

I have the honor to present
to you Project Twelve



Discover SwiftUI

Hicham Bouzyarsit

What is SwiftUI ?



*Apple present
WWDC 2019*

Nouveauté

Framework

What is APPLE ?



1,5 milliard d'appareil actif

L'App Store dynamique et sûre

500 millions de visite

31e édition

Nouveau format

*23 millions d'inscrits
155 pays et régions*

Évolution ou Révolution

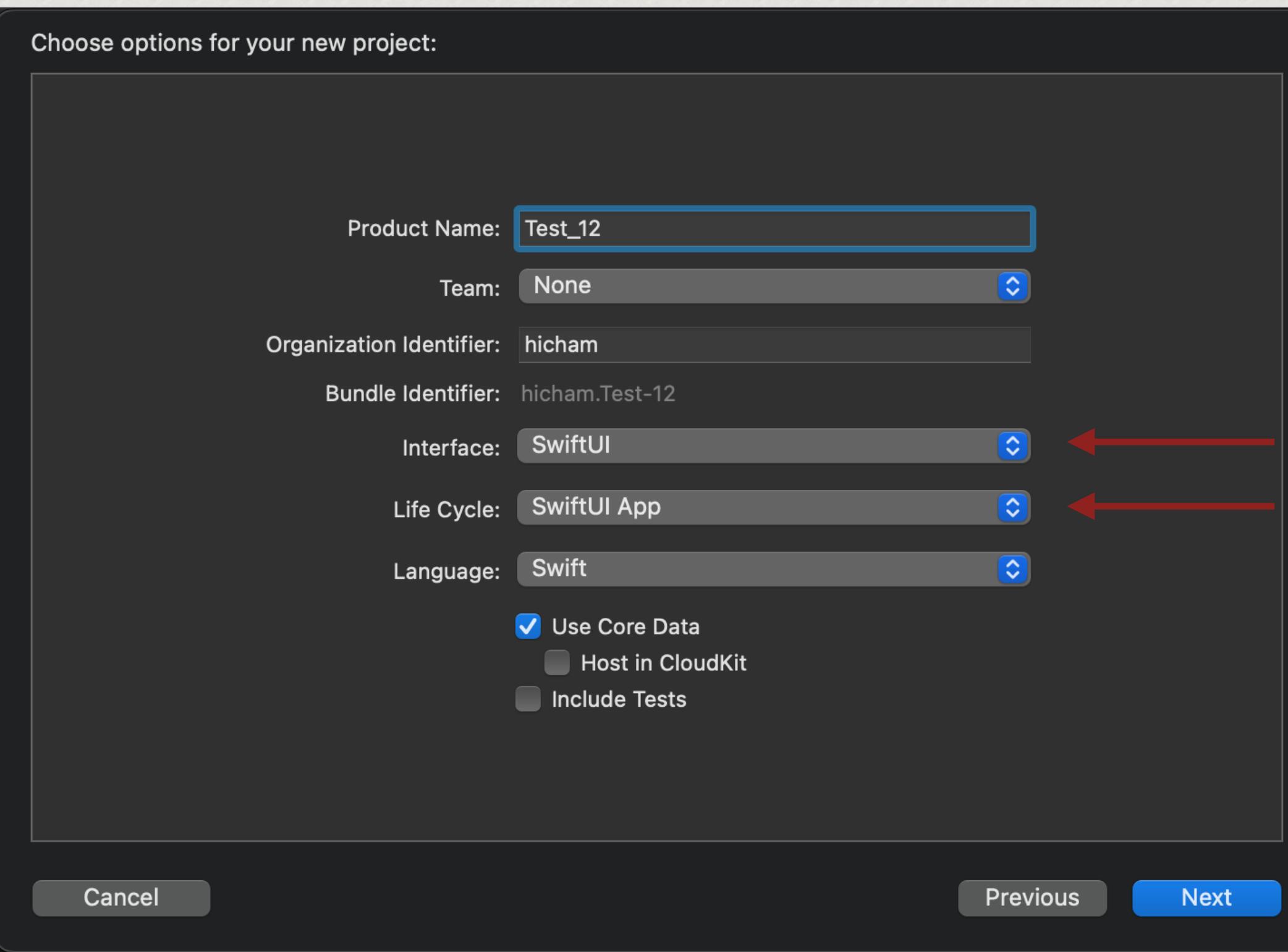


Avantages

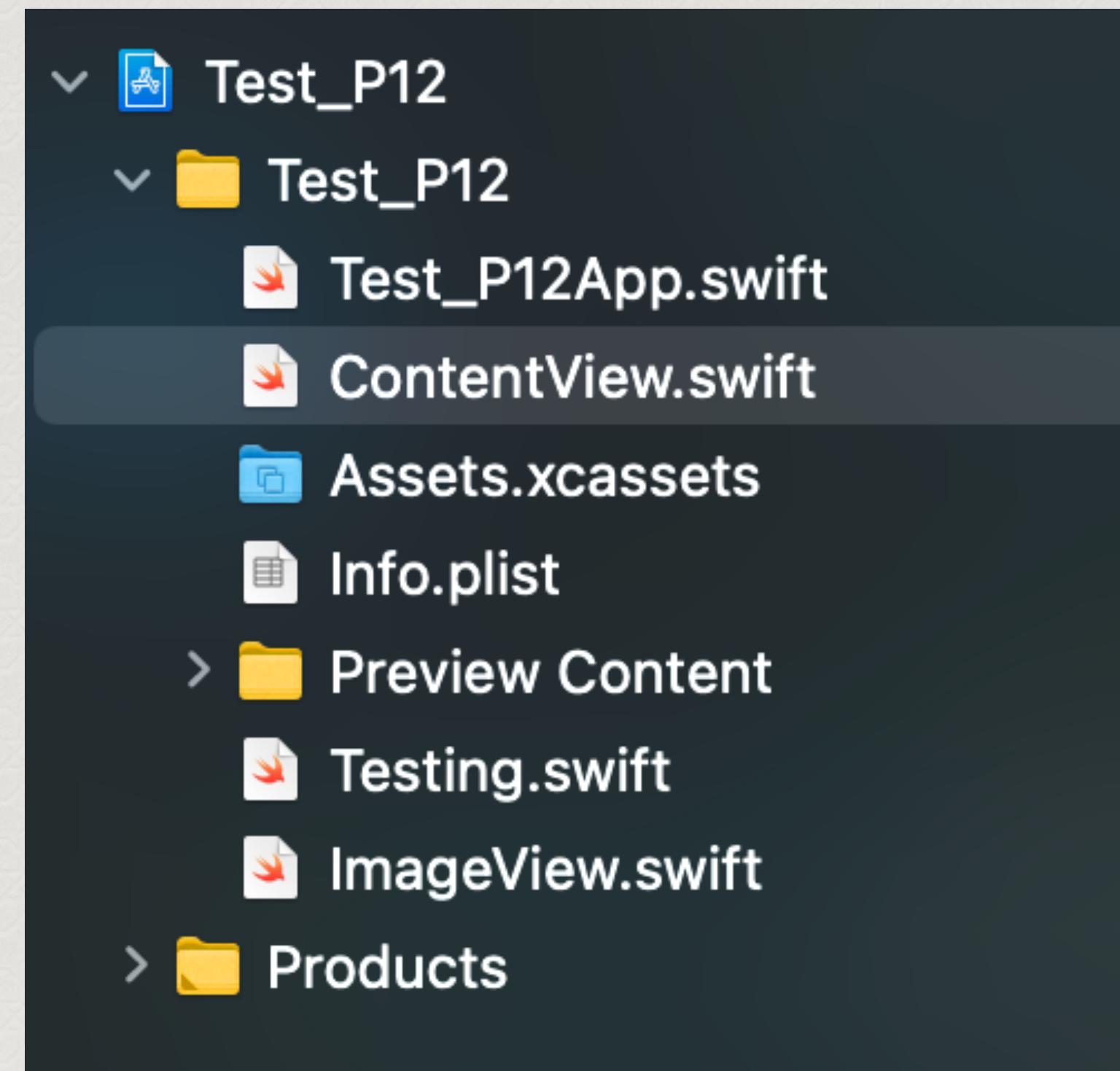
Inconvénients

Présentation d'Xcode

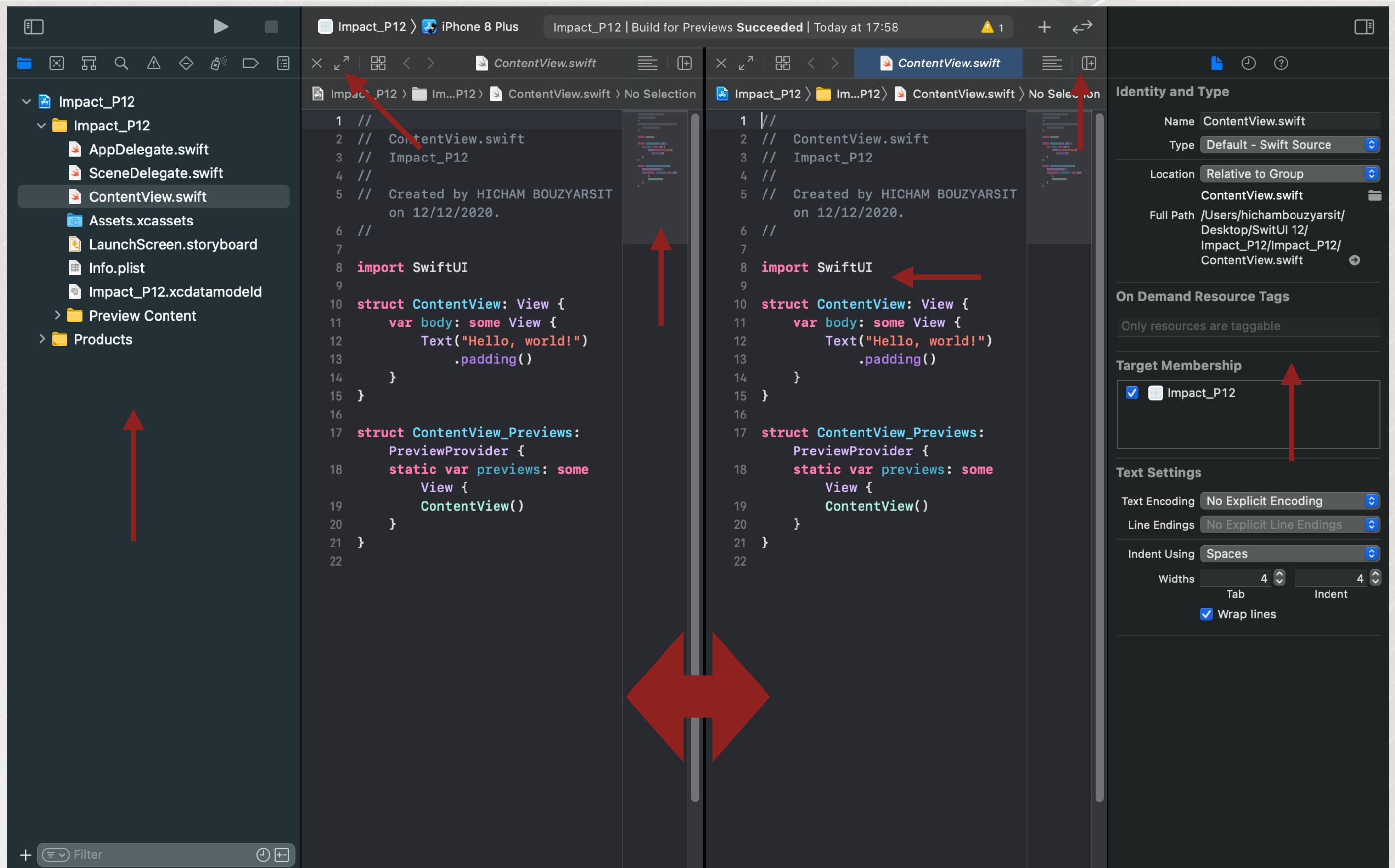
Création d'un projet



Nouveau fichier



Point important Xcode



Navigateur

Panneau principale

Les utilitaires

Ajout fichier

Minimap

SwiftUI

Editeur de texte

The screenshot shows two open files in Xcode:

- Test_P12App.swift**: This file contains the main application structure. It includes imports for SwiftUI and the Test_P12App module. The `@main` attribute is applied to the `Test_P12App` struct, which defines a `Scene` with a `ContentView` as its body.
- ContentView.swift**: This file contains the definition of the `ContentView` view. It imports SwiftUI and defines a `struct ContentView: View`. Inside, there is a `var body: some View` property containing a `Text("Hello, world!")` and a `.padding()` modifier. A preview provider named `ContentView_Previews` is also defined.

Red arrows point from the `@main` annotation in `Test_P12App.swift` and the `body` property in `ContentView.swift` towards the corresponding text in the slide's title and subtitle.

```
1 //  
2 //  Test_P12App.swift  
3 //  Test_P12  
4 //  
5 //  Created by HICHAM BOUZYARSIT on 15/12/2020.  
6 //  
7  
8 import SwiftUI  
9  
10 @main  
11 struct Test_P12App: App {  
12     var body: some Scene {  
13         WindowGroup {  
14             ContentView()  
15         }  
16     }  
17 }  
18
```

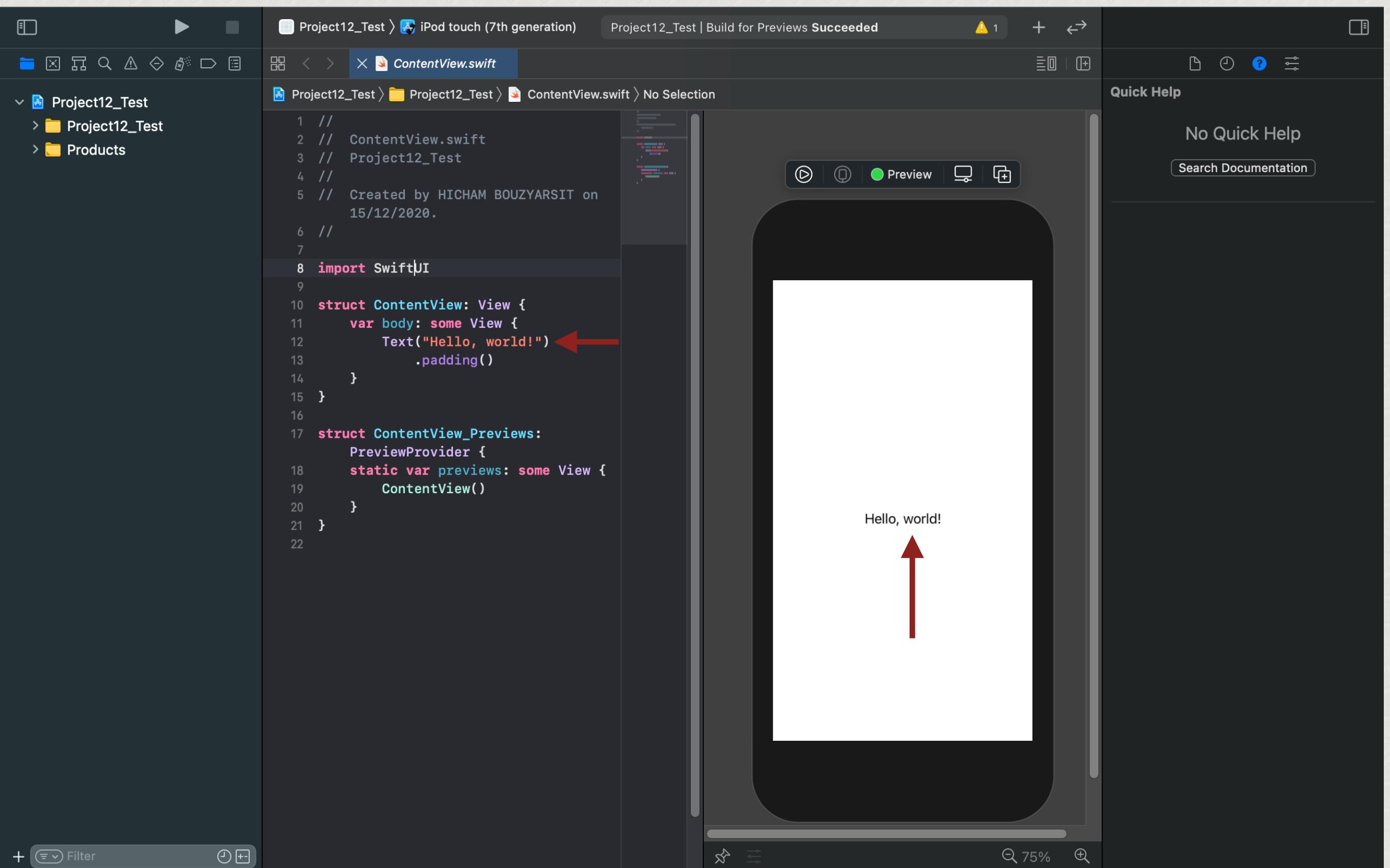
```
1 //  
2 //  ContentView.swift  
3 //  Test_P12  
4 //  
5 //  Created by HICHAM BOUZYARSIT on 15/12/2020.  
6 //  
7  
8 import SwiftUI  
9  
10 struct ContentView: View {  
11     var body: some View {  
12         Text("Hello, world!")  
13         .padding()  
14     }  
15 }  
16  
17 struct ContentView_Previews: PreviewProvider {  
18     static var previews: some View {  
19         ContentView()  
20     }  
21 }  
22
```

Nouveau fichier

@main

ContentView.swift

Le Canvas



The screenshot shows the Xcode interface with the following details:

- Project Structure:** Project12_Test > Project12_Test > ContentView.swift
- Code Editor:** ContentView.swift file content:

```
1 //  
2 // Content View  
3 // Project12_Test  
4 //  
5 // Created by HICHAM BOUZYARSIT on  
15/12/2020.  
6 //  
7  
8 import SwiftUI  
9  
10 struct ContentView: View {  
11     var body: some View {  
12         Text("Hello, world!")  
13             .padding()  
14     }  
15 }  
16  
17 struct ContentView_Previews:  
18     PreviewProvider {  
19         static var previews: some View {  
20             ContentView()  
21         }  
22     }
```
- Preview Area:** Shows a mobile device preview with the text "Hello, world!" displayed. A red arrow points from the code editor to the preview area.

Nouveau

Instantané

Rapide

Le Live Preview

A l'appui du bouton



@State, @Binding , \$

@Binding

```
struct Subview : View {  
    var viewColor: Color  
    @Binding var selectedColor: Color  
    var body: some View {  
        Button(action: {  
            self.selectedColor = viewColor  
        }, label: {  
            viewColor.frame(width: 50, height: 50, alignment: .center).clipShape(Circle())  
        })  
    }  
}
```

@State

```
@State var textColor = Color.black  
var body: some View {  
    VStack {  
        Text("Changement de couleur avec le @Binding")  
            .font(.largeTitle)  
            .foregroundColor(textColor)  
            .multilineTextAlignment(.center)  
    }  
}
```

\$

```
Subview(viewColor: .blue, selectedColor: $textColor)  
Subview(viewColor: .red, selectedColor: $textColor)
```

Les éléments qui composent notre code

VStack

```
struct Testing: View {
    var body: some View {

        VStack {
            Text("Hello World")
            Text("This is inside a
                stack")
        }
    }
}
```



Hello World
This is inside a stack

HStack

```
import SwiftUI

struct Testing: View {
    var body: some View {

        HStack(spacing: 20) {
            Text("Hello World")
            Text("I am student iOS")
        }
    }
}
```



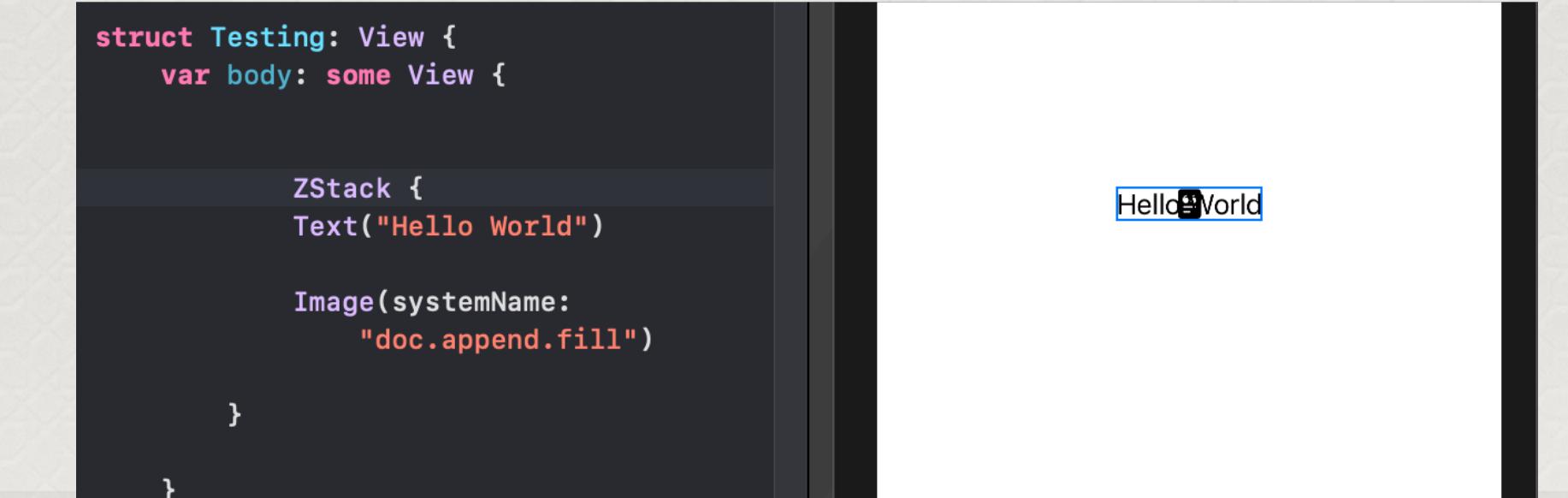
Hello World I am student iOS

ZStack

```
struct Testing: View {
    var body: some View {

        ZStack {
            Text("Hello World")

            Image(systemName:
                "doc.append.fill")
        }
    }
}
```



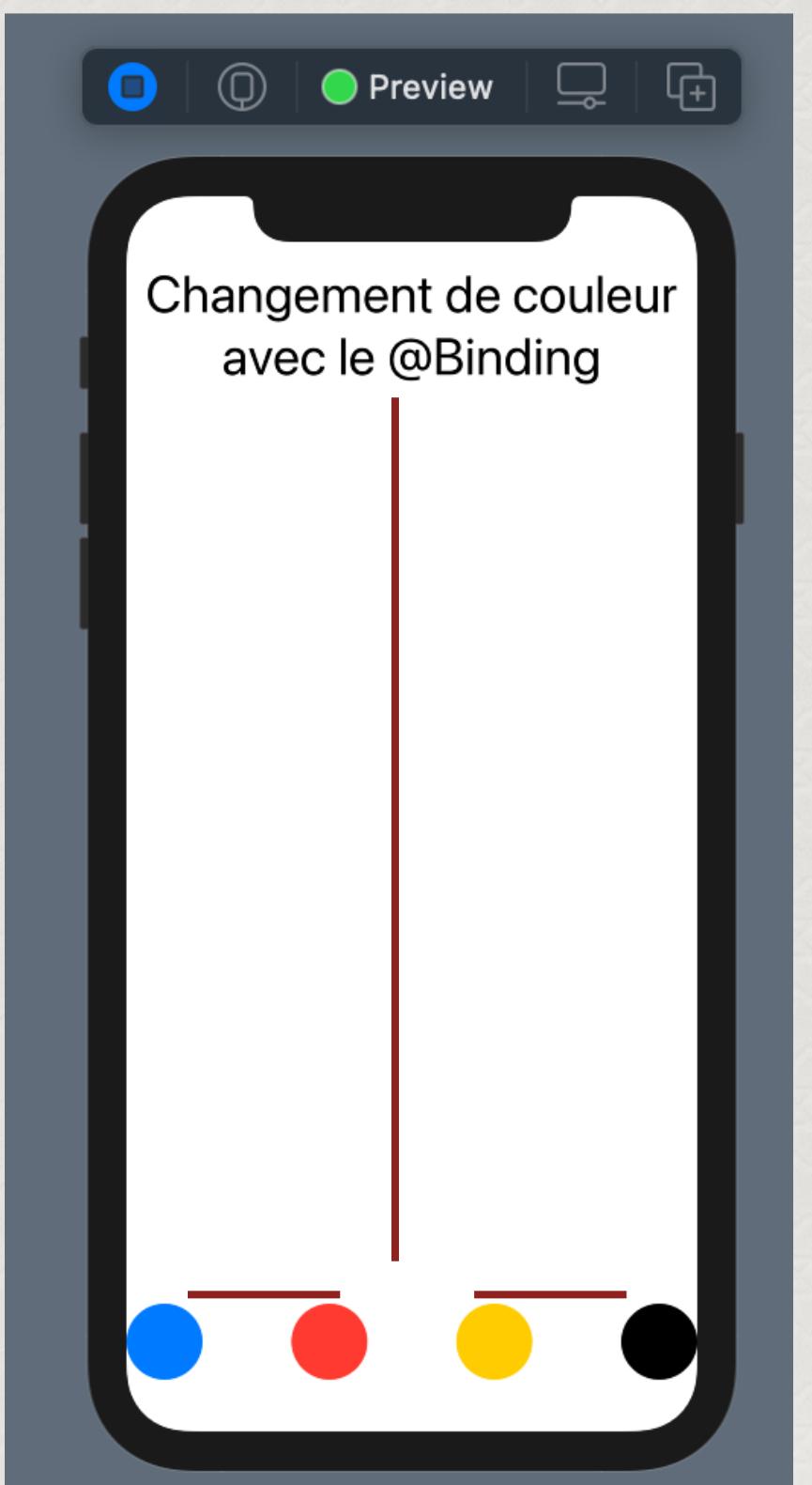
HelloWorld

Spacer()

Code

```
struct Testing: View {
    @State var textColor = Color.black
    var body: some View {
        VStack {
            Text("Changement de couleur avec le @Binding")
                .font(.largeTitle)
                .foregroundColor(textColor)
                .multilineTextAlignment(.center)
            Spacer()
            HStack {
                Subview(viewColor: .blue, selectedColor: $textColor)
                Spacer()
                Subview(viewColor: .red, selectedColor: $textColor)
                Spacer()
                Subview(viewColor: .yellow, selectedColor: $textColor)
                Spacer()
                Subview(viewColor: .black, selectedColor: $textColor)
            }
        }
    }
}
```

Résultat



*Spacer() =
Espace entre
les éléments*

Divider()

Code

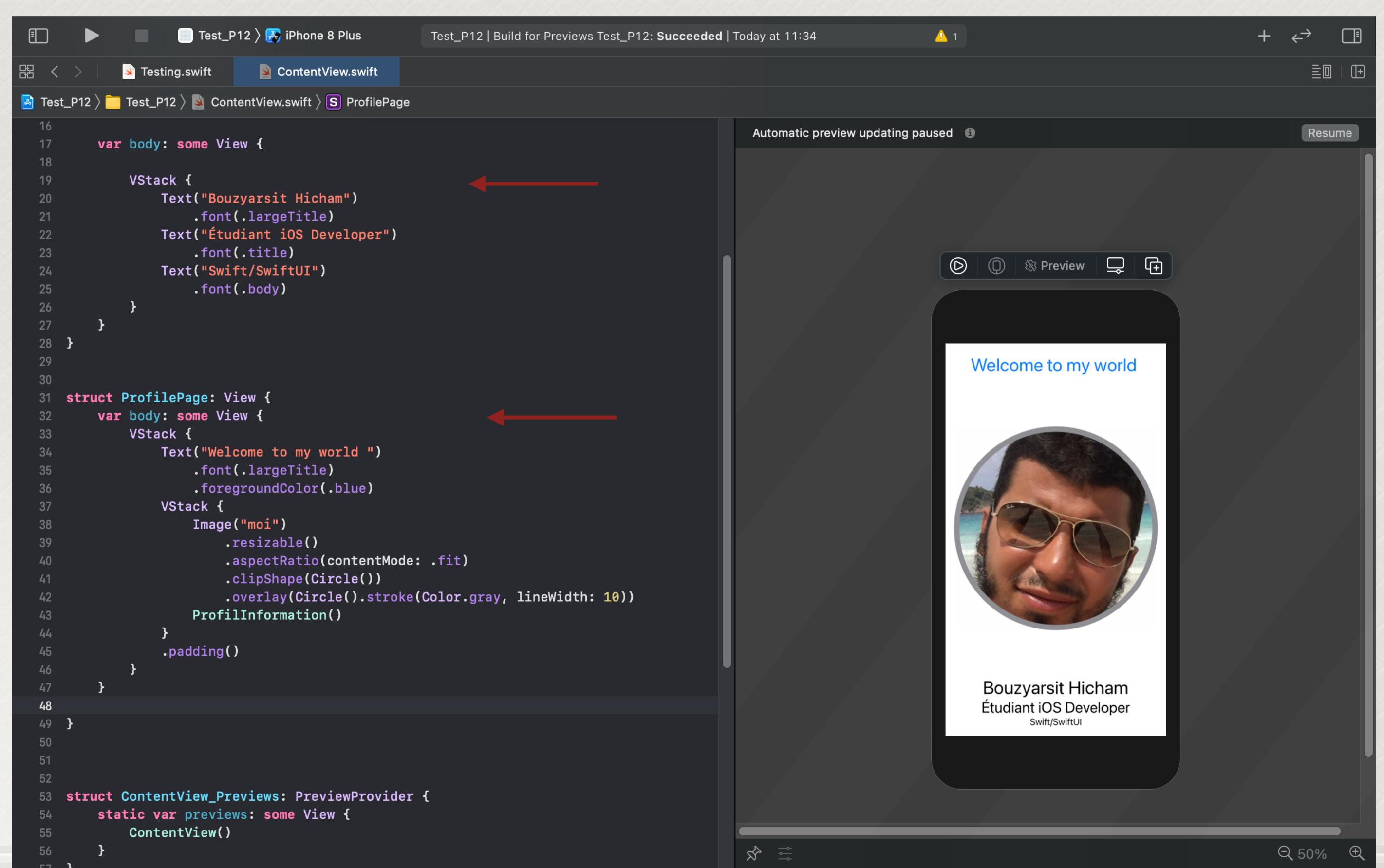
```
var body: some View {
    VStack {
        Text("Changement de couleur avec le @Binding")
            .font(.largeTitle)
            .foregroundColor(textColor)
            .multilineTextAlignment(.center)
        Divider()
        HStack {
            Subview(viewColor: .blue, selectedColor:
                $textColor)
```

Résultat



*Divider() =
Trait entre
les éléments*

Un composant



The screenshot shows the Xcode interface with two main panes. The left pane displays the code for `ContentView.swift`, and the right pane shows a preview of the SwiftUI view on an iPhone 8 Plus simulator.

```
16  var body: some View {  
17      VStack {  
18          Text("Bouzyarsit Hicham")  
19              .font(.largeTitle)  
20          Text("Étudiant iOS Developer")  
21              .font(.title)  
22          Text("Swift/SwiftUI")  
23              .font(.body)  
24      }  
25  }  
26 }  
27 }  
28 }  
29 }  
30 }  
31 struct ProfilePage: View {  
32     var body: some View {  
33         VStack {  
34             Text("Welcome to my world ")  
35                 .font(.largeTitle)  
36                 .foregroundColor(.blue)  
37             VStack {  
38                 Image("moi")  
39                     .resizable()  
40                     .aspectRatio(contentMode: .fit)  
41                     .clipShape(Circle())  
42                     .overlay(Circle().stroke(Color.gray, lineWidth: 10))  
43             ProfilInformation()  
44         }  
45     }  
46     .padding()  
47 }  
48 }  
49 }  
50 }  
51 }  
52 }  
53 struct ContentView_Previews: PreviewProvider {  
54     static var previews: some View {  
55         ContentView()  
56     }  
57 }
```

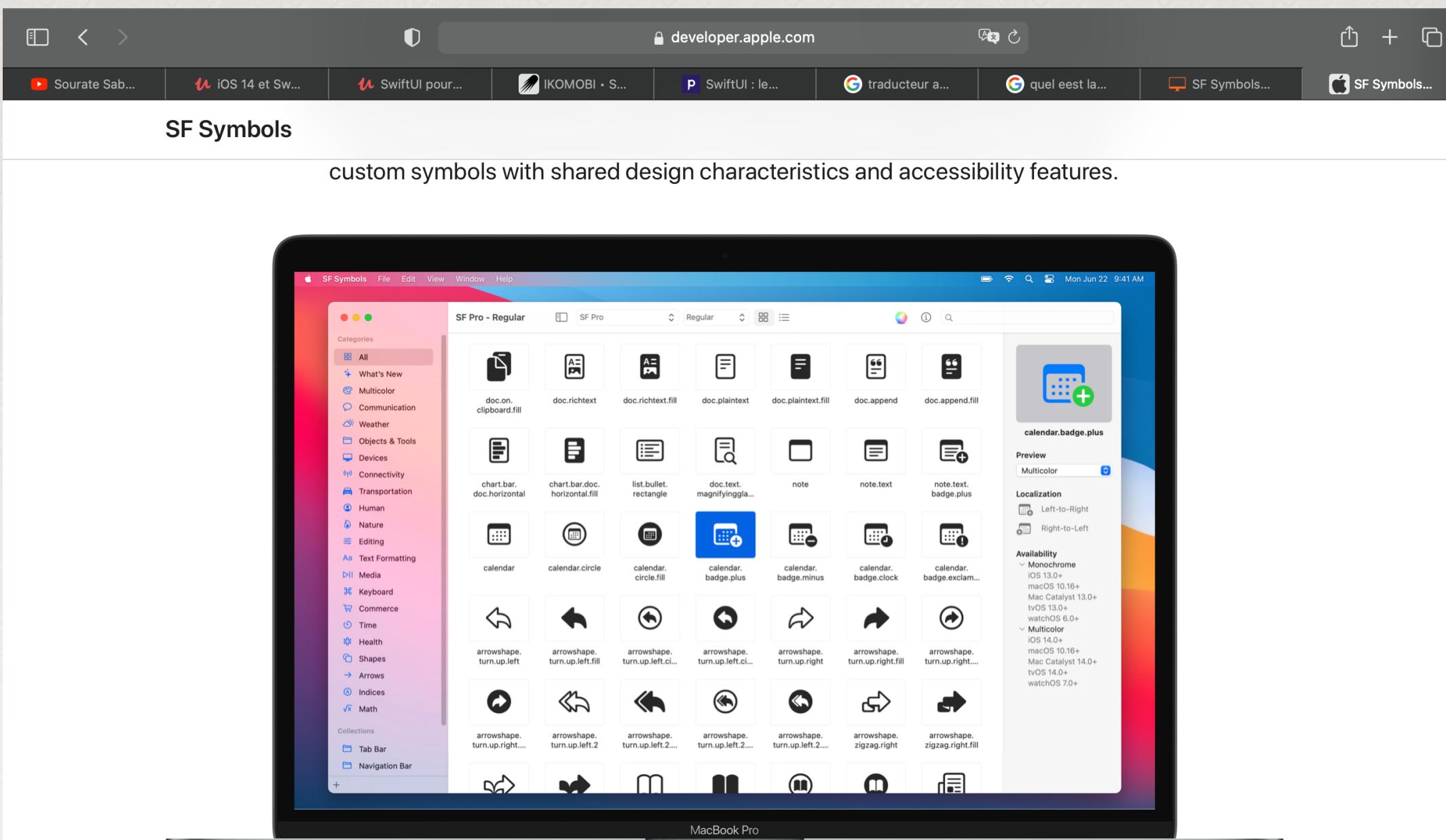
Two red arrows point from the code to specific parts of the preview window. One arrow points to the first `VStack` in the `ProfilePage` struct, and another arrow points to the second `VStack` in the same struct. The preview window shows a circular profile picture of a man wearing sunglasses and a welcome message.

Ce qui va composer ma vue

Les ressources

Lien de téléchargement :

<https://developer.apple.com/sf-symbols/>



Download the app



SF Symbols 2.1

Requires macOS 10.15.3 or later

[Download \(152.9 MB\)](#)



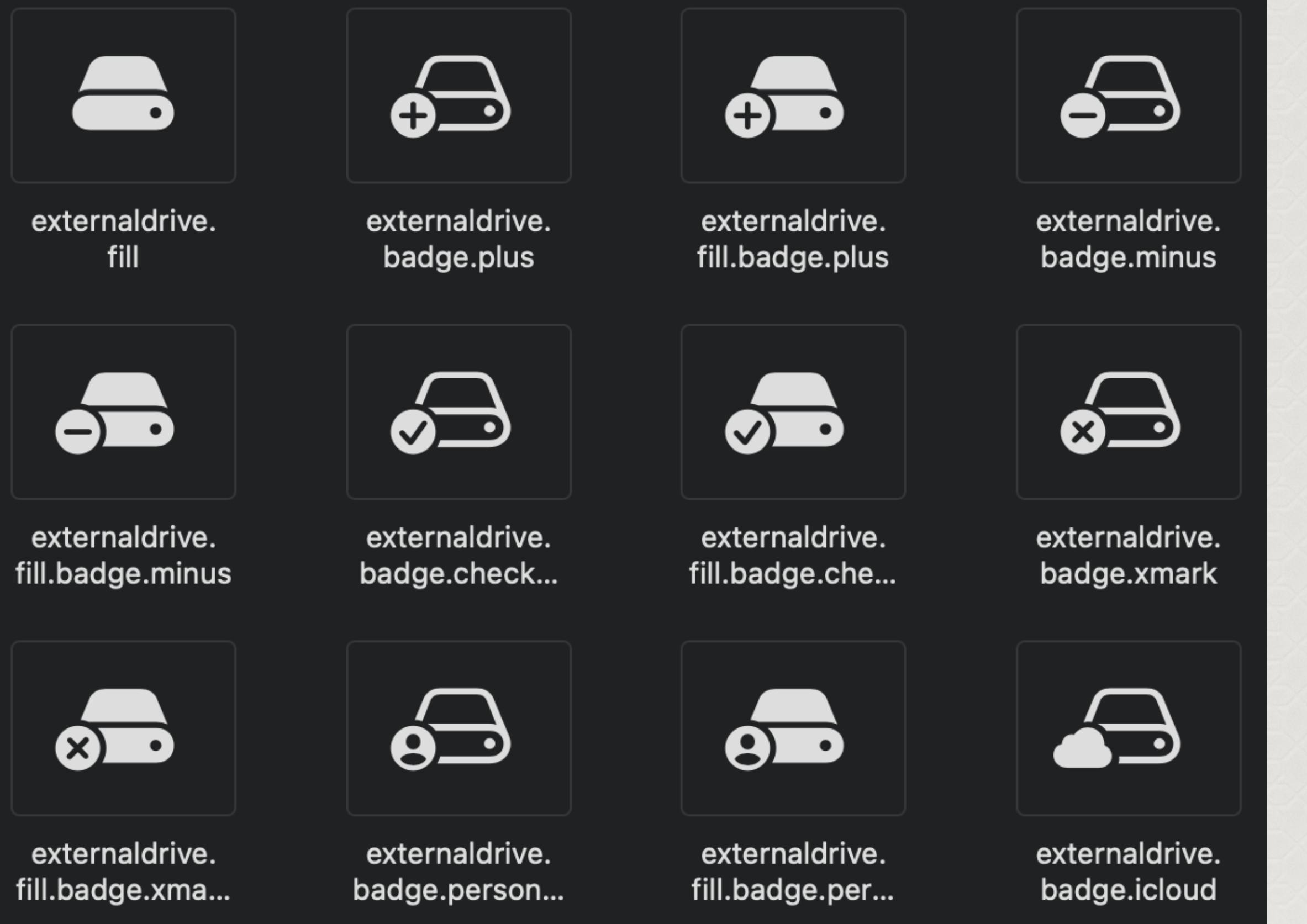
SF Symbols 1.1

Requires macOS 10.14.4 or later

[Download \(58 MB\)](#)

Application Installer

- Appareils
- Jeux
- Connectivité
- Transports
- Humain
- Nature
- Édition
- Formatage du texte
- Multimédia
- Clavier
- Commerce
- Temps
- Santé
- Formes
- Flèches



SF Symbols

2000 icônes

Gratuite

Exemple dans le code

The screenshot shows the Xcode interface with the following details:

- Project Navigator:** Shows files like Testing.swift, ContentView.swift, and ImageView.swift.
- Editor:** The ImageView.swift file is open, containing the following code:

```
1 //  
2 //  ImageView.swift  
3 //  Test_P12  
4 //  
5 //  Created by HICHAM BOUZYARSIT on 17/12/2020.  
6 //  
7 import SwiftUI  
8  
9 struct ImageView: View {  
10     var body: some View {  
11         Image(systemName: "arrowshape.turn.up.left")  
12             .padding(.trailing)  
13             .font(.largeTitle)  
14             .foregroundColor(.blue)  
15     }  
16 }  
17  
18 struct ImageView_Previews: PreviewProvider {  
19     static var previews: some View {  
20         ImageView()  
21     }  
22 }  
23 }
```

A red arrow points from the line `Image(systemName: "arrowshape.turn.up.left")` to the preview area.
- Preview Area:** Shows a smartphone-shaped preview window with a blue arrow pointing to the right.
- Attributes Inspector:** On the right, it shows the following settings for the image:
 - Image:** System Name: arrowshape.turn.up.left
 - Modifiers:** None
 - Padding:** Padding: Default (checkbox checked)
 - Frame:** Size: Inherit, Width: 58, Height: 35
 - Font:** Large Title
 - Foreground Color:** Blue
 - Add Modifier:** NoneA red arrow points from the `.padding(.trailing)` line in the code to the **Padding** section in the Attributes Inspector.

Image(systemName: "arrowshape.turn.up.left")

Preview

Padding()

Framework Combine

I/ Fourni ce qu'on l'appelle les éditeurs et les abonnées

Nouveau Framework

Présenté par
Apple

Programmation
asynchrone

Permet d'intégrer la notion
de programmation réactive au sein des apps.

Déclarative

Réactive

fonctionnelle

SwiftUI

Il s'agit de chaîner
des fonctions et de
passer des valeurs
de l'une à l'autre

+

Combine

=

Plus complet ,
plus puissant

@Published

2/ Editeurs = Observables et abonnées = les observateurs

SwiftUI utilise les wrappers de propriété :

Avec @Published

Sans @Published

@Published

@ObservedObject

Notifie que les valeurs ont changé

Aucune notification à la struct qui observe, donc aucun changement.

Fournis par Combine, pour créer implicitement un éditeur et prendre en charge ses mécanismes de vue déclarative.

@ObservableObject

3/ Un éditeur expose des valeurs qui permet de changer sur lesquelles,
Les abonnées s'abonne pour recevoir toute les mises à jours.

Données plus
complexe

→ Ces données seront des objets

Pour qu'une classe
soit observée,
Besoin d'une
SwiftUI qui observe
cet Objet.

→ @ObservedObject

Résultat

Pour garder le source OfThruth
Il devront être conforme,
au protocole Observableobject

Toutes les propriétés
@Published de la classe
@ObservableObject
seront mis à jour au besoin

La documentation



Discover

Design

Develop

Distribute

Support

Account



SwiftUI Tutorials

Introducing SwiftUI

SwiftUI is a modern way to declare user interfaces for any Apple platform. Create beautiful, dynamic apps faster than ever before.

⌚ 4hr 25min Estimated Time

Get started



Thank you