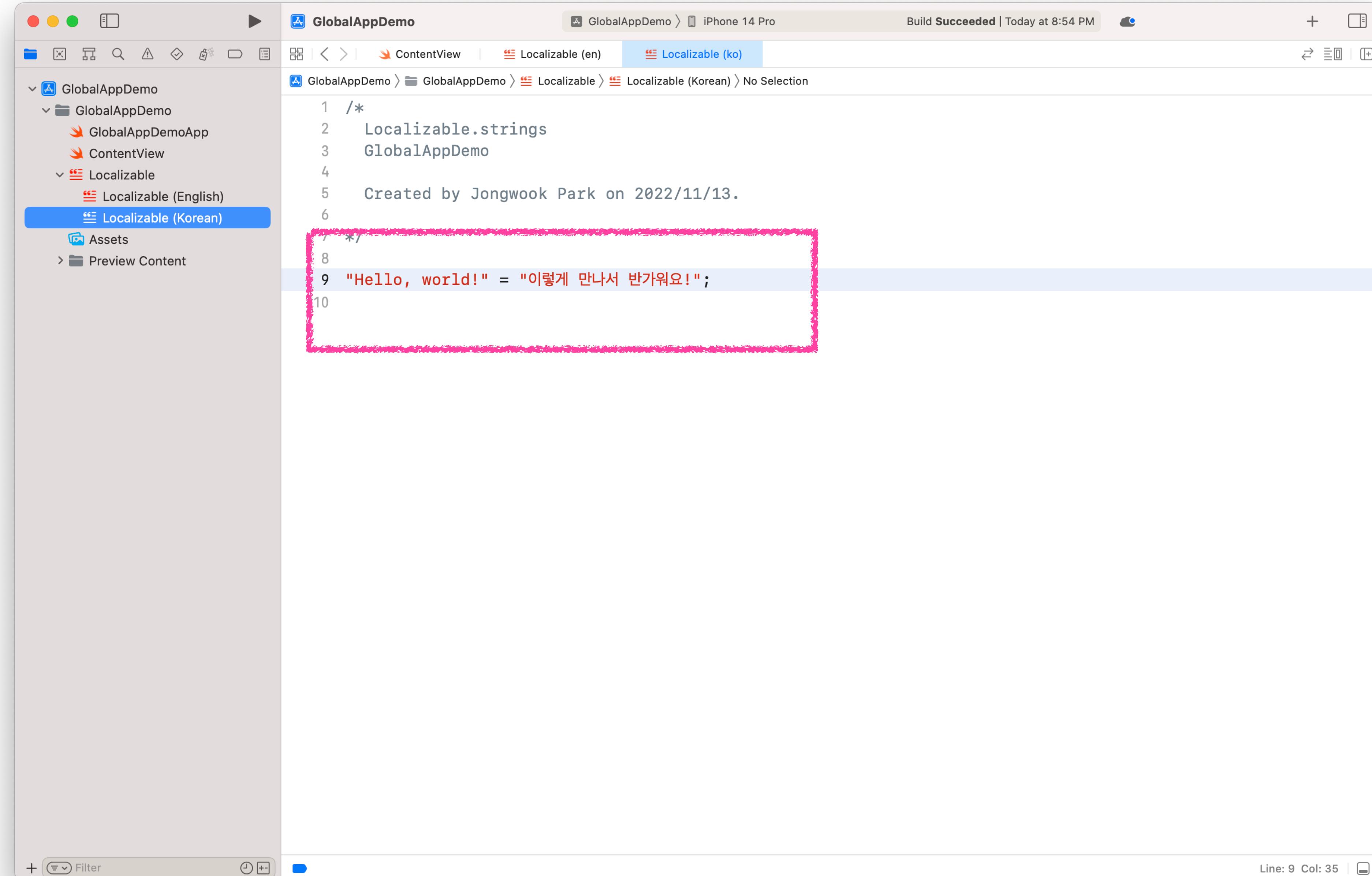
```
1 /*  
2 Localizable.strings  
3 GlobalAppDemo  
4  
5 Created by Jongwook Park on 2022/11/13.  
6  
7 */  
8  
9 "Hello, world!" = "Hello, world!";  
10
```

A red rectangular box highlights the string entry "9 \"Hello, world!\" = \"Hello, world!\";".

SwiftUI

다국어 문자열 준비

- String File 만들기
 - Localizable.strings
 - Localization 처리
 - 처리결과 확인
 - 영어 외 언어 추가
 - 다국어 문자열 정의



The screenshot shows the Xcode interface with the project "GlobalAppDemo" open. The "Localizable (ko)" tab is selected in the top navigation bar. The left sidebar shows the project structure with "Localizable" expanded, revealing "Localizable (English)" and "Localizable (Korean)". The main editor area displays the "Localizable.strings" file for the Korean localization. The file contains the following code:

```
1 /*  
2 Localizable.strings  
3 GlobalAppDemo  
4  
5 Created by Jongwook Park on 2022/11/13.  
6  
7 */  
8  
9 "Hello, world!" = "이렇게 만나서 반가워요!";  
10
```

A red rectangular box highlights the Korean translation line: "Hello, world!" = "이렇게 만나서 반가워요!";".

SwiftUI

다국어 문자열 준비

- String File 만들기
 - Localizable.strings
 - Localization 처리
 - 처리결과 확인
 - 영어 외 언어 추가
 - 다국어 문자열 정의
 - 정의된 문자열 활용

The screenshot shows the Xcode interface with the project 'GlobalAppDemo' open. The left sidebar shows the file structure, with 'ContentView.swift' selected. The main editor area displays the following Swift code:

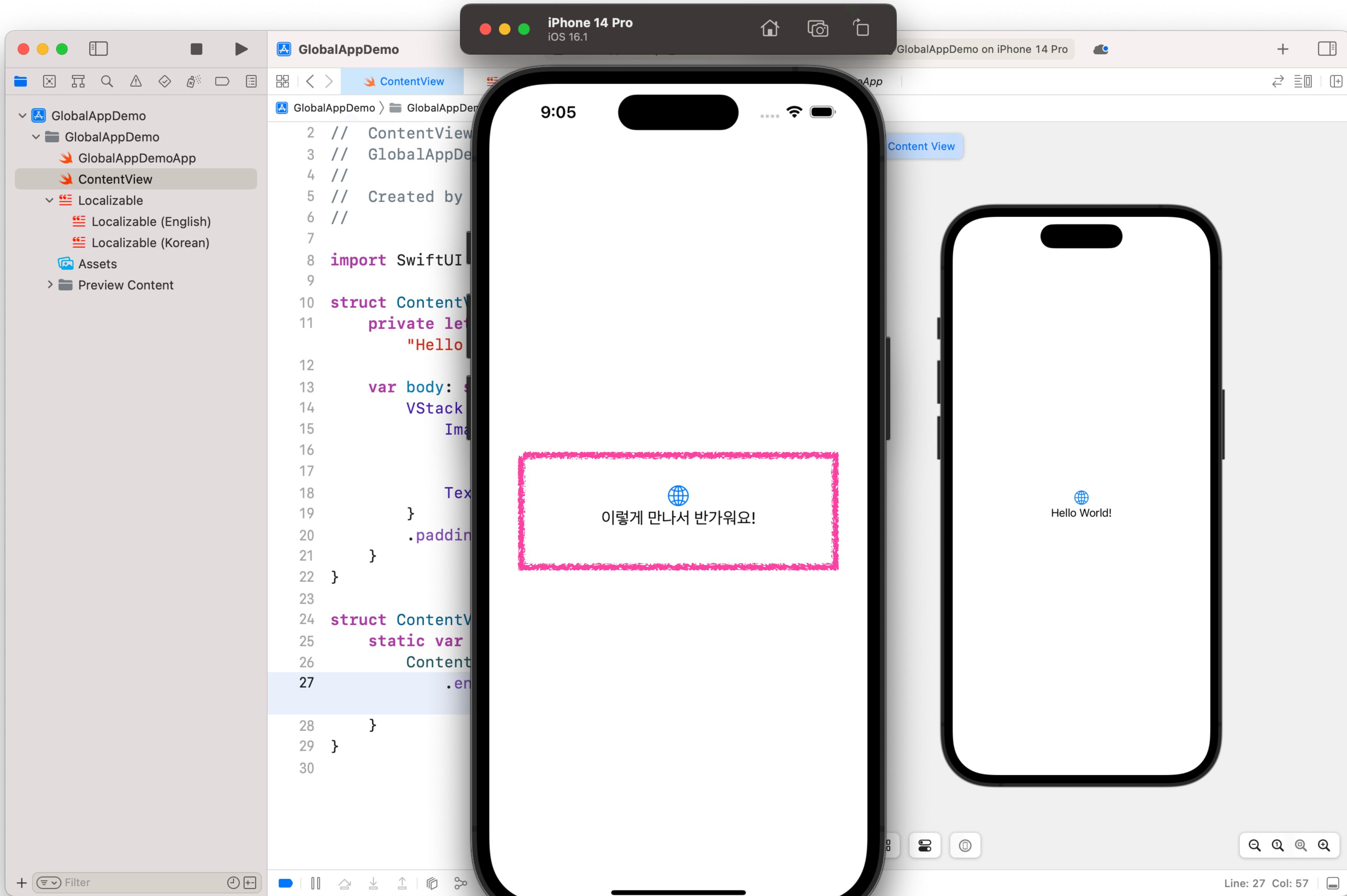
```
2 // ContentView.swift
3 // GlobalAppDemo
4 //
5 // Created by Jongwook Park on 2022/11/13.
6 //
7
8 import SwiftUI
9
10 struct ContentView: View {
11     private let HelloWorld: LocalizedStringKey =
12         "Hello World!"
13
14     var body: some View {
15         VStack {
16             Image(systemName: "globe")
17                 .imageScale(.large)
18                 .foregroundColor(.accentColor)
19             Text(HelloWorld)
20         }
21         .padding()
22     }
23
24 struct ContentView_Previews: PreviewProvider {
25     static var previews: some View {
26         ContentView()
27             .environment(\.locale,
28                         .init(identifier: "ko"))
29     }
30 }
```

A pink rectangular box highlights the line `private let HelloWorld: LocalizedStringKey = "Hello World!"`. To the right of the editor, an iPhone 14 Pro simulator is shown running the app, displaying the text "Hello World!" above a globe icon.

SwiftUI

다국어 문자열 준비

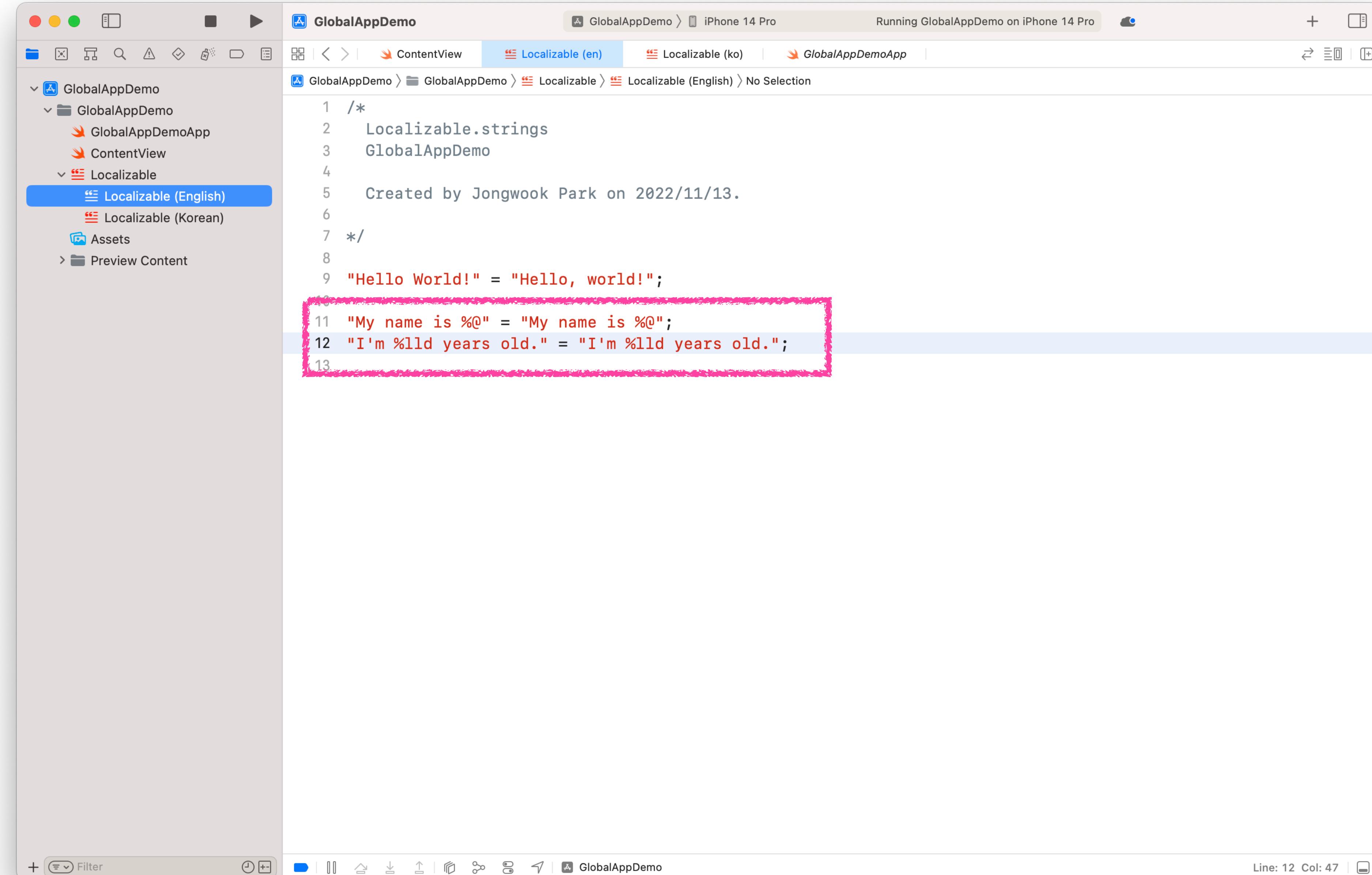
- String File 만들기
 - Localizable.strings
 - Localization 처리
 - 처리결과 확인
 - 영어 외 언어 추가
 - 다국어 문자열 정의
 - 정의된 문자열 활용
 - 적용 확인



SwiftUI

다국어 문자열 준비

- String File 만들기
 - Localizable.strings
 - Localization 처리
 - 처리결과 확인
 - 영어 외 언어 추가
 - 다국어 문자열 정의
 - 정의된 문자열 활용
 - 적용 확인
- 문자열은 %@
- 숫자는 %lld



The screenshot shows the Xcode interface with the project 'GlobalAppDemo' open. The 'ContentView' tab is selected in the top navigation bar. The left sidebar shows the project structure with 'Localizable' expanded, and 'Localizable (English)' selected. The main editor area displays the 'Localizable.strings' file content:

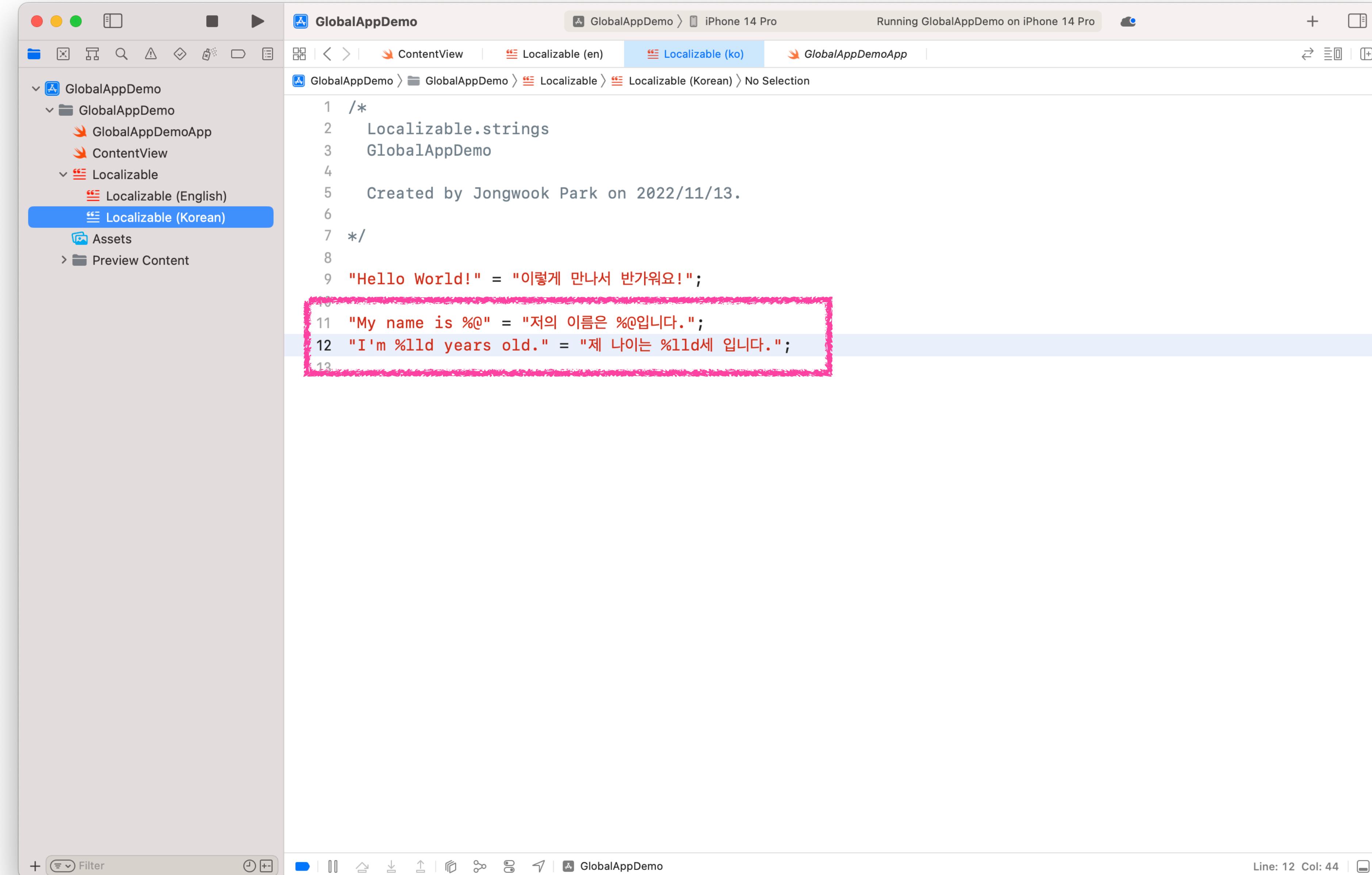
```
1 /*  
2 Localizable.strings  
3 GlobalAppDemo  
4  
5 Created by Jongwook Park on 2022/11/13.  
6  
7 */  
8  
9 "Hello World!" = "Hello, world!";  
10  
11 "My name is %@", "My name is %@",  
12 "I'm %lld years old.", "I'm %lld years old.",  
13
```

Lines 11 and 12 are highlighted with a red box.

SwiftUI

다국어 문자열 준비

- String File 만들기
 - Localizable.strings
 - Localization 처리
 - 처리결과 확인
 - 영어 외 언어 추가
 - 다국어 문자열 정의
 - 정의된 문자열 활용
 - 적용 확인
- 문자열은 %@
- 숫자는 %lld



The screenshot shows the Xcode interface with the project 'GlobalAppDemo' open. The 'Localizable (ko)' tab is selected in the top navigation bar. The left sidebar shows the project structure with 'Localizable (English)' and 'Localizable (Korean)' selected. The main editor area displays the 'Localizable.strings' file content:

```
1 /*  
2 Localizable.strings  
3 GlobalAppDemo  
4  
5 Created by Jongwook Park on 2022/11/13.  
6  
7 */  
8  
9 "Hello World!" = "이렇게 만나서 반가워요!";  
10  
11 "My name is %@", "저의 이름은 %@입니다.";  
12 "I'm %lld years old." = "제 나이는 %lld세입니다.";  
13
```

A red box highlights the Korean translations for the strings 'Hello World!', 'My name is %@', and 'I'm %lld years old.'.

SwiftUI

다국어 문자열 준비

- String File 만들기
 - Localizable.strings
 - Localization 처리
 - 처리결과 확인
 - 영어 외 언어 추가
 - 다국어 문자열 정의
 - 정의된 문자열 활용
 - 적용 확인
- 문자열은 %@
- 숫자는 %lld

The screenshot shows the Xcode interface with the following details:

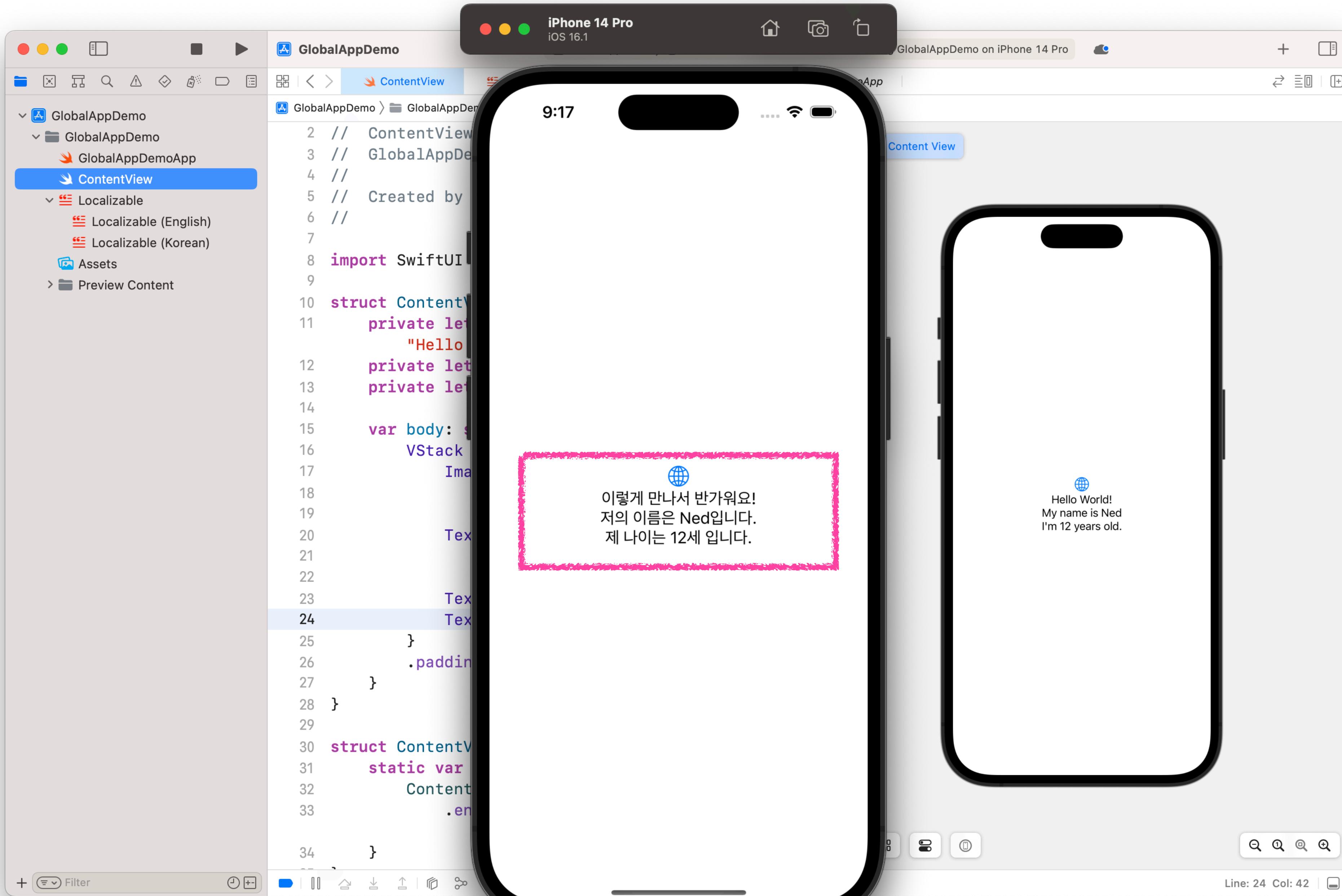
- Project Structure:** GlobalAppDemo > GlobalAppDemo > ContentView
- Code Editor:** ContentView.swift
- Code Content:**

```
2 // ContentView.swift
3 // GlobalAppDemo
4 //
5 // Created by Jongwook Park on 2022/11/13.
6 //
7
8 import SwiftUI
9
10 struct ContentView: View {
11     private let HelloWorld: LocalizedStringKey =
12         "Hello World!"
13     private let name: String = "Ned"
14     private let age: Int = 12
15
16     var body: some View {
17         VStack {
18             Image(systemName: "globe")
19                 .imageScale(.large)
20                 .foregroundColor(.accentColor)
21             Text(HelloWorld)
22
23             Text("My name is \(name)")
24             Text("I'm \(age) years old.")
25         }
26         .padding()
27     }
28 }
29
30 struct ContentView_Previews: PreviewProvider {
31     static var previews: some View {
32         ContentView()
33             .environment(\.locale,
34                         .init(identifier: "ko"))
35     }
36 }
```
- Preview Area:** Shows an iPhone 14 Pro displaying the app's interface with the text "Hello World!", "My name is Ned", and "I'm 12 years old."
- Bottom Status Bar:** Line: 24 Col: 42

SwiftUI

다국어 문자열 준비

- String File 만들기
 - Localizable.strings
 - Localization 처리
 - 처리결과 확인
 - 영어 외 언어 추가
 - 다국어 문자열 정의
 - 정의된 문자열 활용
 - 적용 확인
- 문자열은 %@
- 숫자는 %lld



GlobalAppDemo

GlobalAppDemo > iPhone 14 Pro

Finished running GlobalAppDemo on iPhone 14 Pro

ContentView

GlobalAppDemo | Localizable (en) | Localizable (ko)

GlobalAppDemo > GlobalAppDemo > ContentView > body

```
7
8 import SwiftUI
9
10 struct ContentView: View {
11     var body: some View {
12         NavigationStack {
13             VStack {
14                 Image(systemName: "globe")
15                     .imageScale(.large)
16                     .foregroundColor(.accentColor)
17                 Text("Hello World!")
18                 Label("Message", image: "msgSymbol")
19
20                 Label {
21                     Text("Message",
22                         comment: "A label that displays 'Message'
23                                     and a corresponding image.")
24                     } icon: {
25                         Image("msgSymbol")
26                     }
27                     .font(.largeTitle)
28                     .padding()
29                     .navigationTitle(Text("Landmarks", comment:
30                                     "navigation title"))
31             }
32         }
33
34 struct ContentView_Previews: PreviewProvider {
35     static var previews: some View {
36         ContentView()
37             .environment(\.locale, Locale(identifier: "ko"))
38     }
39 }
```

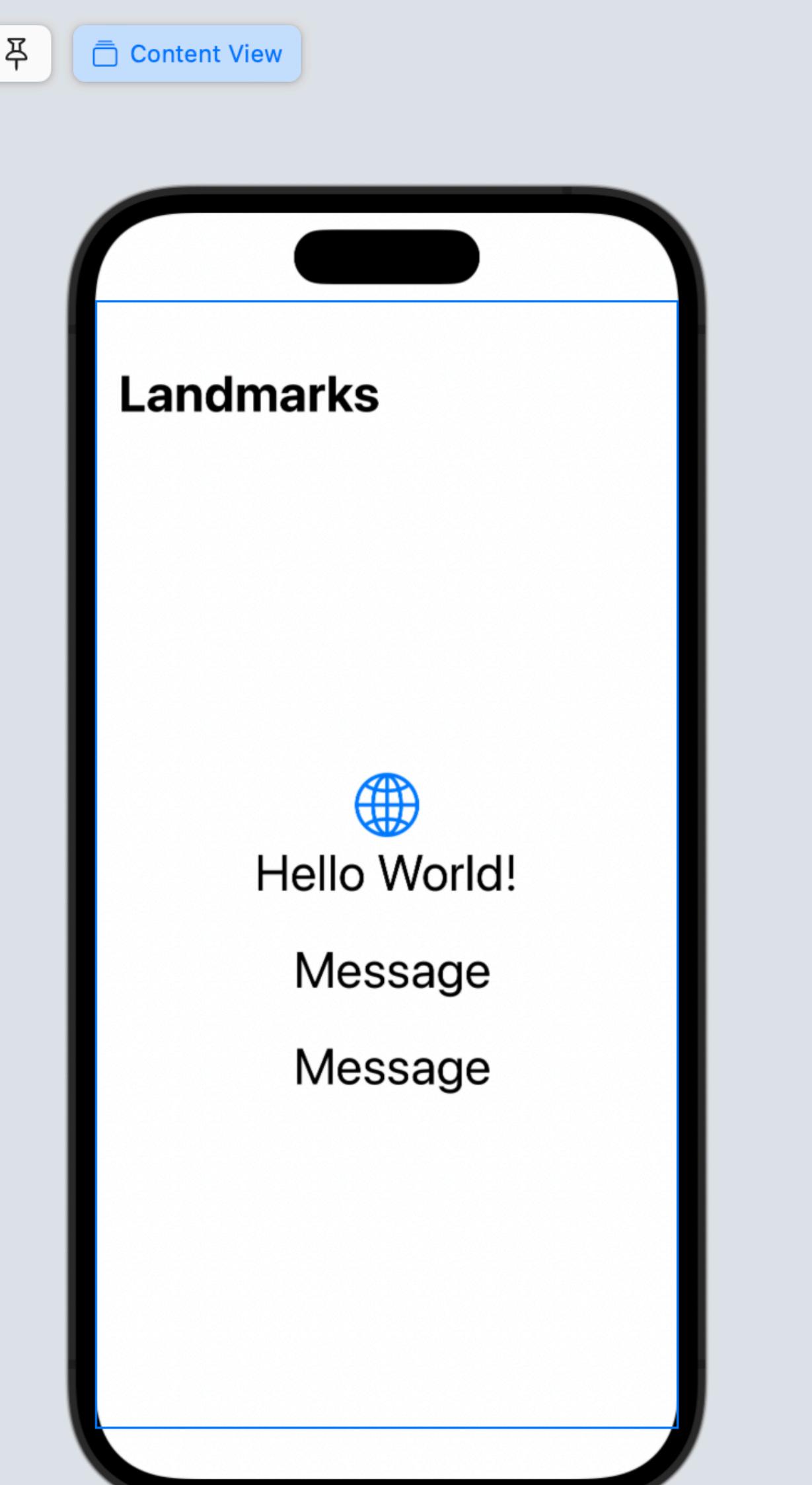
Content View

Landmarks

Hello World!

Message

Message



iPhone 14 Pro GlobalAppDemo iPhone 14 Pro Running GlobalAppDemo on iPhone 14 Pro

ContentView GlobalAppDemo Localizable (en) Localizable (ko)

GlobalAppDemo > ContentView > body

10:09

랜드마크

사과

감자

포도

당근

```
1 // ContentView.swift
2 // GlobalAppDemo
3 // Created by Jongwook Park on 2022/11/13.
4 //
5
6 import SwiftUI
7
8 struct ContentView: View {
9     let items = ["Apple", "Potato", "Grape", "Carrot"]
10
11     var body: some View {
12         NavigationStack {
13             List {
14                 ForEach(items, id:\.self) { item in
15                     Text(LocalizedStringKey(item))
16                         .font(.largeTitle)
17                 }
18             }
19         }
20     }
21     .navigationTitle(Text("Landmarks", comment:
22         "navigation title"))
23 }
24 }
25
26 struct ContentView_Previews: PreviewProvider {
27     static var previews: some View {
28         ContentView()
29             .environment(\.locale, Locale(identifier: "ko"))
30     }
31 }
32
```

Content View

Preview paused

Landmarks

Apple

Potato

Grape

Carrot

단위 현지화

GlobalAppDemo

GlobalAppDemo > iPhone 14 Pro

Running GlobalAppDemo on iPhone 14 Pro

ContentView GlobalAppDemo Localizable (en) Localizable (ko)

GlobalAppDemo > GlobalAppDemo > ContentView > body

```
10 struct ContentView: View {  
11     let formatter: LengthFormatter  
12     let subject: NSNumber  
13  
14     init() {  
15         self.formatter = LengthFormatter()  
16         self.formatter.numberFormatter.minimumFractionDigits = 2  
17         self.formatter.numberFormatter.maximumFractionDigits = 2  
18         self.subject = NSNumber(value: 123.45)  
19     }  
20  
21     var body: some View {  
22         VStack {  
23             Image(systemName: "globe")  
24                 .imageScale(.large)  
25                 .foregroundColor(.accentColor)  
26             Text(Date(), style: .date)  
27             Text(subject, formatter: formatter)|  
28         }  
29         .font(.largeTitle)  
30         .padding()  
31     }  
32 }  
33  
34 struct ContentView_Previews: PreviewProvider {  
35     static var previews: some View {  
36         ContentView()  
37             .environment(\.locale, Locale(identifier: "en"))  
38     }  
39 }  
40
```

Content View

November 13, 2022

135.00 yd

Filter

Line: 27 Col: 48

GlobalAppDemo

GlobalAppDemo > iPhone 14 Pro

Running GlobalAppDemo on iPhone 14 Pro

ContentView GlobalAppDemo Localizable (en) Localizable (ko)

GlobalAppDemo > GlobalAppDemo > ContentView > body

```
10 struct ContentView: View {  
11     let formatter: LengthFormatter  
12     let subject: NSNumber  
13  
14     init() {  
15         self.formatter = LengthFormatter()  
16         self.formatter.numberFormatter.minimumFractionDigits = 2  
17         self.formatter.numberFormatter.maximumFractionDigits = 2  
18         self.subject = NSNumber(value: 123.45)  
19     }  
20  
21     var body: some View {  
22         VStack {  
23             Image(systemName: "globe")  
24                 .imageScale(.large)  
25                 .foregroundColor(.accentColor)  
26             Text(Date(), style: .date)  
27             Text(subject, formatter: formatter)|  
28         }  
29         .font(.largeTitle)  
30         .padding()  
31     }  
32 }  
33  
34 struct ContentView_Previews: PreviewProvider {  
35     static var previews: some View {  
36         ContentView()  
37             .environment(\.locale, Locale(identifier: "ko"))  
38     }  
39 }  
40
```

Content View

2022년 11월 13일
123.45m

Filter

Line: 27 Col: 48

SwiftUI

단위 현지화

- 각종 Formatter 사용
- LengthFormatter
 - Deprecated

Class

LengthFormatter

A formatter that provides localized descriptions of linear distances, such as length and height measurements.

iOS 8.0+

iPadOS 8.0+

macOS 10.10+

Mac Catalyst 13.1+

tvOS 9.0+

watchOS 2.0+

Declaration

```
class LengthFormatter : Formatter
```

참고자료

WWDC22

Build global apps: Localization by example

Andreas Neusüß, Software Localization Engineer

개요

전문

코드

글로벌 앱 빌드: 사례별 현지화

WWDC22

22분 남음

코드 복사 지원



사용 중인 언어에 관계없이 전 세계의 기기에서 앱을 실행하고 모든 사용자에게 탁월한 경험을 제공하도록 지원하는 방법을 알아보세요. 다양한 고객층을 위한 앱 제작 시 Apple API를 통해 탄탄한 기반을 얻을 수 있는 방법을 알아보고, 그동안의 경험을 바탕으로 사례, 문제 및 모범 사례를 공유합니다.

자료

The screenshot shows a Mac OS X desktop environment with a window open to the Apple Developer Documentation website at developer.apple.com. The window title is "Documentation / Xcode / Localization". The main content area features a large dark purple header with the word "Localization" in white. Below the header, there's a sub-header "Expand the market for your app by supporting multiple languages and regions." followed by a section titled "Overview". This section contains text about the localization process and a paragraph about translating and adapting apps. To the left of the main content is a sidebar titled "Xcode" with a "Localization" section expanded. The "Localization" section includes links for Internationalization, Languages and regions, Resources and assets, Translation and adaptation, Testing, and a "Filter" button. The URL in the browser bar is https://developer.apple.com/documentation/Xcode/localization.

Localization

Expand the market for your app by supporting multiple languages and regions.

Overview

Localization is the process of translating and adapting your app into multiple languages and regions. Localize your app to provide access for users who speak a variety of languages, and who download from different App Store territories.

First, *internationalize* your code with APIs that automatically format and translate strings correctly for the language and region. Then add support for content that includes plural nouns and verbs by following language plural rules to increase the accuracy of your translations..

Translate and adapt your app

In Xcode, localization refers specifically to the set of resources for a specific language and region that you support.

You add a localization to your project and select the resources you want to include for that language and region. Export the localization and send the files to *localizers*, who translate the user-facing text and adapt resources for particular cultures and regions. Finally, you import the localized files and test the app in that language and region directly in Xcode.

developer.apple.com

Documentation Archive

Internationalization and Localization Guide

Table of Contents

Introduction

- ▶ Reviewing Language and Region Settings
- ▶ Internationalizing the User Interface
- ▶ Internationalizing Your Code
- ▶ Formatting Data Using the Locale Settings
- ▶ Supporting Right-to-Left Languages
- ▶ Localizing Your App
- ▶ Testing Your Internationalized App
- ▶ Appendix A: Managing Strings Files Yourself
- ▶ Appendix B: Language and Locale IDs
- ▶ Appendix C: Stringsdict File Format

Revision History

Glossary

Next

About Internationalization and Localization

Note: This document was previously titled *Internationalization Programming Topics*.

Localization is the process of translating your app into multiple languages. But before you can localize your app, you internationalize it. Internationalization is the process of making your app able to adapt to different languages, regions, and cultures. Because a single language can be used in multiple parts of the world, your app should adapt to the regional and cultural conventions of where a person resides. An internationalized app appears as if it is a native app in all the languages and regions it supports.

The App Store is available in over 150 different countries, and internationalizing your app is the first step to reach this global market. In App Store Connect, you specify whether your app is available in all territories or specific territories. Then you customize your app for each target market that you want to support. Users in other countries want to use your app in a language they understand and see dates, times, and numbers in familiar, regional formats.





APP SCHOOL : iOS

SwiftUI 기초

NavigationStack



Navigation API

앱의 탐색 동작을 프로그래밍한 방식으로 제어하여
실행 상태 설정, 크기 클래스 간 전환 관리, 딥 링크에
응답하는 등의 작업을 할 수 있습니다.

SwiftUI4Demo

ContentView

SwiftUI4Demo ContentView.swift body

```
1 //  
2 // ContentView.swift  
3 // SwiftUI4Demo  
4 //  
5 // Created by Jongwook Park on 2022/11/14.  
6 //  
7  
8 import SwiftUI  
9  
10 struct ContentView: View {  
11     var body: some View {  
12         NavigationStack {  
13             List(1..<30) { index in  
14                 NavigationLink {  
15                     Text("Employee No. \(index)")  
16                 } label: {  
17                     Label("Employee No. \(index)",  
18                         systemImage: "\((index).square")  
19                 }  
20             }.navigationTitle("Hello Inc.")  
21         }  
22     }  
23 }  
24  
25 struct ContentView_Previews: PreviewProvider {  
26     static var previews: some View {  
27         ContentView()  
28     }  
29 }  
30
```

Content View

Hello Inc.

Employee No. 1
Employee No. 2
Employee No. 3
Employee No. 4
Employee No. 5
Employee No. 6
Employee No. 7
Employee No. 8
Employee No. 9
Employee No. 10
Employee No. 11
Employee No. 12
Employee No. 13
Employee No. 14
Employee No. 15
Employee No. 16

Line: 12 Col: 26

SwiftUI4Demo

ContentView

SwiftUI4Demo ContentView body

```
1 //  
2 // ContentView.swift  
3 // SwiftUI4Demo  
4 //  
5 // Created by Jongwook Park on 2022/11/14.  
6 //  
7  
8 import SwiftUI  
9  
10 struct ContentView: View {  
11     var body: some View {  
12         NavigationStack {  
13             List(1..<30) { index in  
14                 NavigationLink {  
15                     Text("Employee No. \(index)")  
16                 } label: {  
17                     Label("Employee No. \(index)",  
18                         systemImage: "\((index).square")  
19                 }  
20             }.navigationTitle("Hello Inc.")  
21         }  
22     }  
23 }  
24  
25 struct ContentView_Previews: PreviewProvider {  
26     static var previews: some View {  
27         ContentView()  
28     }  
29 }
```

Content View

Hello Inc.

Employee No. 1
Employee No. 2
Employee No. 3
Employee No. 4
Employee No. 5
Employee No. 6
Employee No. 7
Employee No. 8
Employee No. 9
Employee No. 10
Employee No. 11
Employee No. 12
Employee No. 13
Employee No. 14
Employee No. 15
Employee No. 16

Line: 12 Col: 26

SwiftUI4Demo

Build Succeeded | Today at 10:26 PM

ContentView

```
1 //  
2 // ContentView.swift  
3 // SwiftUI4Demo  
4 //  
5 // Created by Jongwook Park on 2022/11/14.  
6 //  
7  
8 import SwiftUI  
9  
10 struct ContentView: View {  
11     var body: some View {  
12         NavigationStack {  
13             List(1..<30) { index in  
14                 NavigationLink(value: index) {  
15                     Label("Employee No. \((index)",  
16                         systemImage: "\((index).square")  
17                 }  
18             .navigationDestination(for: Int.self) {  
19                 index in  
20                     Text("Employee No. \((index)")  
21             }  
22         .navigationTitle("Hello Inc.")  
23     }  
24 }  
25  
26 struct ContentView_Previews: PreviewProvider {  
27     static var previews: some View {  
28         ContentView()  
29     }  
30 }  
31
```

Content View

Hello Inc.

Employee No. 1
Employee No. 2
Employee No. 3
Employee No. 4
Employee No. 5
Employee No. 6
Employee No. 7
Employee No. 8
Employee No. 9
Employee No. 10
Employee No. 11
Employee No. 12
Employee No. 13
Employee No. 14
Employee No. 15
Employee No. 16

Line: 18 Col: 25

SwiftUI4Demo

Build Succeeded | Today at 10:26 PM

ContentView

```
1 //  
2 // ContentView.swift  
3 // SwiftUI4Demo  
4 //  
5 // Created by Jongwook Park on 2022/11/14.  
6 //  
7  
8 import SwiftUI  
9  
10 struct ContentView: View {  
11     @State private var previousNumbers = [3, 6, 9]  
12  
13     var body: some View {  
14         NavigationStack(path: $previousNumbers) {  
15             List(1..<30) { index in  
16                 NavigationLink(value: index) {  
17                     Label("Employee No. \((index)",  
18                         systemImage: "\((index).square")  
19                 }  
20             .navigationDestination(for: Int.self) {  
21                 index in  
22                     Text("Employee No. \((index)")  
23             .navigationTitle("Hello Inc.")  
24         }  
25     }  
26 }  
27  
28 struct ContentView_Previews: PreviewProvider {  
29     static var previews: some View {  
30         ContentView()  
31     }  
32 }  
33
```

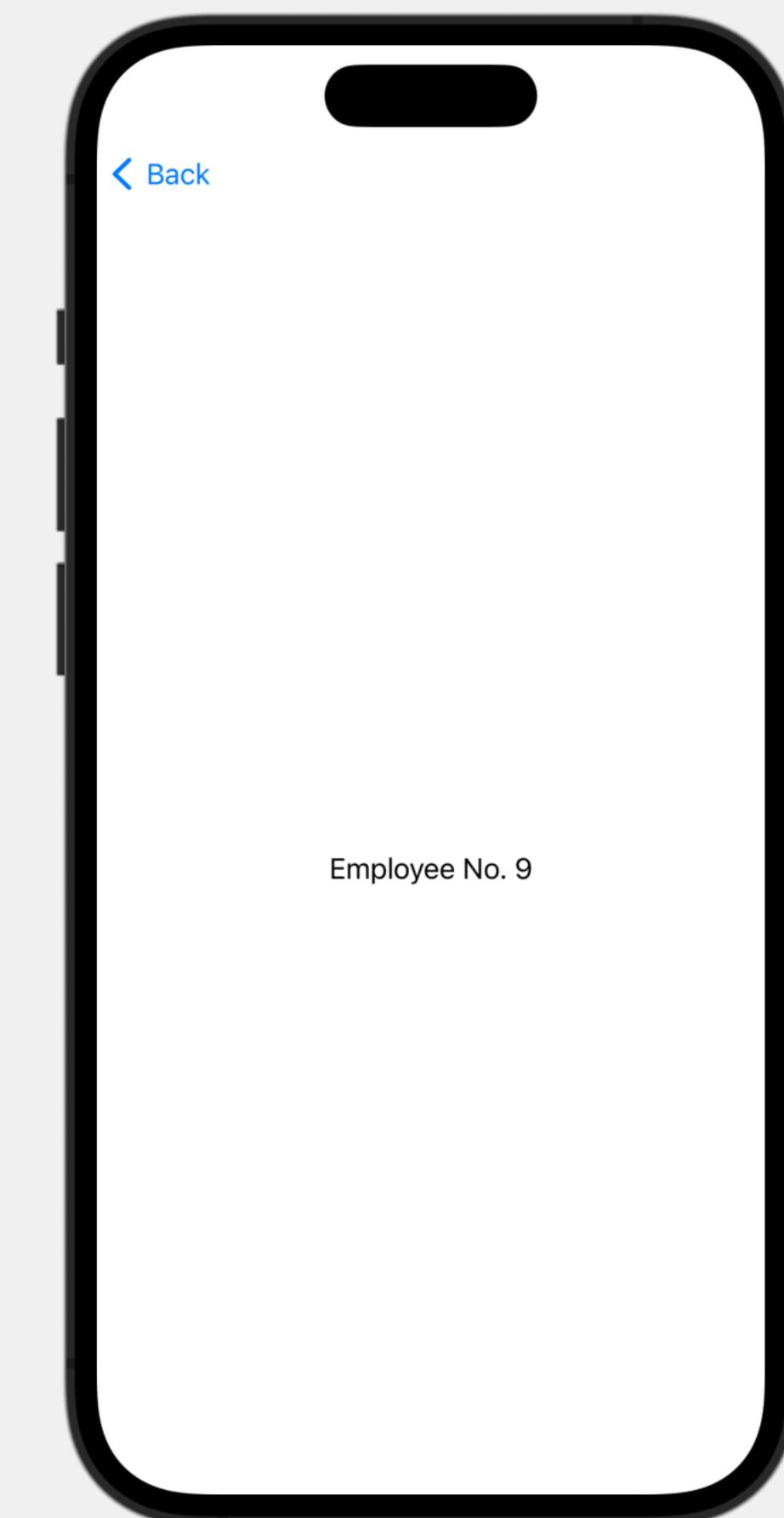
Content View

Back

Employee No. 9

Filter

Line: 14 Col: 50



SwiftUI4Demo

ContentView

SwiftUI4Demo > ContentView > previousNumbers

```
1 //  
2 // ContentView.swift  
3 // SwiftUI4Demo  
4 //  
5 // Created by Jongwook Park on 2022/11/14.  
6 //  
7  
8 import SwiftUI  
9  
10 struct ContentView: View {  
11     @State private var previousNumbers = NavigationPath()  
12  
13     var body: some View {  
14         NavigationStack(path: $previousNumbers) {  
15             NavigationLink(value: "Bob") {  
16                 Text("CEO")  
17             }  
18  
19             List(1..<30) { index in  
20                 NavigationLink(value: index) {  
21                     Label("Employee No. \((index)",  
22                         systemImage: "\((index).square")  
23                 }  
24             }.navigationDestination(for: Int.self) {  
25                 index in  
26                 Text("Employee No. \((index)")  
27             }.navigationDestination(for: String.self) {  
28                 name in  
29                 Text("CEO \(name) is special one")  
30             }.navigationTitle("Hello Inc.")  
31         }  
32     }  
33 }
```

Content View

Hello Inc.

CEO

Employee No. 1
Employee No. 2
Employee No. 3
Employee No. 4
Employee No. 5
Employee No. 6
Employee No. 7
Employee No. 8
Employee No. 9
Employee No. 10
Employee No. 11
Employee No. 12
Employee No. 13
Employee No. 14
Employee No. 15

Line: 11 Col: 43



APP SCHOOL : iOS

SwiftUI 기초

Grid



고급 레이아웃 컨트롤

재사용 가능한 고급 레이아웃을 빌드하여 앱 디자인을 강화할 수 있습니다. VStack 및 Hstack과 더불어 이제 SwiftUI에서는 새로운 Grid API를 제공하므로 가로 및 세로 보기와 동시에 정렬할 수 있습니다. 또한, 새로운 하위 수준의 맞춤형 Layout API를 통해 앱에 필요한 레이아웃을 정확하게 빌드할 수 있습니다. 특정 보기에 대한 여러 가지 변형을 지정하고 SwiftUI에서 자동으로 사용 가능한 공간에 가장 알맞은 변형을 선택하도록 하는 ViewThatFits가 이제 SwiftUI에 포함되어 있습니다.

SwiftUI4Demo

Build Succeeded | Today at 10:26 PM

ContentView

```
3 // SwiftUI4Demo
4 //
5 // Created by Jongwook Park on 2022/11/14.
6 //
7
8 import SwiftUI
9
10 struct ContentView: View {
11     var body: some View {
12         Grid {
13             GridRow {
14                 Text("Office")
15                     .background(.red)
16
17                 Text("Xbox")
18                     .background(.green)
19             }
20
21             GridRow {
22                 Text("Windows")
23                     .background(.blue)
24
25                 Text("Bing")
26                     .background(.yellow)
27             }
28         }
29         .font(.largeTitle)
30     }
31 }
32
33 struct ContentView_Previews: PreviewProvider {
34     static var previews: some View {
35         ContentView()
36     }
37 }
38
```

Content View

Office Xbox
Windows Bing

Filter

Line: 29 Col: 26

SwiftUI4Demo

Build Succeeded | Today at 10:57 PM

ContentView

```
7
8 import SwiftUI
9
10 struct ContentView: View {
11     @State private var koreanScore: Int = 0
12     @State private var uruguayScore: Int = 0
13
14     var body: some View {
15         Grid {
16             GridRow {
17                 Text("Korea")
18
19                 ForEach(0..

Korea ● ● ● ●



Uruguay ● ● ●



Korea



Uruguay



Content View



Line: 15 Col: 15


```

The Economist | The Big Mac index | The Economist

economist.com/big-mac-index

Menu Weekly edition Search

Subscribe Log in

News | Burgernomics

The Big Mac index

Our interactive currency comparison tool

Jul 21st 2022 Share

The Big Mac index

Country	2000 — 2022	Under/over valued, %
Switzerland Franc		30.3
Norway Krone		21.6
Uruguay Peso		18.1
Sweden Krona		8.5
Canada C\$		2.0
United States US\$	BASE CURRENCY	
Lebanon Pound		-1.4
Israel Shekel		-4.0
UAE Dirham		-4.8
Euro area Euro		-7.5
Australia A\$		-10.2
Argentina Peso		-11.3
Saudi Arabia Riyal		-12.1

Choose a base currency Show index at

US dollar Jul 2022

Adjust the index to account for GDP per person

Raw index GDP-adjusted

The Swiss franc is 30.3% overvalued against the US dollar

July 2022

Overvalued Undervalued

Swiss franc US dollar

A Big Mac costs SFr6.50 in Switzerland and US\$5.15 in the United States. The implied exchange rate is 1.26. The difference between this and the actual exchange rate, 0.97, suggests the Swiss franc is 30.3% overvalued

Get Espresso from The Economist for essential global news. Start your one-month free trial today. [Discover more](#)

SwiftUI4Demo

ContentView

```
10 struct ContentView: View {  
14     var body: some View {  
15         NavigationStack {  
16             Grid {  
17                 GridRow {  
18                     Text("Switzerland")  
19                     Text("30.3")  
20                 }  
21  
22                 GridRow {  
23                     Text("Norway")  
24                     Text("21.6")  
25                 }  
26  
27                 GridRow {  
28                     Text("Sweden")  
29                     Text("8.5")  
30                 }  
31  
32                 GridRow {  
33                     Text("Canada")  
34                     Text("2.0")  
35                 }  
36  
37                 Divider()  
38  
39                 GridRow {  
40                     Text("United State is base currency")  
41                     .gridCellColumns(2)  
42                     .multilineTextAlignment(.trailing)  
43                 }  
44  
45                 Divider()  
46  
47                 GridRow {  
48                     Text("China")  
49                     Text("-30.9")  
50                 }  
51             }  
52         }  
53     }  
54 }
```

Content View

Big Mac index

Switzerland	30.3	
Norway	21.6	
Sweden	8.5	
Canada	2.0	
United State is base currency		
China	-30.9	
Korea	-32.0	
Japan	-45.1	

Build Succeeded | Today at 10:57 PM

+ Filter

Line: 41 Col: 44

SwiftUI4Demo

Build Succeeded | Today at 10:57 PM

```
ContentView
```

```
8 import SwiftUI
9
10 struct ContentView: View {
11     @State private var koreanScore: Int = 0
12     @State private var uruguayScore: Int = 0
13
14     var body: some View {
15         Grid(horizontalSpacing: 20, verticalSpacing: 20) {
16             GridRow {
17                 Image(systemName: "xmark")
18                 Image(systemName: "xmark")
19                 Image(systemName: "xmark")
20             }
21             GridRow {
22                 Image(systemName: "circle")
23                 Image(systemName: "xmark")
24                 Image(systemName: "circle")
25             }
26             GridRow {
27                 Image(systemName: "xmark")
28                 Image(systemName: "circle")
29                 Image(systemName: "circle")
30             }
31         }
32         .font(.title)
33     }
34
35
36 struct ContentView_Previews: PreviewProvider {
37     static var previews: some View {
38         ContentView()
39     }
40 }
41
```

Content View

The iPhone 14 Pro simulator displays a 3x3 grid of symbols. The first row contains three 'X' marks. The second row contains one circle, one 'X' mark, and one circle. The third row contains one 'X' mark, one circle, and one circle.



APP SCHOOL : iOS

SwiftUI 기초

Sheet

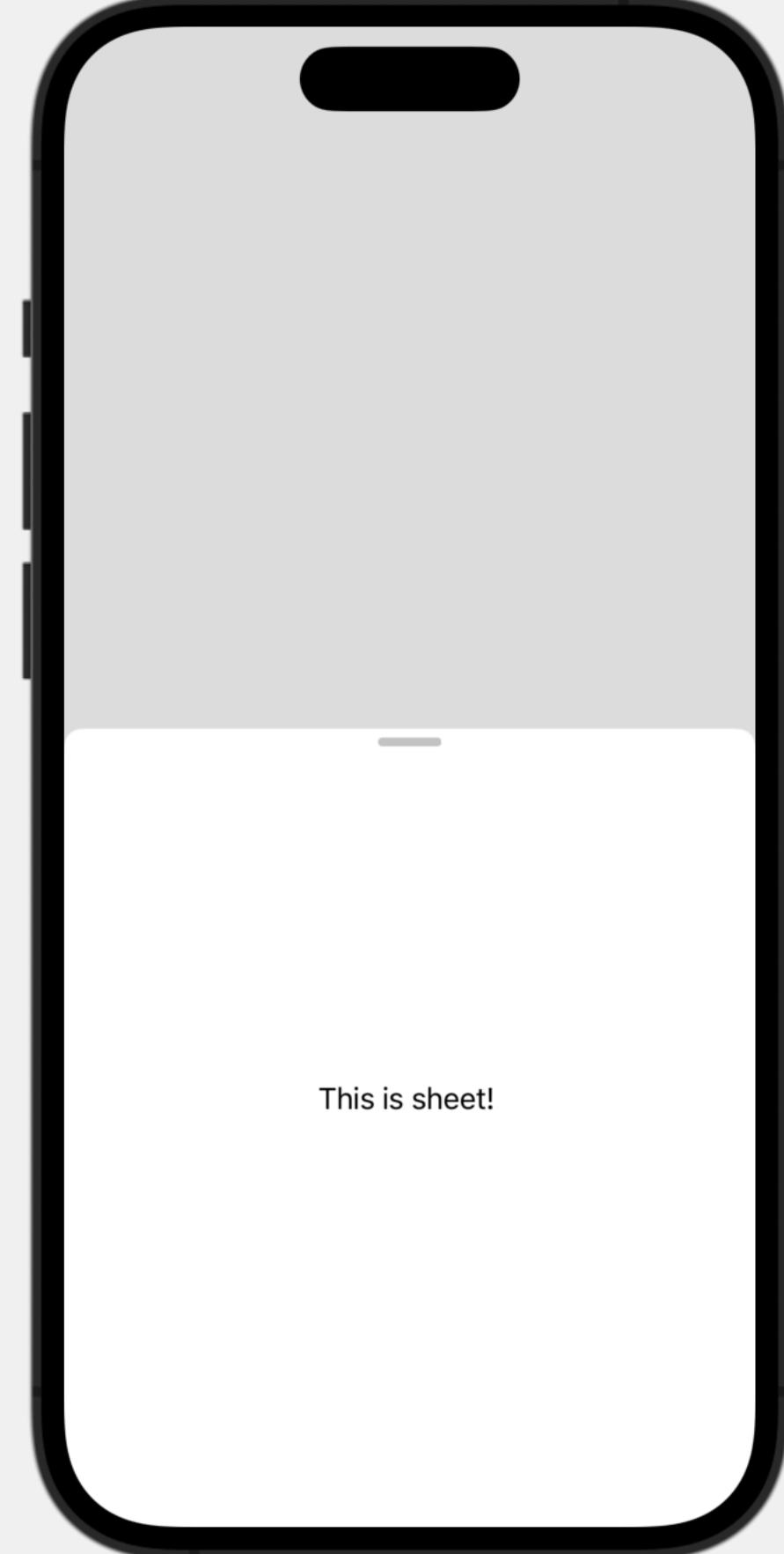
SwiftUI4Demo

Build Succeeded | Today at 10:26 PM

ContentView

```
5 // Created by Jongwook Park on 2022/11/14.
6 //
7
8 import SwiftUI
9
10 struct ContentView: View {
11     @State private var showingSheet = false
12
13     var body: some View {
14         Button("Show sheet") {
15             showingSheet.toggle()
16         }
17         .sheet(isPresented: $showingSheet) {
18             Text("This is sheet! ")
19             .presentationDetents([.medium, .large])
20         }
21     }
22 }
23
24 struct ContentView_Previews: PreviewProvider {
25     static var previews: some View {
26         ContentView()
27     }
28 }
```

This is sheet!



Filter

Line: 19 Col: 54

SwiftUI4Demo

Build Succeeded | Today at 10:26 PM

ContentView

```
5 // Created by Jongwook Park on 2022/11/14.
6 //
7
8 import SwiftUI
9
10 struct ContentView: View {
11     @State private var showingSheet = false
12
13     var body: some View {
14         Button("Show sheet") {
15             showingSheet.toggle()
16         }
17         .sheet(isPresented: $showingSheet) {
18             Text("This is sheet! ")
19                 .presentationDetents([.medium, .large])
20                 .presentationDragIndicator(.hidden)
21         }
22     }
23 }
24
25 struct ContentView_Previews: PreviewProvider {
26     static var previews: some View {
27         ContentView()
28     }
29 }
```

Content View

This is sheet!

SwiftUI4Demo

Build Succeeded | Today at 10:26 PM

ContentView

```
5 // Created by Jongwook Park on 2022/11/14.
6 //
7
8 import SwiftUI
9
10 struct ContentView: View {
11     @State private var showingSheet = false
12
13     var body: some View {
14         Button("Show sheet") {
15             showingSheet.toggle()
16         }
17         .sheet(isPresented: $showingSheet) {
18             Text("This is sheet! ")
19             .presentationDetents([.height(140)])
20         }
21     }
22 }
23
24 struct ContentView_Previews: PreviewProvider {
25     static var previews: some View {
26         ContentView()
27     }
28 }
```

Show sheet

This is sheet!

The screenshot shows a Xcode interface with a SwiftUI project named "SwiftUI4Demo". The code editor displays the ContentView.swift file, which contains code for displaying a sheet modal. The modal is shown on an iPhone 14 Pro simulator, with the text "This is sheet!" visible inside it. The status bar at the top indicates a successful build at 10:26 PM. The bottom right corner of the screen shows the Xcode navigation bar with icons for search, refresh, and other functions.

SwiftUI4Demo

Build Succeeded | Today at 10:26 PM

ContentView

```
5 // Created by Jongwook Park on 2022/11/14.
6 //
7
8 import SwiftUI
9
10 struct ContentView: View {
11     @State private var showingSheet = false
12
13     var body: some View {
14         Button("Show sheet") {
15             showingSheet.toggle()
16         }
17         .sheet(isPresented: $showingSheet) {
18             Text("This is sheet! ")
19             .presentationDetents([.fraction(0.35)])
20         }
21     }
22 }
23
24 struct ContentView_Previews: PreviewProvider {
25     static var previews: some View {
26         ContentView()
27     }
28 }
```

Show sheet

This is sheet!

The screenshot shows the Xcode interface with a SwiftUI project named "SwiftUI4Demo". The code editor displays the ContentView.swift file, which contains code for creating a sheet modal. The preview window on the right shows an iPhone 14 Pro simulator displaying the sheet. The sheet has a gray header bar with the text "Show sheet" and a white content area with the text "This is sheet!". The status bar at the top of the simulator screen also displays the text "Show sheet".



APP SCHOOL : iOS

SwiftUI 기초

Grid



새로운 Sharing API

새로운 Transferable 프로토콜 덕분에 클립보드,
드래그 앤 드롭 및 공유 시트에서 데이터를 사용할 수
있으며, 이제 SwiftUI를 사용하여 직접 호출할 수
있습니다.

developer.apple.com

Apple Developer News Discover Design Develop Distribute Support Account

Documentation / SwiftUI / Controls and indicators / ShareLink

Language: Swift API Changes: Show ▾

SwiftUI

ShareLink

- Sharing an item
 - init(item: String, subject: Text?, message: Text?)
 - init(item: URL, subject: Text?, message: Text?)
 - init(item: String, subject: Text?, message: Text?, label: () -> L...)
 - init(item: URL, subject: Text?, message: Text?, label: () -> Lab...
- Sharing an item with a preview
 - init<l>(item: I, subject: Text?, message: Text?, preview: Share...
 - init<l>(item: I, subject: Text?, message: Text?, preview: Share...
- SharePreview
- Sharing an item with a label
 - init<S>(S, item: URL, subject: Text?, message: Text?)
 - init(LocalizedStringKey, item: URL, subject: Text?, message: T...)
 - init(Text, item: String, subject: Text?, message: Text?)
 - init<S>(S, item: String, subject: Text?, message: Text?)
 - init(LocalizedStringKey, item: String, subject: Text?, message:...)
 - init(Text, item: URL, subject: Text?, message: Text?)
- Sharing an item with a label and a preview
 - init<S, I>(S, item: I, subject: Text?, message: Text?, preview: S...)
 - init<l>(Text, item: I, subject: Text?, message: Text?, preview: ...)
 - init<l>(LocalizedStringKey, item: I, subject: Text?, message: T...)
- SharePreview
- Sharing items
 - init(items: Data, subject: Text?, message: Text?)
 - init(items: Data, subject: Text?, message: Text?)
 - init(items: Data, subject: Text?, message: Text?, label: () -> La...

Filter

Structure

ShareLink

A view that controls a sharing presentation.

iOS 16.0+ iPadOS 16.0+ macOS 13.0+ Mac Catalyst 16.0+ watchOS 9.0+

Declaration

```
struct ShareLink<Data, PreviewImage, PreviewIcon, Label> where Data : RandomAccessCollection, PreviewImage : View, PreviewIcon : Image, Label : View
```

Overview

People tap or click on a share link to present a share interface. The link typically uses a system-standard appearance for the link; you only need to supply the content to share:

```
ShareLink(item: URL(string: "https://developer.apple.com/xcode/swiftui/")!)
```

You can control the appearance of the link by providing view content. For example, you can use a `Label` to display a link with a custom icon:

```
ShareLink(item: URL(string: "https://developer.apple.com/xcode/swiftui/")!) {  
    Label("Share", image: "MyCustomShareIcon")  
}
```

If you only wish to customize the link's title, you can use one of the convenience initializers that takes a string and creates a `Label` for you:

```
ShareLink("Share URL", item: URL(string: "https://developer.apple.com/xcode/swiftui/")!)
```

The link can share any content that is `Transferable`. Many framework types, like `URL`, already conform to



APP SCHOOL : iOS

SwiftUI 기초

PhotoUI

Apple What's new in th... developer.apple.com

Apple Developer News Discover Design Develop Distribute Support Account

Documentation / PhotoKit / Selecting Photos and Videos in iOS Language: Swift API Changes: None

PhotoKit

- Shared Photo Library
 - Delivering an Enhanced Privacy Experience in Your Photos App
 - Browsing and Modifying Photo Albums
- > **C** PHPhotoLibrary
- Photos Picker
 - {}** Selecting Photos and Videos in iOS
- > **C** PHPickerViewController
- Photos Picker View
 - {}** Bringing Photos picker to your SwiftUI app
- > **S** PhotosPicker
- > **S** PhotosPickerItem
- > **S** PhotosPickerSelectionBehavior
- Asset Retrieval
 - Fetching Objects and Requesting Changes
- > **C** PHAsset
- > **C** PHAssetCollection
- > **C** PHCollection
- > **C** PHCollectionList
- > **C** PHObject
- > **C** PHFetchResult
- > **C** PHFetchOptions
- Asset Loading
 - Loading and Caching Assets and Thumbnails
- > **C** PHImageManager
- > **C** PHCachingImageManager
- > **C** PHImageRequestOptions

Sample Code

Selecting Photos and Videos in iOS

Improve the user experience of finding and selecting assets by using the Photos picker.

[Download](#)

Photos PhotosUI iOS 15.0+ iPadOS 15.0+ Xcode 13.0+

Overview

This sample shows how to use the [PhotoKit](#) Photos picker, and illustrates how to filter assets according to the user's selection. After you have a selection, you use an item provider to load the asset to display in the app. The sample also explores the new configuration options in iOS 15.

Note

This sample code project is associated with WWDC21 session [10046: Improve access to Photos in your app](#).

Understand the Photos Picker Benefits

Both [PHPickerViewController](#) and [UIImagePickerController](#) use the out-of-process Photos picker user interface. The PHPickerViewController contains a powerful set of APIs that make it a great alternative to UIImagePickerController. PHPickerViewController improves stability and reliability, and includes several benefits to developers and users, such as the following:

- Deferred image loading and recovery UI
- Reliable handling of large and complex assets, like RAW and panoramic images
- User-selectable assets that aren't available for UIImagePickerController
- Configuration of the picker to only display Live Photos

What's new in th... developer.apple.com

Documentation / PhotoKit / Selecting Photos and Videos in iOS

Language: Swift API Changes: None

PhotoKit

- Shared Photo Library
 - Delivering an Enhanced Privacy Experience in Your Photos App
 - Browsing and Modifying Photo Albums
- > **C** PHPhotoLibrary
- Photos Picker
 - {} Selecting Photos and Videos in iOS
- > **C** PHPickerViewController
 - Photos Picker View
 - {} Bringing Photos picker to your SwiftUI app
- > **S** PhotosPicker
- > **S** PhotosPickerItem
- > **S** PhotosPickerSelectionBehavior
- Asset Retrieval
 - Fetching Objects and Requesting Changes
- > **C** PHAsset
- > **C** PHAssetCollection
- > **C** PHCollection
- > **C** PHCollectionList
- > **C** PHObject
- > **C** PHFetchResult
- > **C** PHFetchOptions
- Asset Loading
 - Loading and Caching Assets and Thumbnails
- > **C** PHImageManager
- > **C** PHCachingImageManager
- > **C** PHImageRequestOptions
- > **C** PHVideoRequestOptions
- > **C** PHLivePhotoRequestOptions

Filter

Configure the Photos Picker

Before displaying the photo library, the sample creates a `PHPickerConfiguration` object using the shared photo library. Creating a configuration without a photo library provides only asset data, and doesn't include asset identifiers.

```
var configuration = PHPickerConfiguration(photoLibrary: .shared())
```

The picker displays all asset types by default. A filter configures the picker to display videos, images with live photos, or live photos only. The sample contains three buttons that reflect the available filter types.

An app creates custom filters by combining filter options. For example, the following code displays live photos and videos:

```
var newFilter = PHPickerFilter.any(of: [.livePhotos, .videos])
```

A new feature in iOS 15 is the capability to change the selection behavior. The sample sets the selection property to `ordered`, which places a numbered checkmark when selecting items, and disables swipe to select.

The preselection API allows for presenting the picker with selected photos. This gives the user the opportunity to select more photos, or to deselect preselected photos.

```
// Set the filter type according to the user's selection.  
configuration.filter = filter  
// Set the mode to avoid transcoding, if possible, if your app supports arbitrary image.  
configuration.preferredAssetRepresentationMode = .current  
// Set the selection behavior to respect the user's selection order.  
configuration.selection = .ordered  
// Set the selection limit to enable multiselection.  
configuration.selectionLimit = 0  
// Set the preselected asset identifiers with the identifiers that the app tracks.  
configuration.preselectedAssetIdentifiers = selectedAssetIdentifiers
```

Display a Picker with the Configuration

developer.apple.com

Selecting Photos...

Apple Developer News Discover Design Develop Distribute Support Account

Videos Collections Topics All Videos About

Back to Search

import SwiftUI
import PhotosUI

```
struct ContentView: View {  
    @Binding selection: [PhotosPickerItem]  
  
    var body: some View {  
        PhotosPicker(  
            selection: $selection,  
            matching: .images  
        ) {  
            Text("Select Photos")  
        }  
    }  
}
```

iPhone screen showing the Photos app interface with a grid of photos and a "Show Selected (3)" button at the bottom.

피커가 지원되는 모든 플랫폼에서요

08:03 14:30

Overview Transcript

What's new in the Photos picker

PHPicker provides simple and secure integration between your app and the system Photos library. Learn how SwiftUI and Transferable can help you offer integration across iOS, iPadOS, macOS, and watchOS. We'll also show you how you can use AppKit and NSOpenPanel to bring the Photos picker on Mac into your macOS apps. For even more on the Photos picker, watch "Improve access to Photos in your app" from WWDC21.

Resources

Have a question? Ask with tag [wwdc2022-10023](#)



APP SCHOOL : iOS

SwiftUI 기초

Swift Charts



Swift Charts

Apple 플랫폼 어디에서나 멋지게 선보일 수 있는
맞춤화가 가능한 차트를 사용하여 데이터를 시각화해
보십시오. Swift Charts는 SwiftUI의 작문적
구문을 사용하여 선 및 막대 차트부터 스트림 그래프와
같은 고급 유형까지 다양한 가능성을 내포한 보기들
생성합니다. 또한 Swift Charts에서는
VoiceOver를 지원하여 모든 사용자에게 정보를
제공할 수 있습니다.

Apple developer.com

News Discover Design Develop Distribute Support Account

Documentation / Swift Charts Language: Swift API Changes: Show ▾

Swift Charts

Framework

Swift Charts

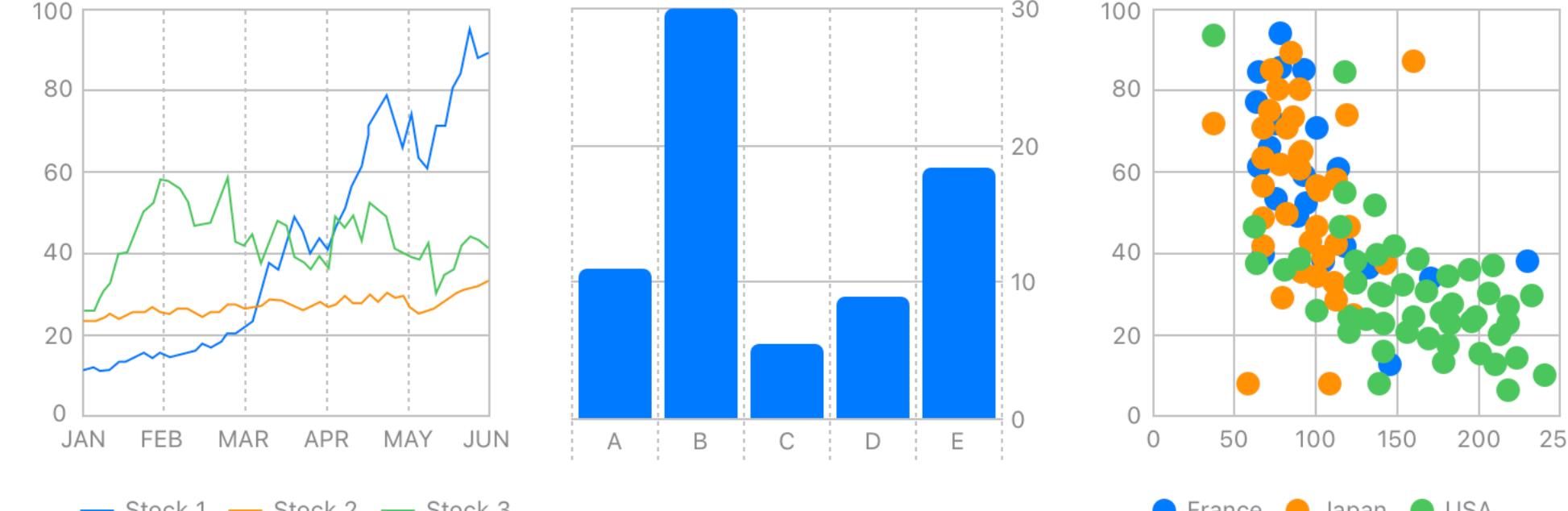
Construct and customize charts on every Apple platform.

iOS 16.0+ iPadOS 16.0+ macOS 13.0+ Mac Catalyst 16.0+ tvOS 16.0+ watchOS 9.0+



Overview

Swift Charts is a powerful and concise SwiftUI framework for transforming your data into informative visualizations. With Swift Charts, you can build effective and customizable charts with minimal code. This framework provides marks, scales, axes, and legends as building blocks that you can combine to develop a broad range of data-driven charts.



There are many ways you can use Swift Charts to communicate patterns or trends in your data. You can create a variety of charts including line charts, bar charts, and scatter plots as shown above. When you create a chart using this framework, it automatically generates scales and axes that fit your data.

Charts

- Creating a chart using Swift C...
- Visualizing your app's data
- > Chart
- > ChartContent
- > ChartContentBuilder
- > Plot

Marks

- > AreaMark
- > LineMark
- > PointMark
- > RectangleMark
- > RuleMark
- > BarMark

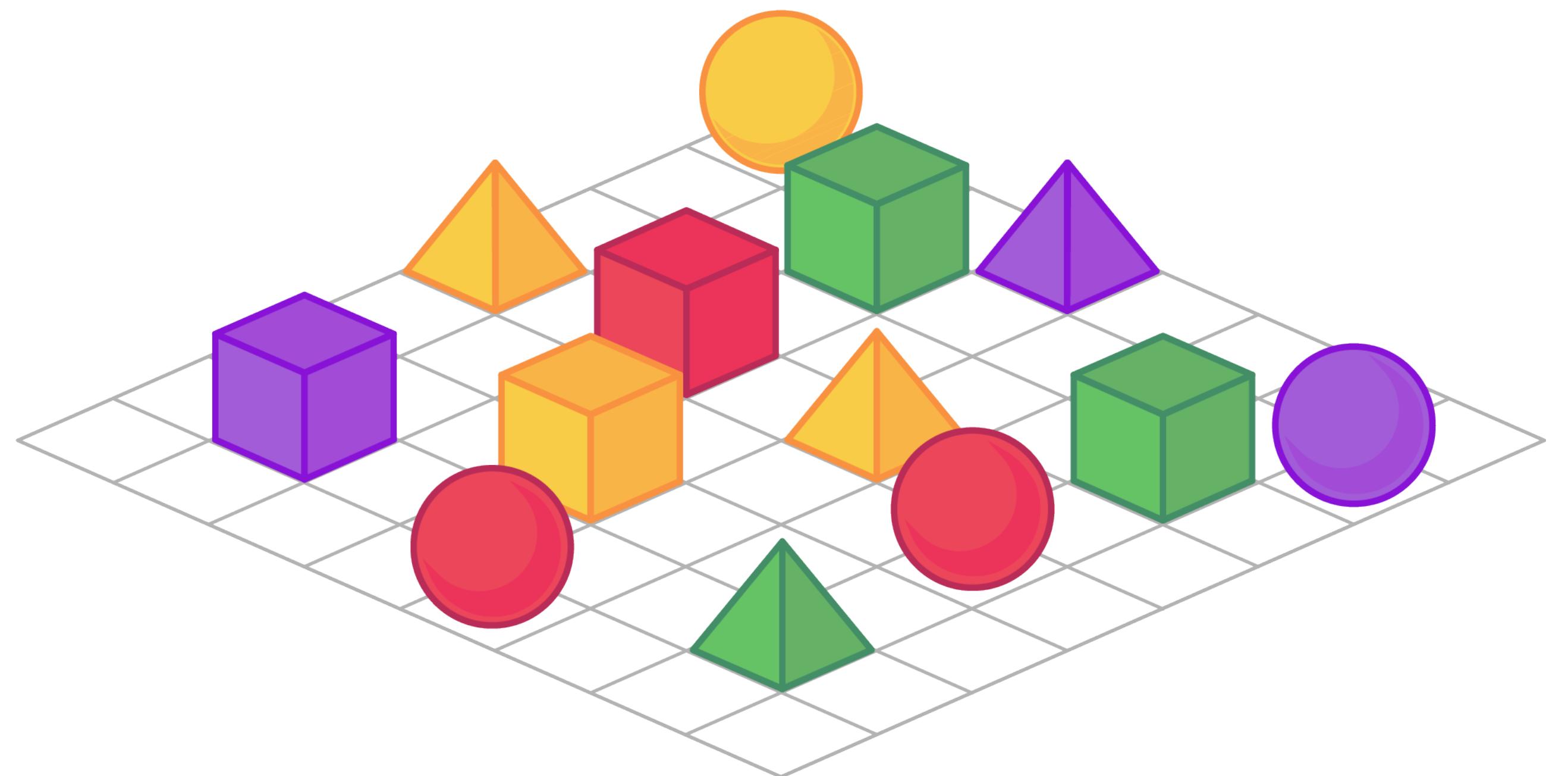
Mark configuration

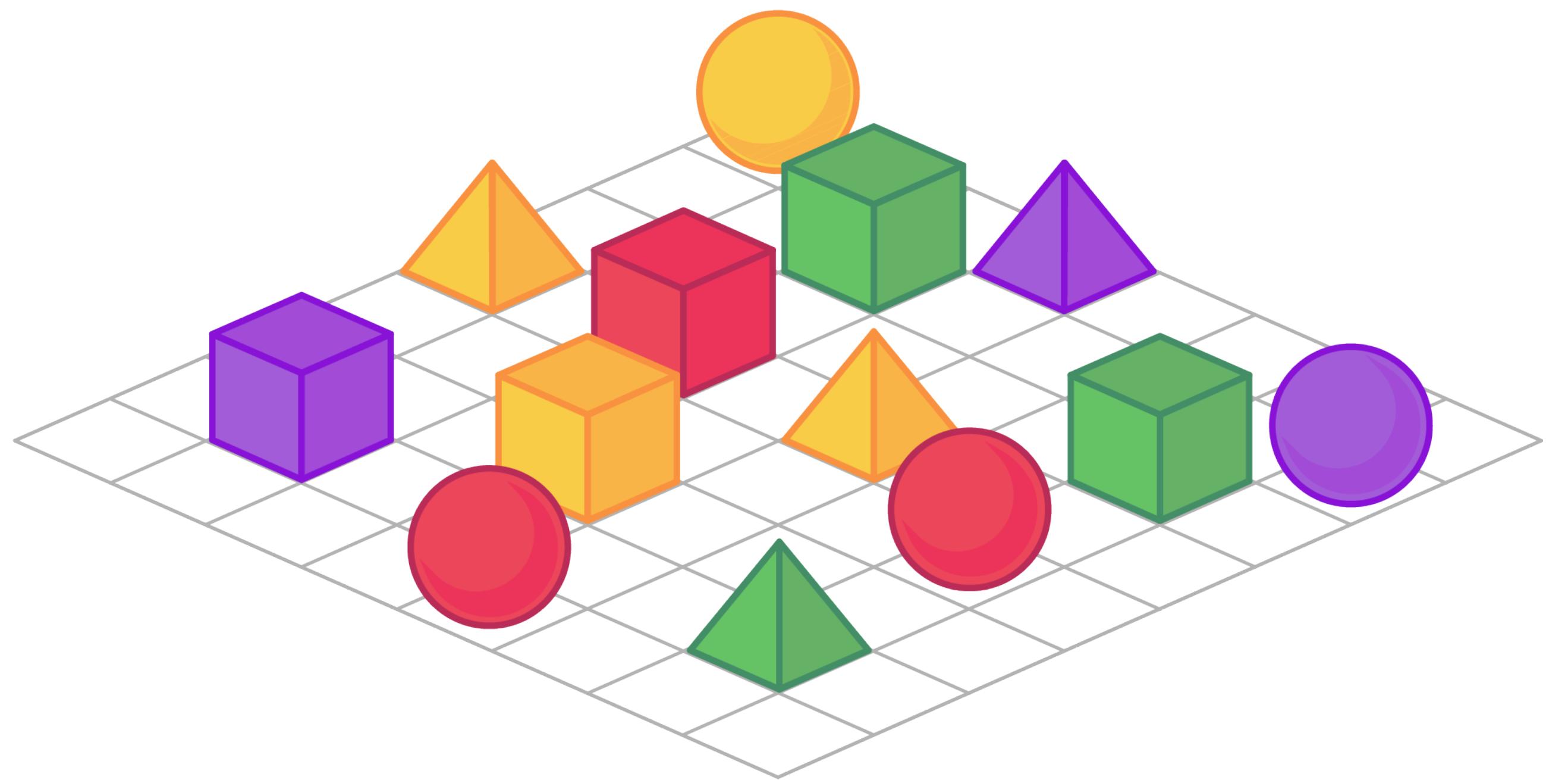
- > MarkStackingMethod
- > MarkDimension
- > InterpolationMethod
- > ChartSymbolShape
- > AnyChartSymbolShape

Labeled data

Filter

각 모양과
색상별로
몇 개씩 있을까요?





	Cube	Sphere	Pyramid	Total
Pink	1	2	0	3
Yellow	1	1	2	4
Purple	1	1	1	3
Green	2	0	1	3
Total	5	4	4	13

Define the data source

ToyCharts

ToyCharts > iPhone 14 Pro Connecting to ned_AppleWatch_SE via ned_iPhone_14_Pro for development

ToyShape

```
1 //  
2 // ToyShape.swift  
3 // ToyCharts  
4 //  
5 // Created by Jongwook Park on 2022/11/16.  
6 //  
7  
8 import Foundation  
9  
10 struct ToyShape: Identifiable {  
11     var type: String  
12     var count: Double  
13     var id = UUID()  
14 }  
15  
16  
17
```

+ Filter

Line: 10 Col: 32

ToyCharts

ToyCharts > iPhone 14 Pro

ToyCharts: Ready | Today at 3:06 PM

ToyShape

ToyCharts > ToyCharts > ToyShape > No Selection

```
1 //  
2 // ToyShape.swift  
3 // ToyCharts  
4 //  
5 // Created by Jongwook Park on 2022/11/16.  
6 //  
7  
8 import Foundation  
9  
10 struct ToyShape: Identifiable {  
11     var type: String  
12     var count: Double  
13     var id = UUID()  
14 }  
15  
16 var data: [ToyShape] = [  
17     .init(type: "Cube", count: 5),  
18     .init(type: "Sphere", count: 4),  
19     .init(type: "Pyramid", count: 4)  
20 ]  
21
```

+ Filter

Line: 16 Col: 25

**Initialize a chart view
and create marks**

ToyCharts

ToyCharts > iPhone 14 Pro

ToyCharts: Ready | Today at 3:07 PM

ToyShape ContentView

ToyCharts > ToyCharts > ContentView > body

```
1 //  
2 // ContentView.swift  
3 // ToyCharts  
4 //  
5 // Created by Jongwook Park on 2022/11/16.  
6 //  
7  
8 import SwiftUI  
9 import Charts  
10  
11 struct ContentView: View {  
12     var body: some View {  
13         Chart {  
14             BarMark(  
15                 x: .value("Shape Type", data[0].type),  
16                 y: .value("Total Count", data[0].count)  
17             )  
18             BarMark(  
19                 x: .value("Shape Type", data[1].type),  
20                 y: .value("Total Count", data[1].count)  
21             )  
22             BarMark(  
23                 x: .value("Shape Type", data[2].type),  
24                 y: .value("Total Count", data[2].count)  
25             )  
26         }  
27     }  
28 }  
29  
30 struct ContentView_Previews: PreviewProvider {  
31     static var previews: some View {  
32         ContentView()  
33     }  
34 }  
35
```

Content View

The screenshot shows the Xcode interface with the ToyCharts project open. The ContentView.swift file is the active code editor. The code defines a SwiftUI View named ContentView that contains a Chart with three blue bars. The bars represent the total count for three shapes: Cube, Sphere, and Pyramid. The simulator window displays the iPhone 14 Pro with the chart view. The status bar at the top right indicates 'ToyCharts: Ready | Today at 3:07 PM'.

ToyCharts

ToyCharts > iPhone 14 Pro

Build Succeeded | Today at 3:08 PM

ToyShape ContentView

ToyCharts > ToyCharts > ContentView > body

```
1 //  
2 // ContentView.swift  
3 // ToyCharts  
4 //  
5 // Created by Jongwook Park on 2022/11/16.  
6 //  
7  
8 import SwiftUI  
9 import Charts  
10  
11 struct ContentView: View {  
12     var body: some View {  
13         Chart {  
14             ForEach(data) { shape in  
15                 BarMark(  
16                     x: .value("Shape Type", shape.type),  
17                     y: .value("Total Count", shape.count)  
18                 )  
19             }  
20         }  
21     }  
22 }  
23  
24 struct ContentView_Previews: PreviewProvider {  
25     static var previews: some View {  
26         ContentView()  
27     }  
28 }
```

Content View

The image shows an iPhone 14 Pro simulator displaying a bar chart. The chart has three blue bars representing different shapes: Cube, Sphere, and Pyramid. The x-axis labels are "Cube", "Sphere", and "Pyramid". The y-axis has numerical tick marks at 0, 2, 4, and 6. The bars have heights corresponding to their total count.

Line: 18 Col: 18

Explore additional data properties

The screenshot shows the Xcode interface with the following details:

- Project Navigator:** Shows the project structure under "ToyCharts". The "ToyShape" file is selected.
- Editor:** Displays the "ToyShape.swift" file content. The code defines a struct "ToyShape" with properties: color, type, count, and id. It also initializes a list of "ToyShape" objects with various colors, types, and counts.
- Build Status:** "Build Succeeded | Today at 3:08 PM"
- File Inspector:** Shows the file path: ToyCharts > ToyCharts > ToyShape.swift
- Document Outline:** Shows the file path: ToyCharts > ToyCharts > ToyShape > ToyShape
- Bottom Bar:** Includes buttons for Filter, Run, Stop, and a status bar indicating "Line: 11 Col: 22".

```
1 //  
2 // ToyShape.swift  
3 // ToyCharts  
4 //  
5 // Created by Jongwook Park on 2022/11/16.  
6 //  
7  
8 import Foundation  
9  
10 struct ToyShape: Identifiable {  
11     var color: String|  
12     var type: String  
13     var count: Double  
14     var id = UUID()  
15 }  
16  
17 var stackedBarData: [ToyShape] = [  
18     .init(color: "Green", type: "Cube", count: 2),  
19     .init(color: "Green", type: "Sphere", count: 0),  
20     .init(color: "Green", type: "Pyramid", count: 1),  
21     .init(color: "Purple", type: "Cube", count: 1),  
22     .init(color: "Purple", type: "Sphere", count: 1),  
23     .init(color: "Purple", type: "Pyramid", count: 1),  
24     .init(color: "Pink", type: "Cube", count: 1),  
25     .init(color: "Pink", type: "Sphere", count: 2),  
26     .init(color: "Pink", type: "Pyramid", count: 0),  
27     .init(color: "Yellow", type: "Cube", count: 1),  
28     .init(color: "Yellow", type: "Sphere", count: 1),  
29     .init(color: "Yellow", type: "Pyramid", count: 2)  
30 ]  
31
```

ToyCharts

ToyCharts > iPhone 14 Pro

Build Succeeded | Today at 3:08 PM

ToyShape ContentView

ToyCharts > ToyCharts > ContentView > body

```
1 //  
2 // ContentView.swift  
3 // ToyCharts  
4 //  
5 // Created by Jongwook Park on 2022/11/16.  
6 //  
7  
8 import SwiftUI  
9 import Charts  
10  
11 struct ContentView: View {  
12     var body: some View {  
13         Chart {  
14             ForEach(stackedBarData) { shape in  
15                 BarMark(  
16                     x: .value("Shape Type", shape.type),  
17                     y: .value("Total Count", shape.count)  
18                 )  
19                     .foregroundStyle(by: .value("Shape Color", shape.color))  
20             }  
21         }  
22     }  
23 }  
24  
25 struct ContentView_Previews: PreviewProvider {  
26     static var previews: some View {  
27         ContentView()  
28     }  
29 }  
30
```

Content View

iPhone 14 Pro

Build Succeeded | Today at 3:08 PM

Line: 19 Col: 41

Customize your chart

ToyCharts

ToyCharts > iPhone 14 Pro

Build Succeeded | Today at 3:10 PM

ToyShape ContentView

ToyCharts > ToyCharts > ContentView > body

```
1 //  
2 // ContentView.swift  
3 // ToyCharts  
4 //  
5 // Created by Jongwook Park on 2022/11/16.  
6 //  
7  
8 import SwiftUI  
9 import Charts  
10  
11 struct ContentView: View {  
12     var body: some View {  
13         Chart {  
14             ForEach(stackedBarData) { shape in  
15                 BarMark(  
16                     x: .value("Shape Type", shape.type),  
17                     y: .value("Total Count", shape.count)  
18                 )  
19                     .foregroundStyle(by: .value("Shape Color", shape.color))  
20             }  
21         }  
22         .chartForegroundStyleScale([  
23             "Green": .green, "Purple": .purple, "Pink": .pink, "Yellow":  
24                 .yellow  
25         ])  
26     }  
27  
28 struct ContentView_Previews: PreviewProvider {  
29     static var previews: some View {  
30         ContentView()  
31     }  
32 }  
33
```

Content View

Line: 22 Col: 37

또 다른 예제

ToyCharts

ToyCharts > iPhone 14 Pro Connecting to ned_AppleWatch_SE via ned_iPhone_14_Pro for development

ToyShape | ChartView | **ChartMoreView** | ContentView | ToyChartsApp |

ToyCharts > ToyCharts > ChartMoreView > No Selection

```
1 //  
2 //  ChartMoreView.swift  
3 //  ToyCharts  
4 //  
5 //  Created by Jongwook Park on 2022/11/16.  
6 //  
7  
8 import SwiftUI  
9 import Charts
```

10
11 struct ChartMoreView: View {
12 var body: some View {
13 Chart {
14 BarMark(
15 x: .value("Day", "Monday"),
16 y: .value("Steps", 6019)
17)
18
19 BarMark(
20 x: .value("Day", "Tuesday"),
21 y: .value("Steps", 7200)
22)
23 }
24 .padding()
25 .navigationTitle("Weekly Report")
26 }
27 }
28
29 struct ChartMoreView_Previews: PreviewProvider {
30 static var previews: some View {
31 NavigationStack {
32 ChartMoreView()
33 }
34 }
35 }
36

Chart More View

iPhone 14 Pro

Weekly Report

Monday Tuesday

7,500
5,000
2,500
0

Line: 9 Col: 14

ToyCharts

ToyCharts > iPhone 14 Pro

Build Succeeded | Today at 3:10 PM

ToyShape | ChartView | **ChartMoreView** | ContentView | ToyChartsApp |

ToyCharts > ToyCharts > ChartMoreView > body

```
1 //  
2 //  ChartMoreView.swift  
3 //  ToyCharts  
4 //  
5 //  Created by Jongwook Park on 2022/11/16.  
6 //  
7  
8 import SwiftUI  
9 import Charts  
10  
11 struct ChartMoreView: View {  
12     let weekdays = Calendar.current.shortWeekdaySymbols  
13     let steps = [ 10531, 6019, 7200, 8311, 7403, 6503, 9230 ]  
14  
15     var body: some View {  
16         Chart {  
17             ForEach (weekdays.indices, id: \.self) { index in  
18                 BarMark(  
19                     x: .value("Day", weekdays[index]),  
20                     y: .value("Steps", steps[index])  
21                 )  
22             }  
23         }  
24         .padding()  
25         .navigationTitle("Weekly Report")  
26     }  
27 }  
28  
29 struct ChartMoreView_Previews: PreviewProvider {  
30     static var previews: some View {  
31         NavigationStack {  
32             ChartMoreView()  
33         }  
34     }  
35 }  
36
```

Chart More View

Weekly Report

Day	Steps
Sun	10531
Mon	6019
Tue	7200
Wed	8311
Thu	7403
Fri	6503
Sat	9230

Line: 17 Col: 62

ToyCharts

ToyCharts > iPhone 14 Pro

Build Succeeded | Today at 3:10 PM

ToyShape | ChartView | **ChartMoreView** | ContentView | ToyChartsApp |

ToyCharts > ToyCharts > ChartMoreView > body

```
1 //  
2 //  ChartMoreView.swift  
3 //  ToyCharts  
4 //  
5 //  Created by Jongwook Park on 2022/11/16.  
6 //  
7  
8 import SwiftUI  
9 import Charts  
10  
11 struct ChartMoreView: View {  
12     let weekdays = Calendar.current.shortWeekdaySymbols  
13     let steps = [ 10531, 6019, 7200, 8311, 7403, 6503, 9230 ]  
14  
15     var body: some View {  
16         Chart {  
17             ForEach (weekdays.indices, id: \.self) { index in  
18                 BarMark(  
19                     x: .value("Day", weekdays[index]),  
20                     y: .value("Steps", steps[index])  
21                 )  
22                     .foregroundStyle(by: .value("Day", weekdays[index]))  
23             }  
24         }  
25         .padding()  
26         .navigationTitle("Weekly Report")  
27     }  
28 }  
29  
30 struct ChartMoreView_Previews: PreviewProvider {  
31     static var previews: some View {  
32         NavigationStack {  
33             ChartMoreView()  
34         }  
35     }  
36 }  
37
```

Chart More View

Weekly Report

Day	Steps
Sun	10531
Mon	6019
Tue	7200
Wed	8311
Thu	7403
Fri	6503
Sat	9230

Line: 22 Col: 17

ToyCharts

ToyCharts > iPhone 14 Pro

Build Succeeded | Today at 3:10 PM

ToyShape | ChartView | **ChartMoreView** | ContentView | ToyChartsApp |

ToyCharts > ToyCharts > ChartMoreView > body

```
1 //  
2 //  ChartMoreView.swift  
3 //  ToyCharts  
4 //  
5 //  Created by Jongwook Park on 2022/11/16.  
6 //  
7  
8 import SwiftUI  
9 import Charts  
10  
11 struct ChartMoreView: View {  
12     let weekdays = Calendar.current.shortWeekdaySymbols  
13     let steps = [ 10531, 6019, 7200, 8311, 7403, 6503, 9230 ]  
14  
15     var body: some View {  
16         Chart {  
17             ForEach (weekdays.indices, id: \.self) { index in  
18                 BarMark(  
19                     x: .value("Day", weekdays[index]),  
20                     y: .value("Steps", steps[index])  
21                 )  
22                     .foregroundStyle(by: .value("Day", weekdays[index]))  
23                     .annotation {  
24                         Text("\(steps[index])")  
25                     }  
26                 }  
27             }  
28             .padding()  
29             .navigationTitle("Weekly Report")  
30         }  
31     }  
32  
33 struct ChartMoreView_Previews: PreviewProvider {  
34     static var previews: some View {  
35         NavigationStack {  
36             ChartMoreView()  
37         }  
38     }  
39 }
```

Chart More View

Weekly Report

The chart displays the following data:

Day	Steps
Sun	10,531
Mon	6,019
Tue	7,200
Wed	8,311
Thu	7,403
Fri	6,503
Sat	9,230

Line: 23 Col: 30

ToyCharts

ToyCharts > iPhone 14 Pro

Build Succeeded | Today at 3:10 PM

ToyShape | ChartView | **ChartMoreView** | ContentView | ToyChartsApp |

ToyCharts > ToyCharts > ChartMoreView > body

```
1 //  
2 //  ChartMoreView.swift  
3 //  ToyCharts  
4 //  
5 //  Created by Jongwook Park on 2022/11/16.  
6 //  
7  
8 import SwiftUI  
9 import Charts  
10  
11 struct ChartMoreView: View {  
12     let weekdays = Calendar.current.shortWeekdaySymbols  
13     let steps = [ 10531, 6019, 7200, 8311, 7403, 6503, 9230 ]  
14  
15     var body: some View {  
16         Chart {  
17             ForEach (weekdays.indices, id: \.self) { index in  
18                 BarMark(  
19                     x: .value("Steps", steps[index]),  
20                     y: .value("Day", weekdays[index])  
21                 )  
22                     .foregroundStyle(by: .value("Day", weekdays[index]))  
23                     .annotation {  
24                         Text("\(steps[index])")  
25                     }  
26                 }  
27             }  
28             .padding()  
29             .navigationTitle("Weekly Report")  
30         }  
31     }  
32  
33 struct ChartMoreView_Previews: PreviewProvider {  
34     static var previews: some View {  
35         NavigationStack {  
36             ChartMoreView()  
37         }  
38     }  
39 }
```

Chart More View

Weekly Report

Day	Steps
Sun	10,531
Mon	6,019
Tue	7,200
Wed	8,311
Thu	7,403
Fri	6,503
Sat	9,230

0 5,000 10,000

● Sun ● Mon ● Tue ● Wed ● Thu ● Fri ● Sat

Filter

2 lines

ToyCharts

ToyCharts > iPhone 14 Pro Connecting to ned_AppleWatch_SE via ned_iPhone_14_Pro for development

ToyShape | ChartView | **ChartMoreView** | ContentView | ToyChartsApp |

ToyCharts > ToyCharts > ChartMoreView > body

```
1 //  
2 //  ChartMoreView.swift  
3 //  ToyCharts  
4 //  
5 //  Created by Jongwook Park on 2022/11/16.  
6 //  
7  
8 import SwiftUI  
9 import Charts  
10  
11 struct ChartMoreView: View {  
12     let weekdays = Calendar.current.shortWeekdaySymbols  
13     let steps = [ 10531, 6019, 7200, 8311, 7403, 6503, 9230 ]  
14  
15     var body: some View {  
16         Chart {  
17             ForEach (weekdays.indices, id: \.self) { index in  
18                 BarMark(  
19                     x: .value("Day", weekdays[index]),  
20                     y: .value("Steps", steps[index]))  
21                 .foregroundStyle(by: .value("Day", weekdays[index]))  
22                 .annotation {  
23                     Text("\(steps[index])")  
24                 }  
25  
26                 LineMark(  
27                     x: .value ("Day", weekdays[index]),  
28                     y: .value ("Steps", steps[index]))  
29                 .foregroundStyle (Color.indigo)  
30                 .lineStyle (StrokeStyle(lineWidth: 4.0))  
31             }  
32         }  
33     }  
34     .padding()  
35     .navigationTitle("Weekly Report")  
36 }  
37 }
```

Chart More View

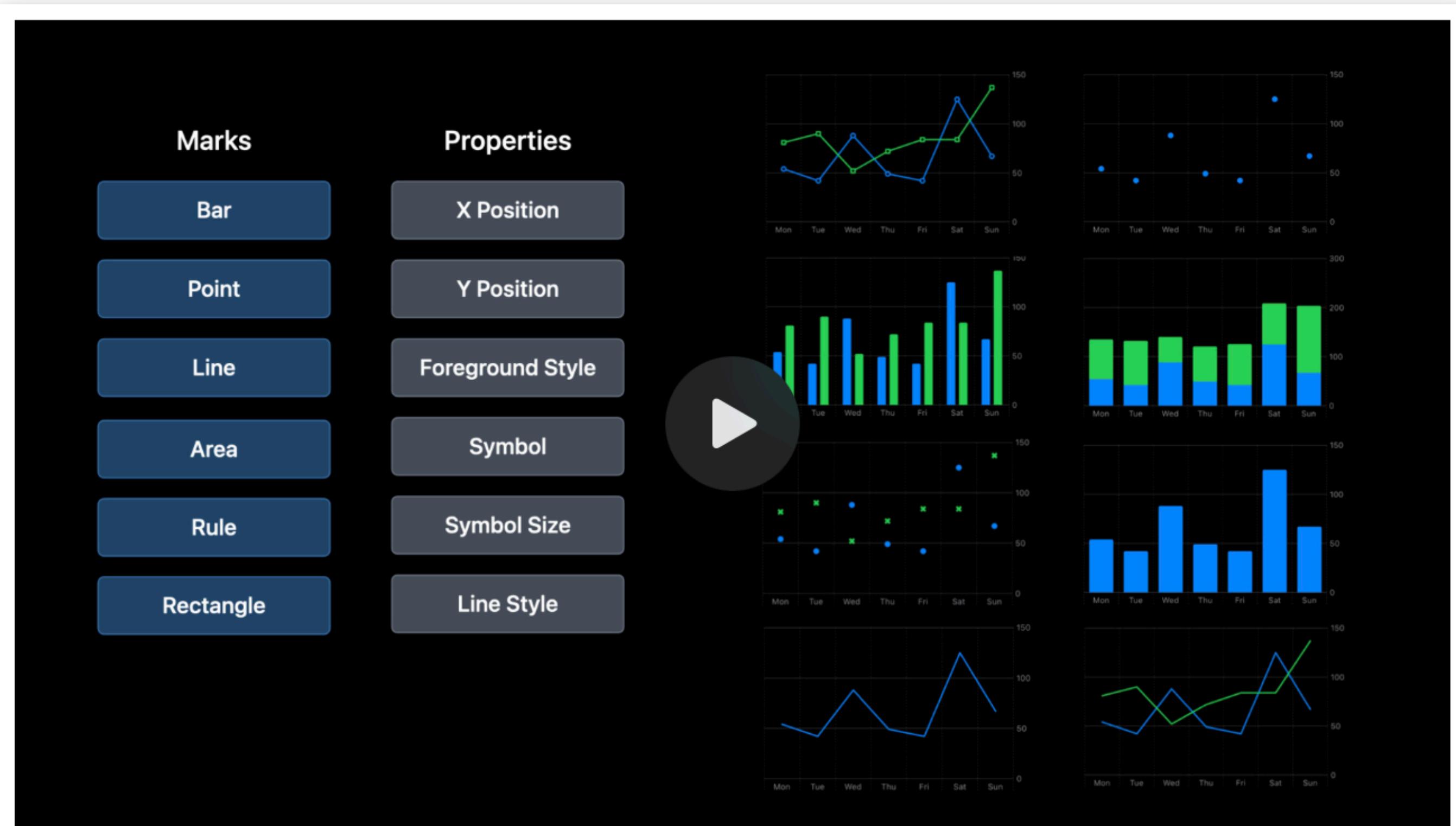
Weekly Report

15,000
10,000
5,000
0

Sun Mon Tue Wed Thu Fri Sat

10,531 6,019 7,200 8,311 7,403 6,503 9,230

● Sun ● Mon ● Tue ● Wed ● Thu ● Fri ● Sat



개요

전문

Swift Charts 소개

WWDC22

17분



모든 Apple 플랫폼에 최적화된 차트를 SwiftUI 하나만으로 제작할 수 있는 유연한 프레임워크인 Swift Charts를 소개합니다. 구성 구문을 사용하여 더 적은 코드로 유익하고 보기 좋고 사용이 손쉬운 차트를 만드는 방법을 알아볼 수 있습니다. Swift Charts를 통해 시각화를 만들기 위한 구성 요소를 소개하고, 간단한 제어자로 차트의 디자인을 변경하는 방법을 살펴보겠습니다. 또한 멋진 경험을 구현하는 데 도움을 주는 Xcode Previews(미리보기)의 최신 업데이트를 안내합니다.



감사합니다