

swifEUI

créer un compte

Dans cet exemple, On va créer un composant (avec cmd + click) qui contient toute la HStack.

On va pouvoir s'en resservir tout simplement et cela nous évite d'avoir des vues avec beaucoup de lignes de code. C'est un petit peu comme faire une fonction, ça permet d'y voir plus clair et on peut réutiliser ce composant autant de fois que l'on veut.

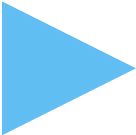
Il est conseillé d'enregistrer ces composants dans des nouveaux fichiers.

```

1 //
2 // ContentView.swift
3 // projet_12
4 //
5 // Created by Maxime Bontemps on 01/09/2019.
6 // Copyright © 2019 Maxime Bontemps. All rights reserved.
7 //
8
9 import SwiftUI
10
11 struct ContentView: View {
12     var body: some View {
13         VStack {
14             Text("Cocktails")
15                 .font(.title)
16
17             HStack {
18                 Image("gintonic")
19                     .resizable()
20                     .aspectRatio(contentMode: .fill)
21                     .frame(width: 50, height: 50)
22                 Text("Gin Tonic")
23
24                 Spacer()
25             }
26
27             Spacer()
28         }
29     }
30 }
31
32 struct ContentView_Previews: PreviewProvider {
33     static var previews: some View {
34         ContentView()
35     }
36 }
37
38
39

```





```

1 //
2 // ContentView.swift
3 // projet_12
4 //
5 // Created by Maxime Bontemps on 01/09/2019.
6 // Copyright © 2019 Maxime Bontemps. All rights reserved.
7 //
8
9 import SwiftUI
10
11 struct ContentView: View {
12     var body: some View {
13         VStack {
14             Text("Cocktails")
15                 .font(.title)
16
17             HStack {
18                 Image("gintonic")
19                     .resizable()
20                     .aspectRatio(contentMode: .fill)
21                     .frame(width: 50, height: 50)
22                 Text("Gin Tonic")
23
24                 Spacer()
25             }
26
27             Spacer()
28         }
29     }
30 }
31
32 struct ContentView_Previews: PreviewProvider {
33     static var previews: some View {
34         ContentView()
35     }
36 }
37
38
39

```



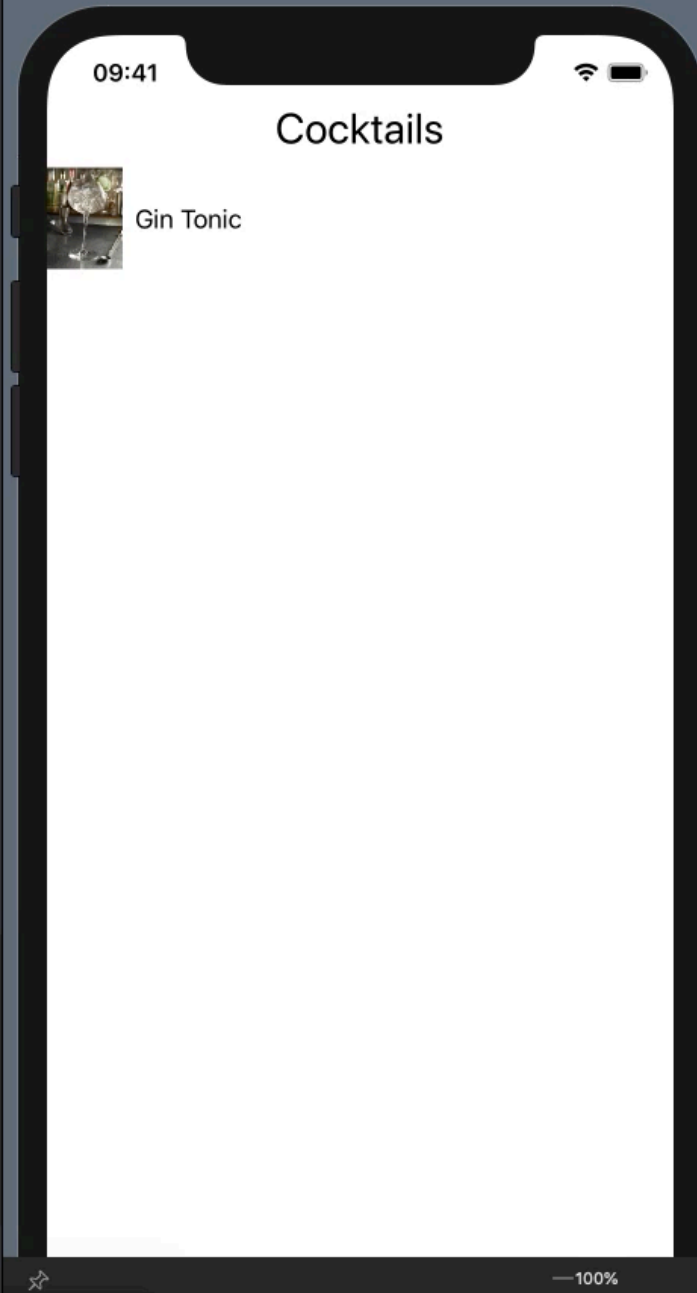

```

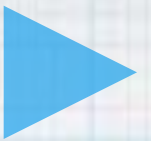
1 //
2 // ContentView.swift
3 // projet_12
4 //
5 // Created by Maxime Bontemps on 01/09/2019.
6 // Copyright © 2019 Maxime Bontemps. All rights reserved.
7 //
8
9 import SwiftUI
10
11 struct ContentView: View {
12     var body: some View {
13         VStack {
14             Text("Cocktails")
15                 .font(.title)
16
17             HStack {
18                 Image("gintonic")
19                     .resizable()
20                     .aspectRatio(contentMode: .fill)
21                     .frame(width:50, height:50)
22                 Text("Gin Tonic")
23
24                 Spacer()
25             }
26
27             Spacer()
28         }
29     }
30 }
31
32 struct ContentView_Previews: PreviewProvider {
33     static var previews: some View {
34         ContentView()
35     }
36 }
37
38
39 |

```

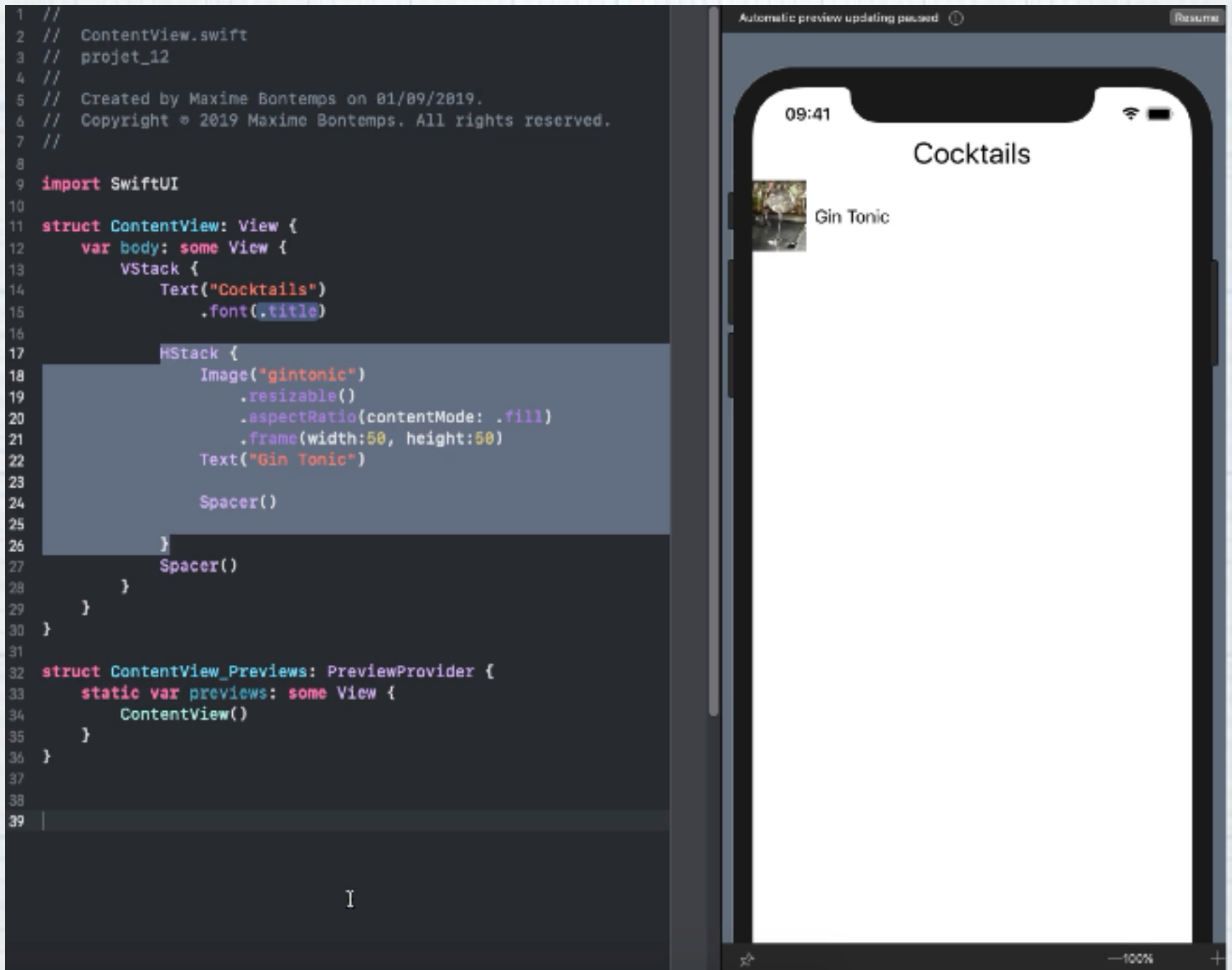
Automatic preview updating paused ⓘ

Resume





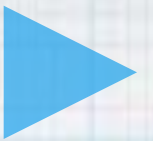
Créer un composant



Dans cet exemple, On va créer un composant (avec cmd + click) qui contient toute la HStack.

On va pouvoir s'en resservir tout simplement et cela nous évite d'avoir des vues avec beaucoup de lignes de code. C'est un petit peu comme faire une fonction, ça permet d'y voir plus clair et on peut réutiliser ce composant autant de fois que l'on veut.

Il est conseillé d'enregistrer ces composants dans des nouveaux fichiers.



Naviguer entre deux écrans

```
import SwiftUI

struct ContentView: View {
    var body: some View {
        NavigationView{
            VStack {
                Text("Cocktails")
                    .font(.title)
                NavigationLink(destination: CocktailDetails()){
                    cocktailsRow()
                }.buttonStyle(PlainButtonStyle())
                Spacer()
            }
        }
    }
}

struct ContentView_Previews: PreviewProvider {
    static var previews: some View {
        ContentView()
    }
}
```

Pour commencer j'ai créé une nouvelle vue nommé `CocktailDetails`. Ensuite j'ai créé un `NavigationLink` qui a pour destination cette vue et j'y ai mis la vue `cocktailRow` (si on appuie sur `cocktailRow` on va naviguer vers `CocktailsDetails`).

Enfin j'ai modifié sa propriété `.buttonStyle` afin d'avoir un affichage correct (par défaut tout ce qui se trouve dans un `navigationLink` est bleu)

