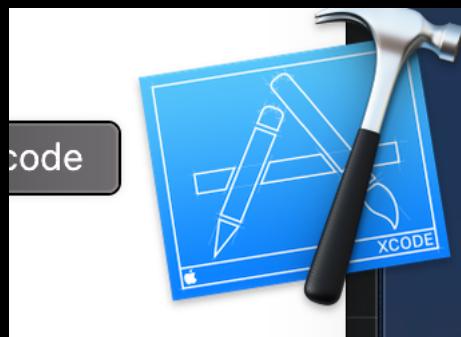


Swift UI 100 days challenge

Week 1 | Day 1 - 5 | Making Xcode Swift UI project, understanding project structure and playing with Text in SwiftUI

Let's get started

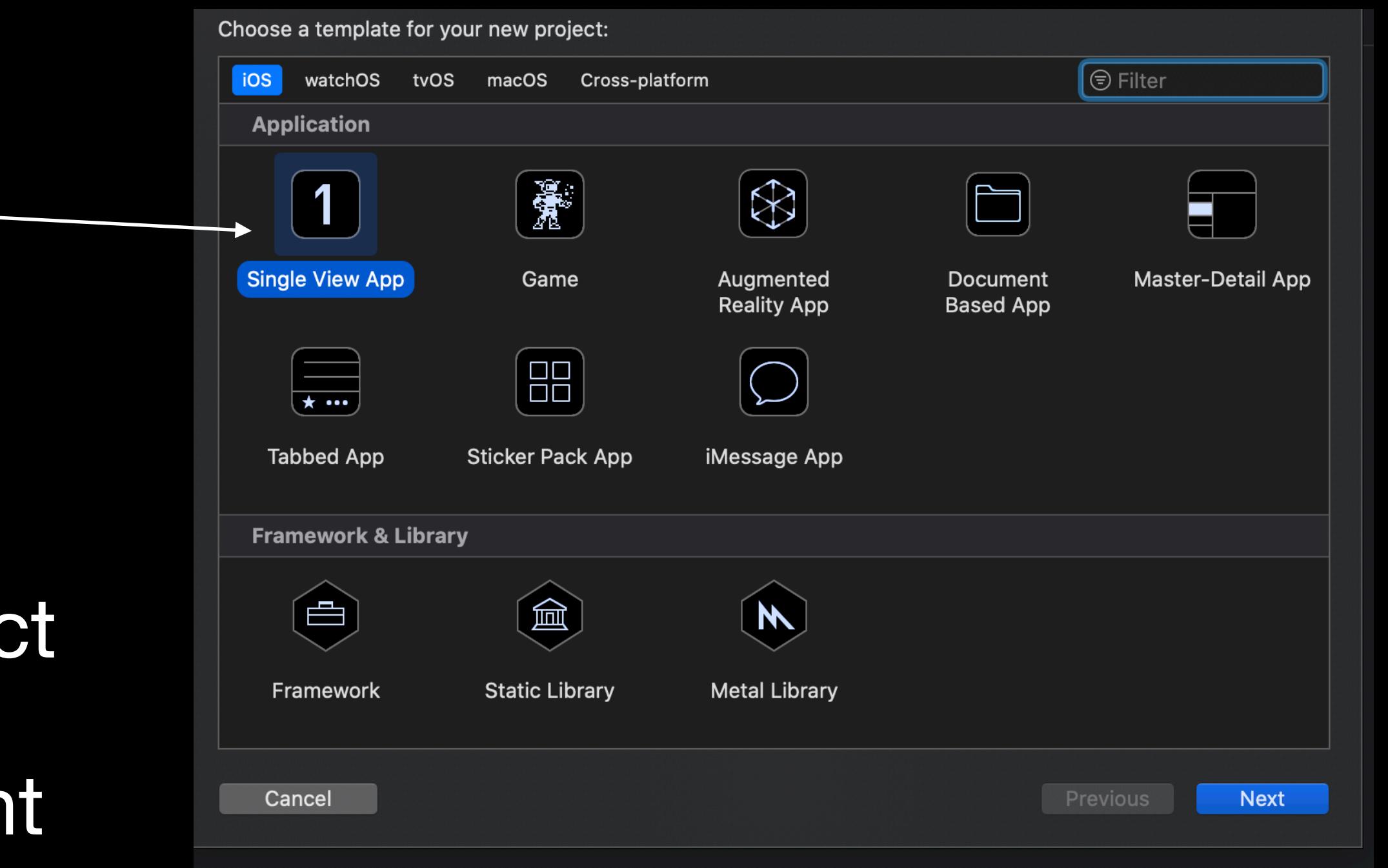
Step 1: Open Xcode →



Step 2: Create a new Xcode project

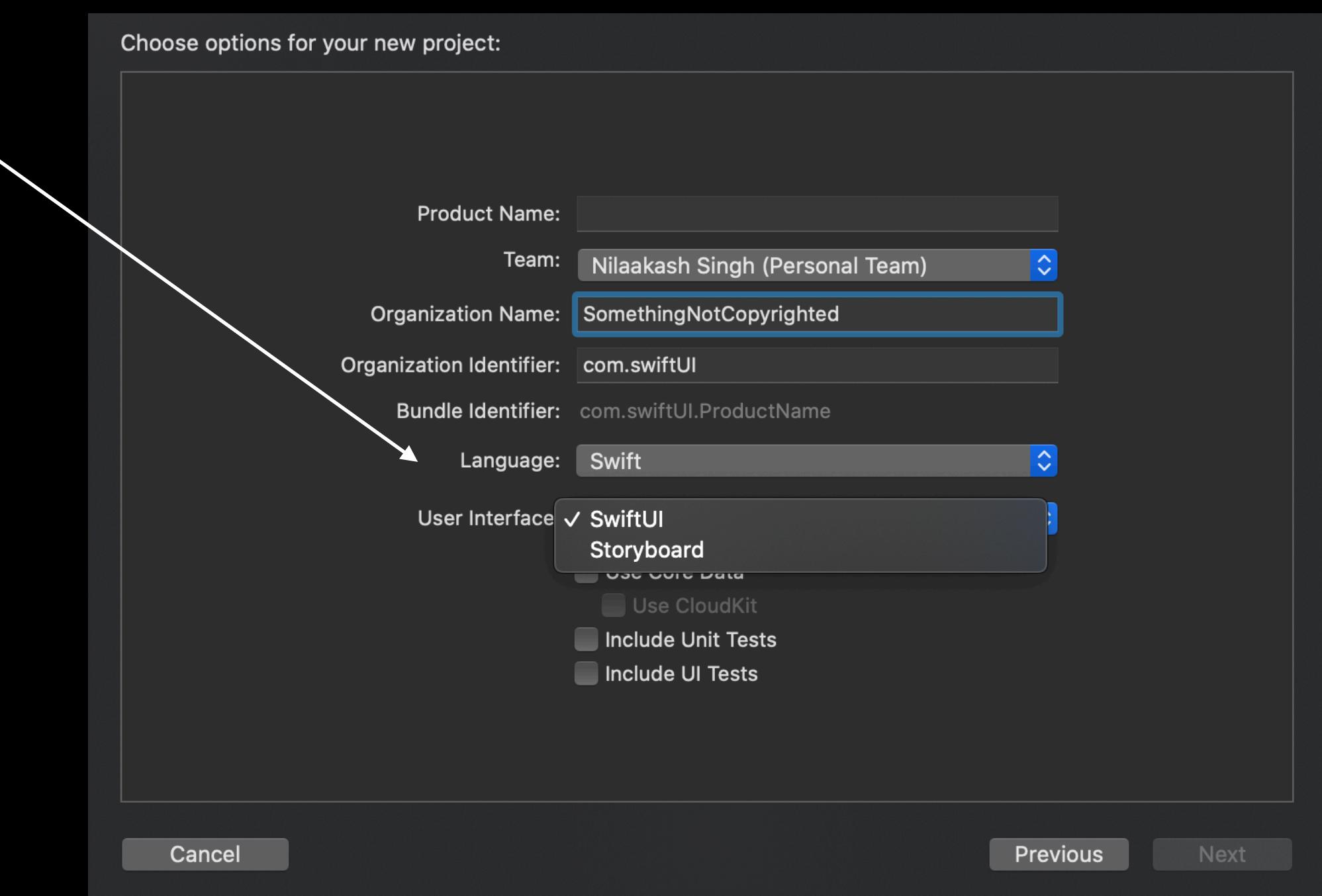


Step 3: Select Single View App and click next



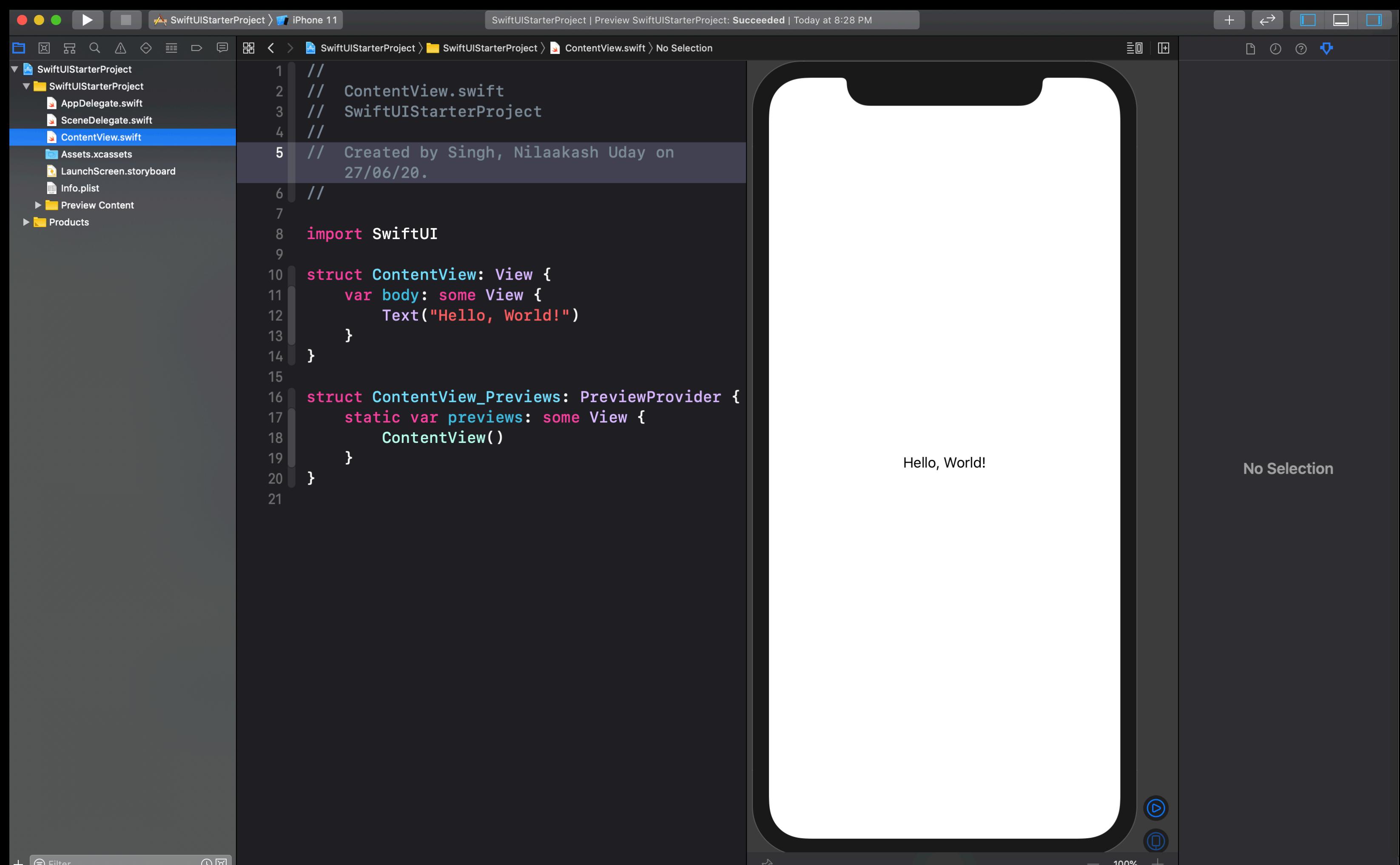
Step 4: Select SwiftUI as User Interface shown in Snippet. Add your project name as SwiftUI_Starter, Also please try not to add any copyright name in organisation.

Final Step: Once everything required is filled on screen to the right, next button will be enabled. Click Next to finish your Project Setup



Learning about SwiftUI Project structure

- This is how your Xcode starter project will look like
- We will get to know each component well as we advance in our challenge.
- For now we should only care about ContentView which is highlighted file in snippet



For some one who wants to learn more about Xcode Project structure

Source

<https://www.hackingwithswift.com/books/ios-swiftui/understanding-the-basic-structure-of-a-swiftui-app>

Inside Xcode you should see the following files in the space on the left, which is called the project navigator:

- AppDelegate.swift contains code for managing your app. It used to be common to add code here, but these days it's quite rare.
- SceneDelegate.swift contains code for launching one window in your app. This doesn't do much on iPhone, but on iPad – where users can have multiple instances of your app open at the same time – this is important.
- ContentView.swift contains the initial user interface (UI) for your program, and is where we'll be doing all the work in this project.
- Assets.xcassets is an *asset catalog* – a collection of pictures that you want to use in your app. You can also add colors here, along with app icons, iMessage stickers, and more.
- LaunchScreen.storyboard is a visual editor for creating a small piece of UI to show when your app is launching.
- Info.plist is a collection of special values that describe to the system how your app works – which version it is, which device orientations you support, and more. Things that aren't code, but are still important.
- Preview Content is a yellow group, with Preview Assets.xcassets inside – this is another asset catalog, this time specifically for example images you want to use when you're designing your user interfaces, to give you an idea of how they might look when the program is running.

Lets Play with text

So when you start your project, You will by default redirected to this file called **ContentView**

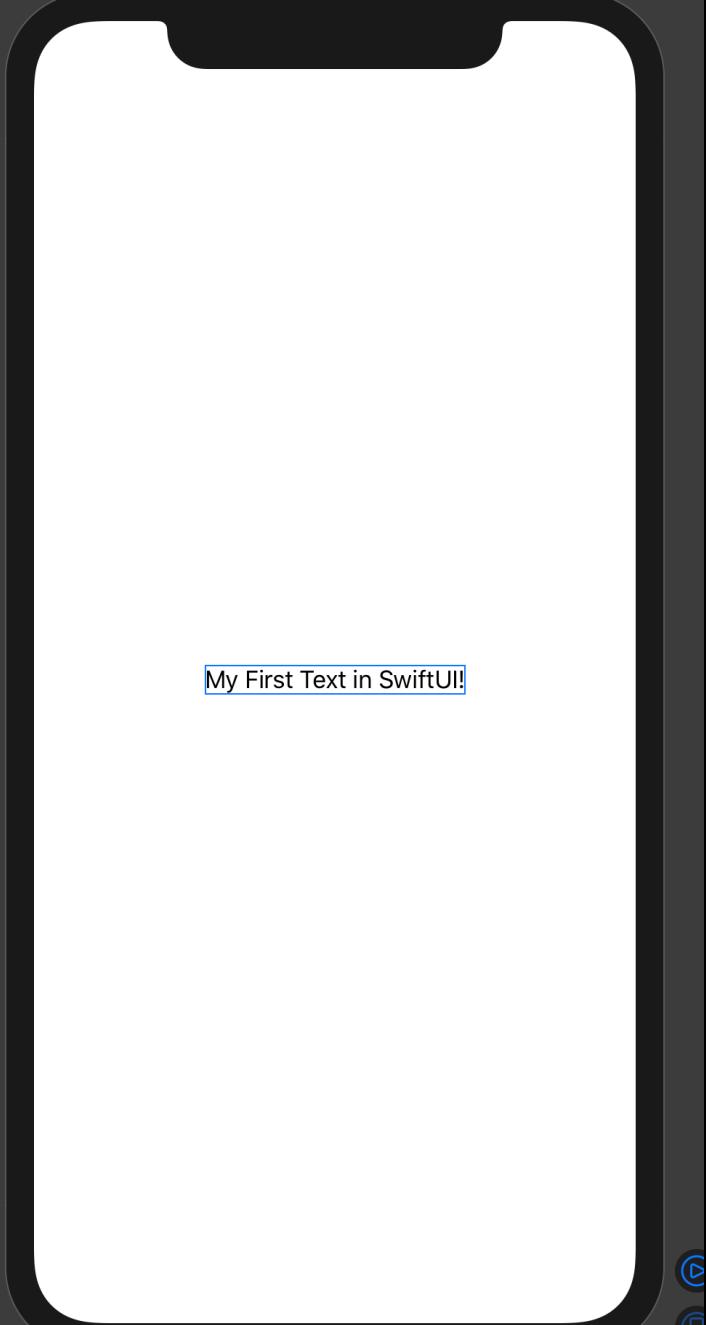
The syntax of Text is **Text("Any content")**

I just changed my Text content to "My First Text in SwiftUI"

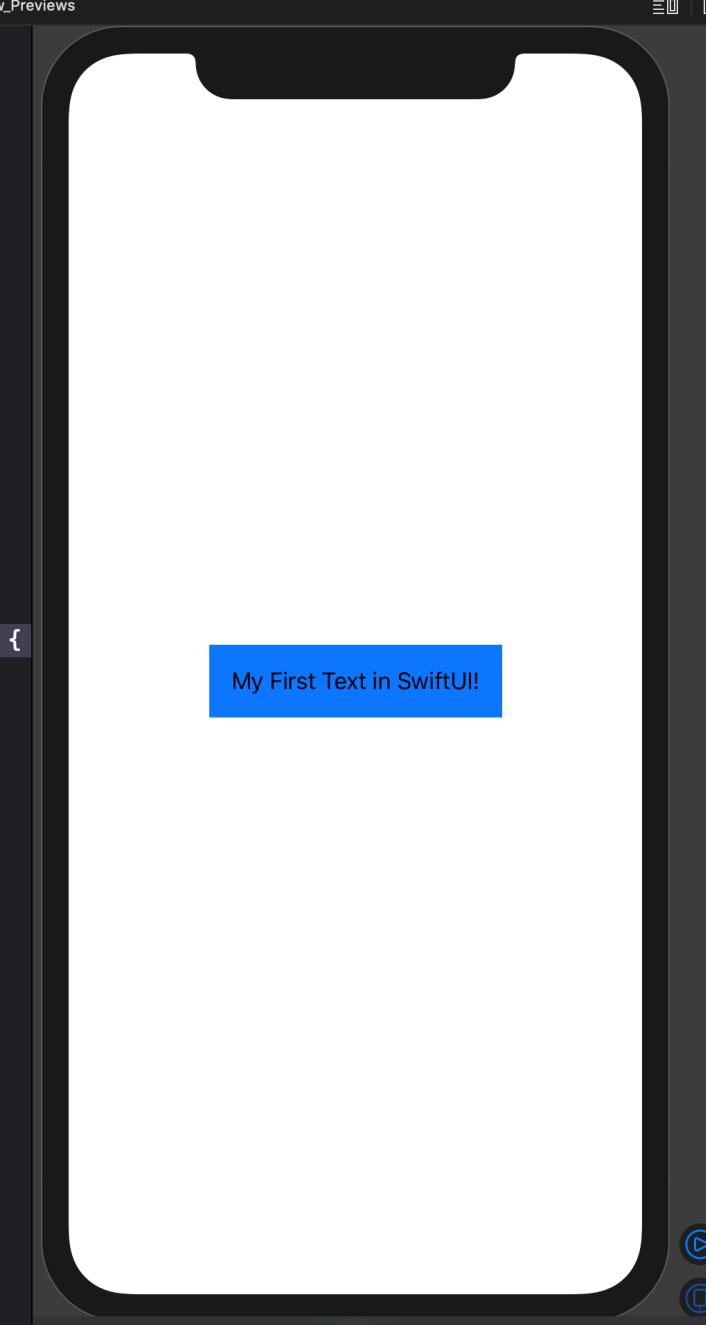
Now lets add some padding and background color to our text also we can see that canvas is getting updated.

Note: For those who don't have canvas support they can directly run the app

```
//  
// ContentView.swift  
// SwiftUIStarterProject  
//  
// Created by Singh, Nilaakash Uday on  
// 27/06/20.  
  
import SwiftUI  
  
struct ContentView: View {  
    var body: some View {  
        Text("My First Text in SwiftUI!")  
    }  
}  
  
struct ContentView_Previews: PreviewProvider {  
    static var previews: some View {  
        ContentView()  
    }  
}
```



```
//  
// ContentView.swift  
// SwiftUIStarterProject  
//  
// Created by Singh, Nilaakash Uday on  
// 27/06/20.  
  
import SwiftUI  
  
struct ContentView: View {  
    var body: some View {  
        Text("My First Text in SwiftUI!")  
            .padding()  
            .background(Color.blue)  
    }  
}  
  
struct ContentView_Previews: PreviewProvider {  
    static var previews: some View {  
        ContentView()  
    }  
}
```



Now as we see earlier, the colour blue doesn't look good with black text color.
So lets change the text color and also lets add corner radius to our Text, the output will be..

Screenshot of Xcode showing the ContentView.swift file and its preview in the iPhone X simulator. The code defines a struct ContentView with a body containing a Text view. The text is "My First Text in SwiftUI!". It has padding, a blue background, white foreground color, and a corner radius of 20. The preview shows a blue rounded rectangle with the text "My First Text in SwiftUI!" inside.

```
1 // ContentView.swift
2 // SwiftUIStarterProject
3 // Created by Singh, Nilaakash Uday on
4 // 27/06/20.
5 //
6 import SwiftUI
7
8 struct ContentView: View {
9     var body: some View {
10         Text("My First Text in SwiftUI!")
11             .padding()
12             .background(Color.blue)
13             .foregroundColor(Color.white)
14             .cornerRadius(20)
15     }
16 }
17
18 struct ContentView_Previews: PreviewProvider {
19     static var previews: some View {
20         ContentView()
21     }
22 }
23
24
25
```

Lets give final touch to our text with Shadow!
Shadow !!! Yaa you read it right, it's just one line of code for shadow. See the snippet

Screenshot of Xcode showing the ContentView.swift file with an additional line of code to add a shadow. The code now includes .shadow(radius: 5) after the corner radius. The preview in the iPhone X simulator shows the same blue rounded rectangle with the text "My First Text in SwiftUI!", but now with a visible shadow below it.

```
1 // ContentView.swift
2 // SwiftUIStarterProject
3 // Created by Singh, Nilaakash Uday on
4 // 27/06/20.
5 //
6 import SwiftUI
7
8 struct ContentView: View {
9     var body: some View {
10         Text("My First Text in SwiftUI!")
11             .padding()
12             .background(Color.blue)
13             .foregroundColor(Color.white)
14             .cornerRadius(20)
15             .shadow(radius: 5)
16     }
17 }
18
19 struct ContentView_Previews: PreviewProvider {
20     static var previews: some View {
21         ContentView()
22     }
23 }
24
25
```

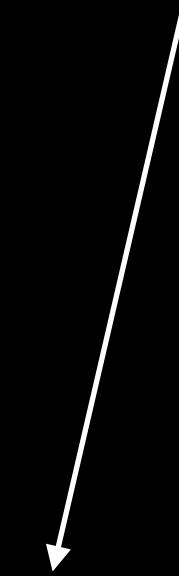
Important note: With addition of every new line, please try to look at output that is how even I develop the UI :)

Moto: - Assume less see more

Thats all for week 1 folks



Feel free to play with your Text more,
for eg: Adding different background colours,
Adding fonts, Making text italic, bold and as
much as your creativity goes :)



Also remember to raise the Pull Request to master repo updating Read.me,
Once you commit all your changes to your own Repo.

Master Repo Link : <https://github.com/NilaakashSingh/100DaysofSwiftUI>

Readme update format- [CommitLink]: Author name

We will be back next week.... to be continued