

WHAT IS SWIFTUI



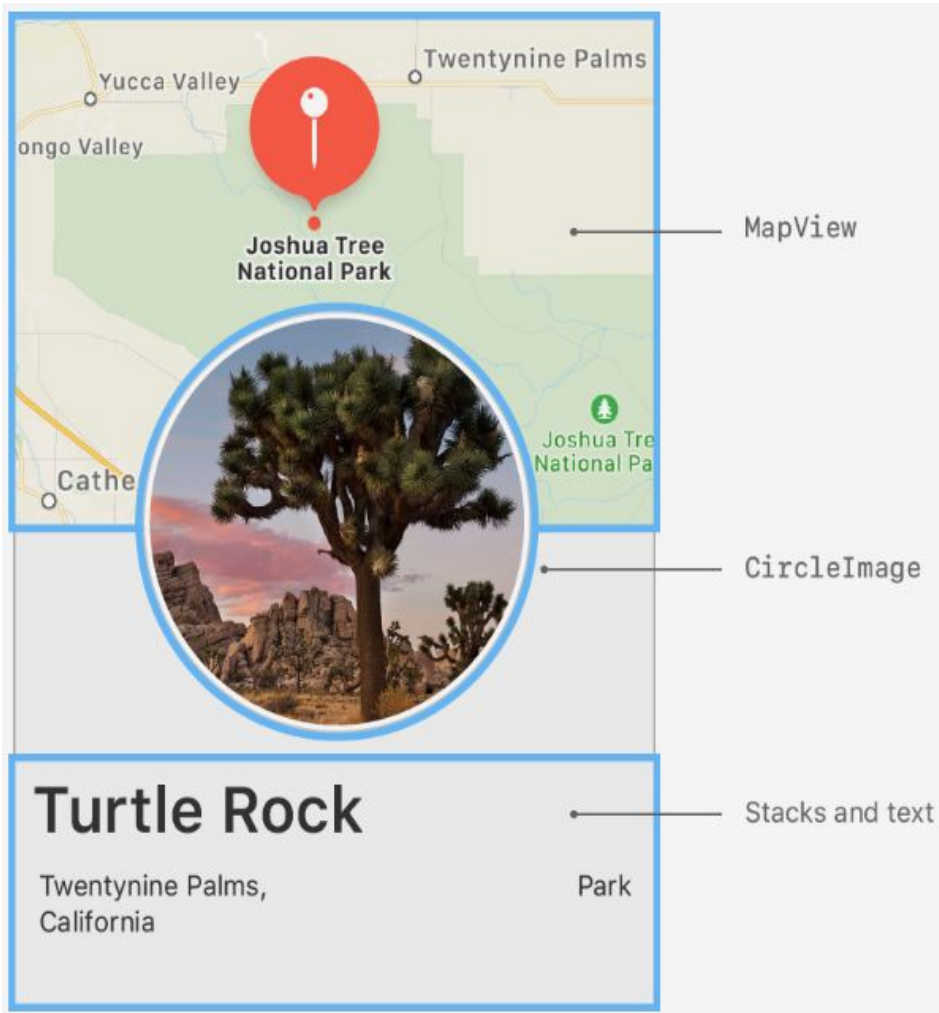
SwiftUI

May 21, 2020

Rediet Negash

SWIFT UI

Swift UI is an innovative and streamlined way to build a user interface on any Apple platform based on Swift's performance.





+



SwiftUI

With an easy-to-read and easy-to-write declarative Swift syntax, SwiftUI seamlessly integrates with the new Xcode design tools, allowing code and design to be fully synchronized.

Travel | Preview Travel: Succeeded | Today at 9:41 AM

Travel > iPhone Xs

Travel > Travel > Discover > DiscoverView.swift > body

```

13
14     var body: some View {
15         let pagingScrollViewController =
16             sceneController.pagingScrollViewController
17         pagingScrollViewController.didChangeToPageHandler = {
18             page in
19             self.selection = DataSource.shared.regions[page]
20         }
21
22         return GeometryReader { container in
23             return ZStack(alignment: .bottom) {
24                 GlobeView(
25                     selection: self.$selection.binding,
26                     sceneController: self.sceneController
27                 )
28                 PagingTilesView(
29                     containerSize: container.size,
30                     pagingScrollViewController:
31                         pagingScrollViewController
32                 ) { region in
33                     self.selection = region
34                 }
35             }
36         }
37     }
38
39     struct PagingTilesView<T> : View where T :
40         PagingScrollViewController {
41         let containerSize: CGSize
42         let pagingScrollViewController: T
43         var selectedTileAction: (Region) -> ()
44
45         var body: some View {
46             let tileWidth = containerSize.width * 0.9
47             let tileHeight = CGFloat(240.0)
48             let verticalTileSpacing = CGFloat(8.0)
49
50             return PagingScrollView(scrollViewController:

```

Preview



View Protocol—
describes the view's
content and layout

ContentView.swift

```
1  import SwiftUI
2
3  struct ContentView: View {
4      var body: some View {
5          Text("Hello World")
6      }
7  }
8
9  struct ContentView_Previews: PreviewProvider {
10     static var previews: some View {
11         ContentView()
12     }
13 }
```

Preview

Hello World

Declares a Preview for
the content and layout
described in the View
protocol

SWIFT UI- EXPLORING CANVAS

SOFTWARE ENGINEERING BENEFITS

Declarative syntax

In Swift UI, you'll write shorter code that's declarative, which makes it all about telling what the UI should do. It's readable and straight to the point

```
.frame(width: 100, height: 100)
.background(Color.blue)
.shadow(radius: 10)
.offset(x: 10, y:10)
.blur()
.padding()
```

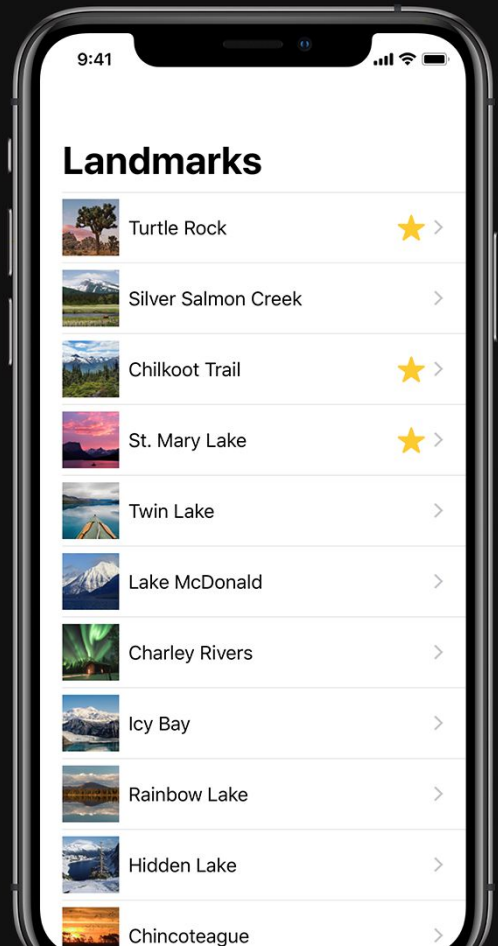
Example

Describe your layout just once.

Declare the content and layout for any state of your view. SwiftUI knows when that state changes, and updates your view's rendering to match.

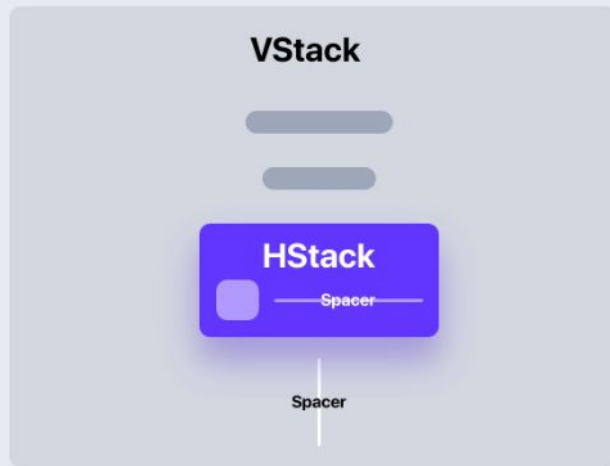
```
List(landmarks) { landmark in
    HStack {
        Image(landmark.thumbnail)
        Text(landmark.name)
        Spacer()

        if landmark.isFavorite {
            Image(systemName: "star.fill")
                .foregroundColor(.yellow)
        }
    }
}
```

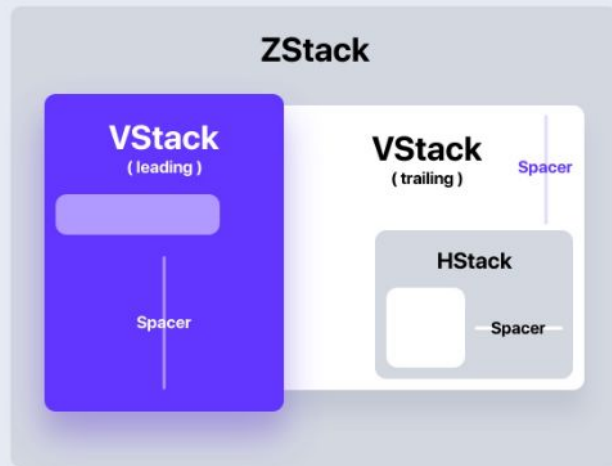


```
VStack {  
    Text("Card Back")  
}  
  
.frame(width: 340, height: 220.0)
```

SIMPLE LAYOUT



COMPLEX LAYOUT

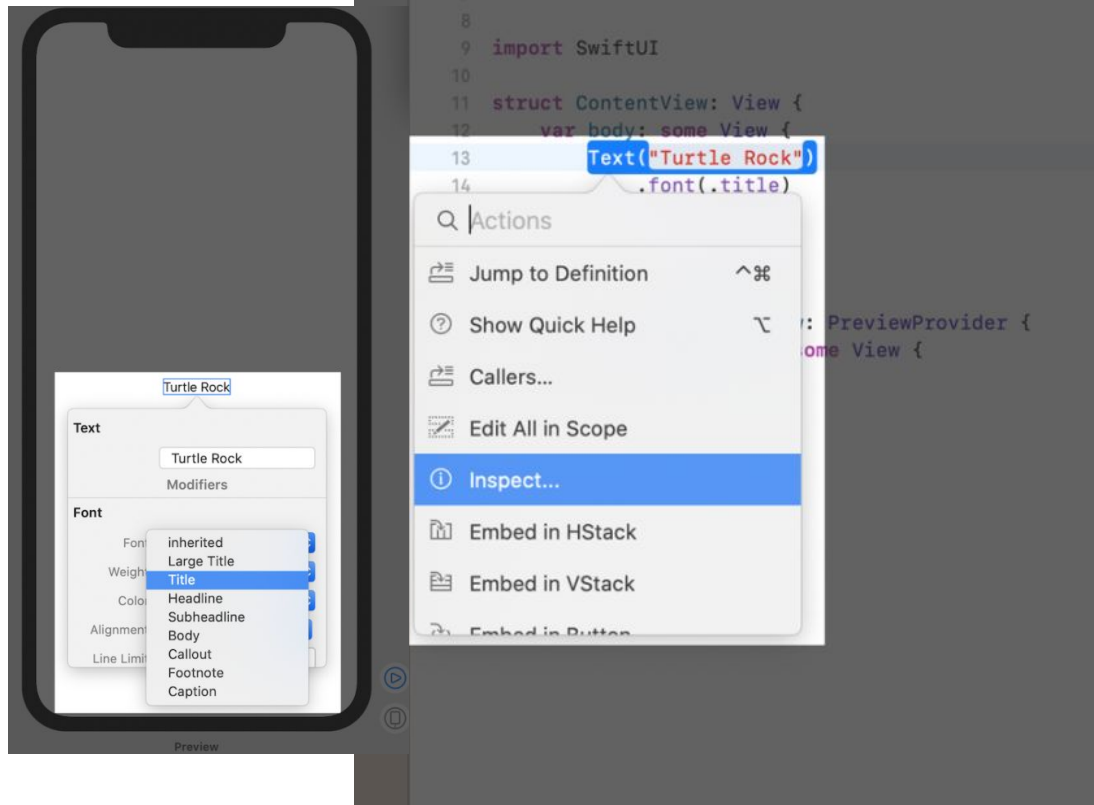


UI Components

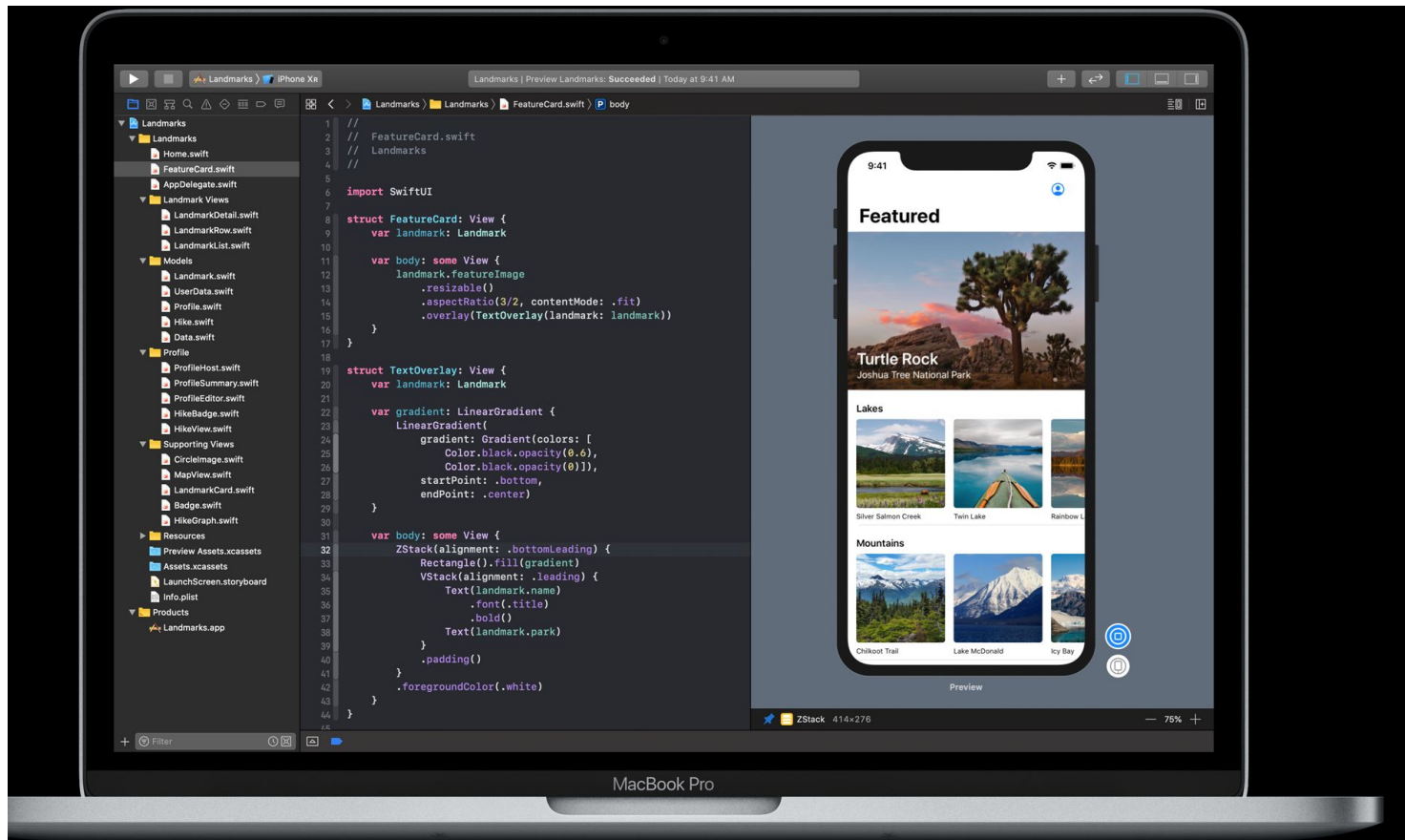
SOFTWARE ENGINEERING BENEFITS

Design tools

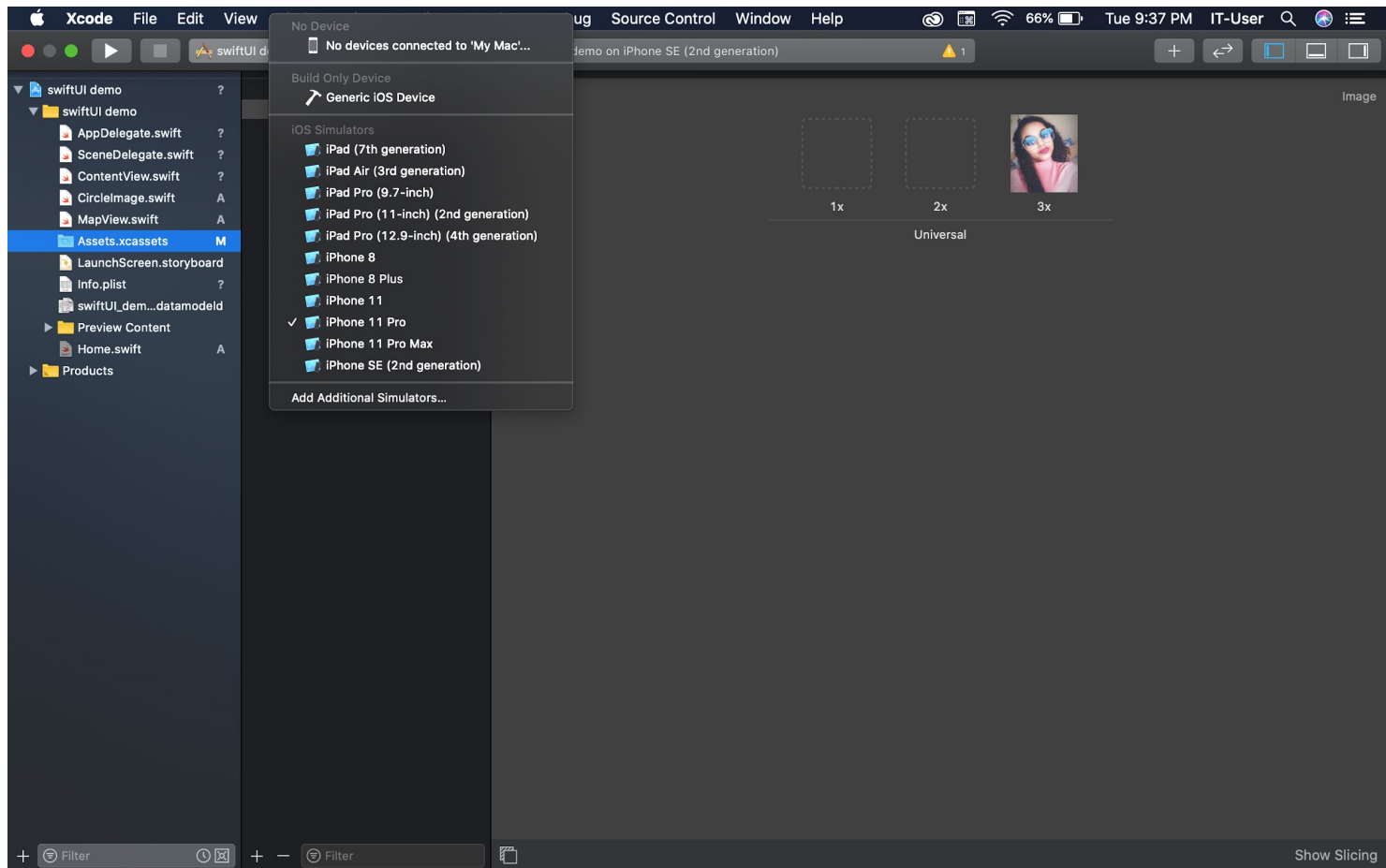
- Drag and drop
- Preview pop-up properties to allow quick changes to UI elements
- Views, Modifiers and Images can be dragged easily to preview



Live mode. Instantly see design changes with one or multiple precision previews. When you put Design Canvas into live mode, you can instantly interact with apps running on Xcode or connected devices.

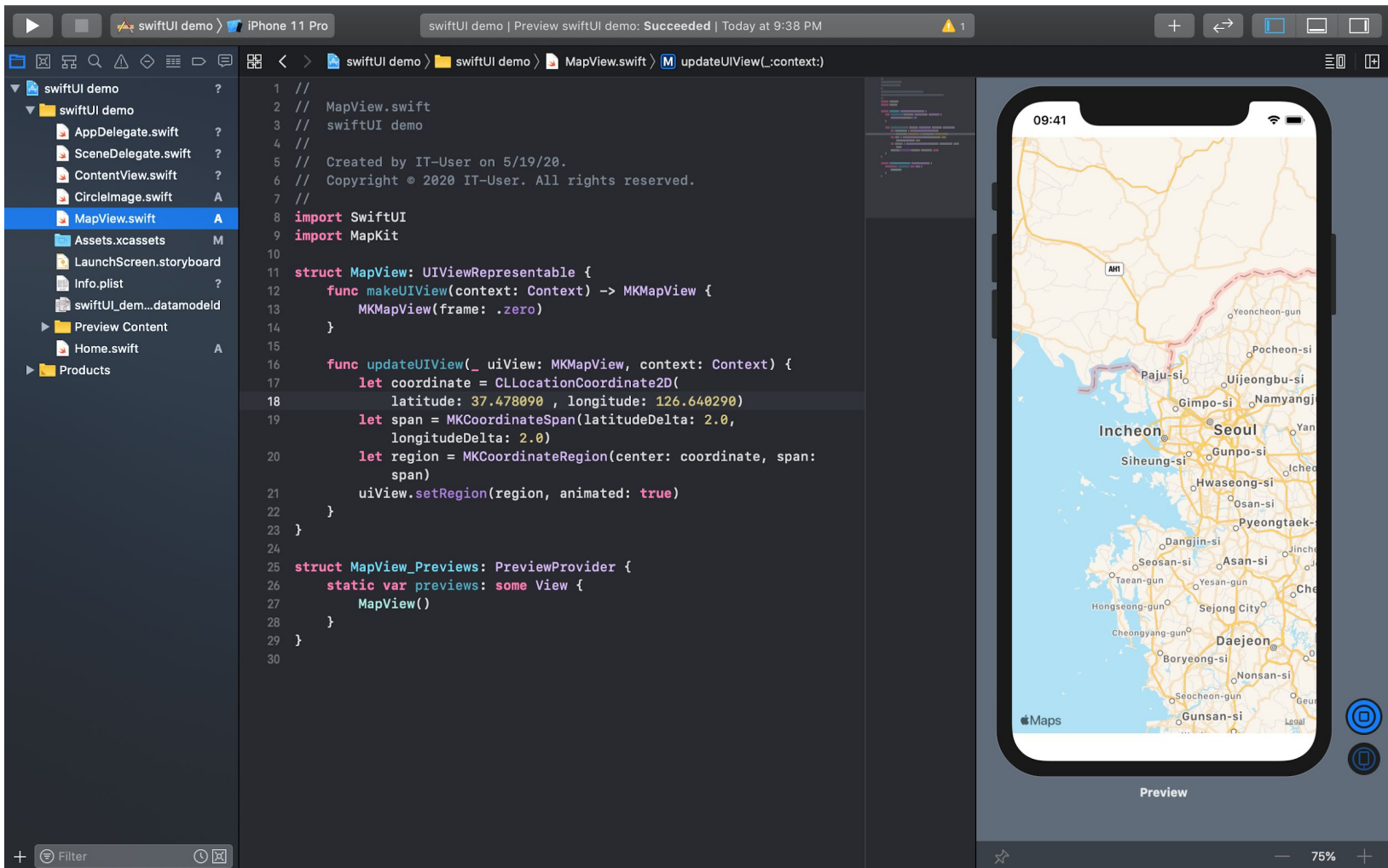


Native support on all Apple platforms



```
1 //
2 // CircleImage.swift
3 // swiftUI demo
4 //
5 // Created by IT-User on 5/19/20.
6 // Copyright © 2020 IT-User. All rights reserved.
7 //
8
9 import SwiftUI
10
11 struct CircleImage: View {
12     var body: some View {
13         Image("turtlerock")
14             .clipShape(Circle())
15             .overlay( Circle().stroke(Color.white, lineWidth: 2))
16             .shadow(radius: 1)
17
18
19     }
20 }
21
22 struct CircleImage_Previews: PreviewProvider {
23     static var previews: some View {
24         CircleImage()
25     }
26 }
27
```





swiftUI demo | Preview swiftUI demo: Succeeded | Today at 9:38 PM

swiftUI demo | swiftUI demo | ContentView.swift | body

swiftUI demo

AppDelegate.swift

SceneDelegate.swift

ContentView.swift

CircleImage.swift

MapView.swift

Assets.xcassets

LaunchScreen.storyboard

Info.plist

swiftUI_demo...datamodeld

Preview Content


Home.swift

Products

```
7 //
8
9 import SwiftUI
10
11 struct ContentView: View {
12     var body: some View {
13         VStack {
14             MapView()
15                 .edgesIgnoringSafeArea(.top)
16                 .frame(height: 300)
17
18             CircleImage()
19                 .offset(y: -130)
20                 .padding(.bottom, -130)
21
22             VStack(alignment: .leading) {
23                 Text("South Korea")
24                     .font(.title)
25                 HStack(alignment: .top) {
26                     Text("Incheon global campus")
27                         .font(.subheadline)
28                     Spacer()
29                     Text("Redi")
30                         .font(.subheadline)
31                 }
32             }
33             .padding()
34
35             Spacer()
36         }
37     }
38 }
39
40
41 struct ContentView_Previews: PreviewProvider {
42     static var previews: some View {
43         ContentView()
44     }
45 }
46
```

09:41

Map of South Korea showing Incheon and Seoul.

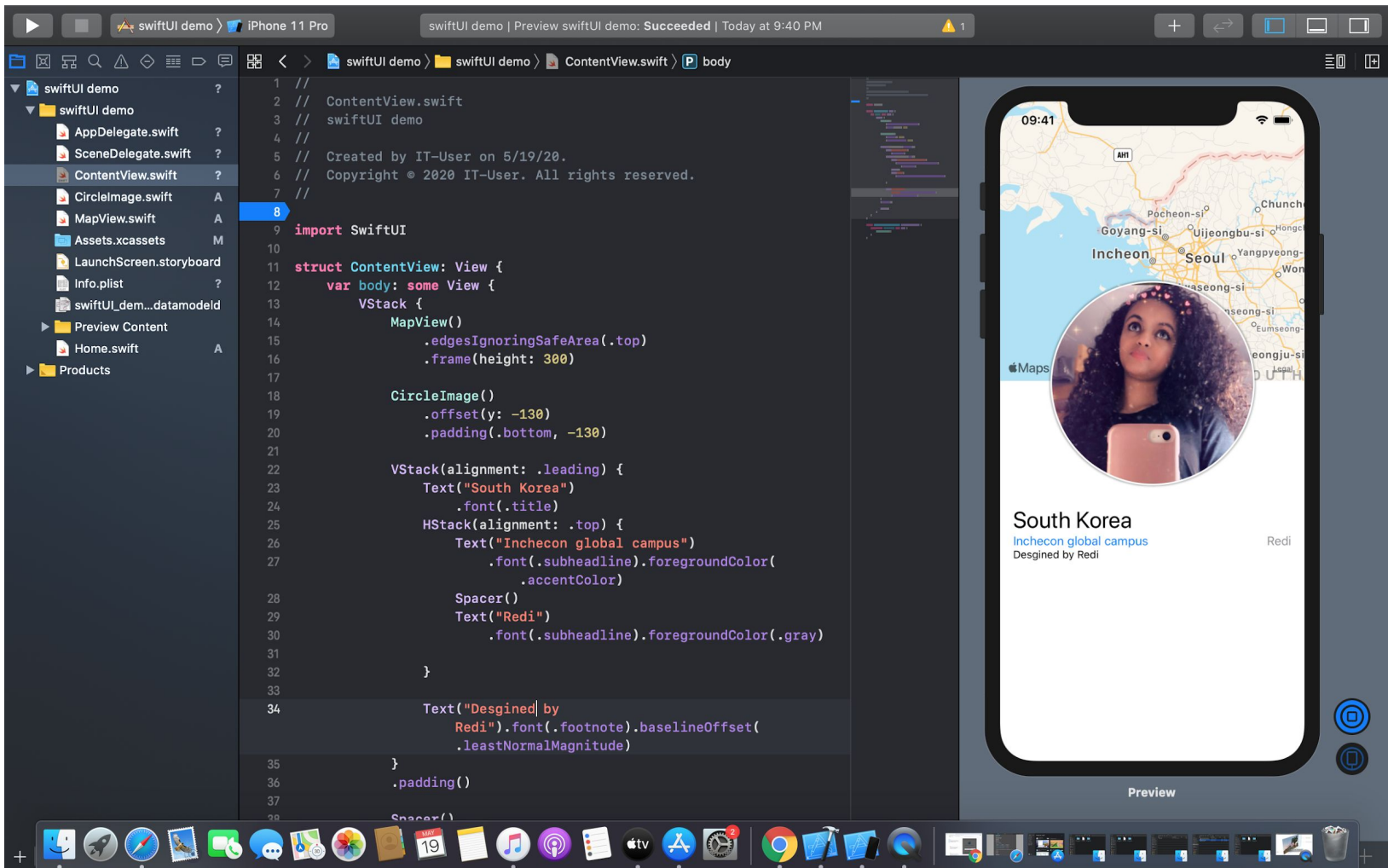


South Korea

Incheon global campus

Redi

Preview



RESOURCES AND LINKS

<https://developer.apple.com/tutorials/swiftui/>

<https://www.hackingwithswift.com/articles/196/learn-swiftui-with-free-tutorials>

<https://designcode.io/swiftui-course>

*The philosophy for Swift UI is not to
write once, apply everywhere, but to **learn once,**
apply anywhere.*

THANK YOU!