

SWIFTUI

Lecture #6 · Lukas Smetana

STRV

FIRST THINGS FIRST

- Attendance
- Charger
- Questions from last lecture

LECTURE #6 GOAL

- @State property
- Unidirectional data flow in SwiftUI
- **Characters screen of our Rick and Morty app**



@STATE

- Fundamental concept in SwiftUI
- Property wrapper
- What is it doing?
 - Read/write
 - Persists Data when View is recreated

```
var randomString = "Hello world!"
```

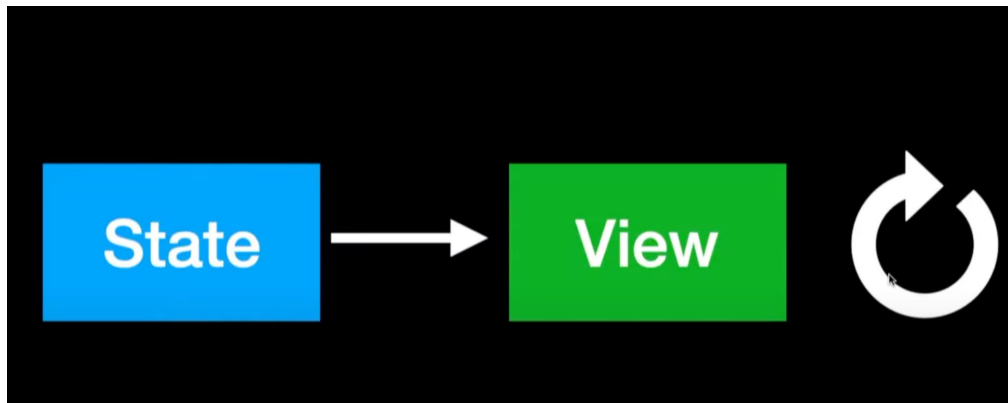
```
@State var randomString = "Hello world!"
```

@STATE

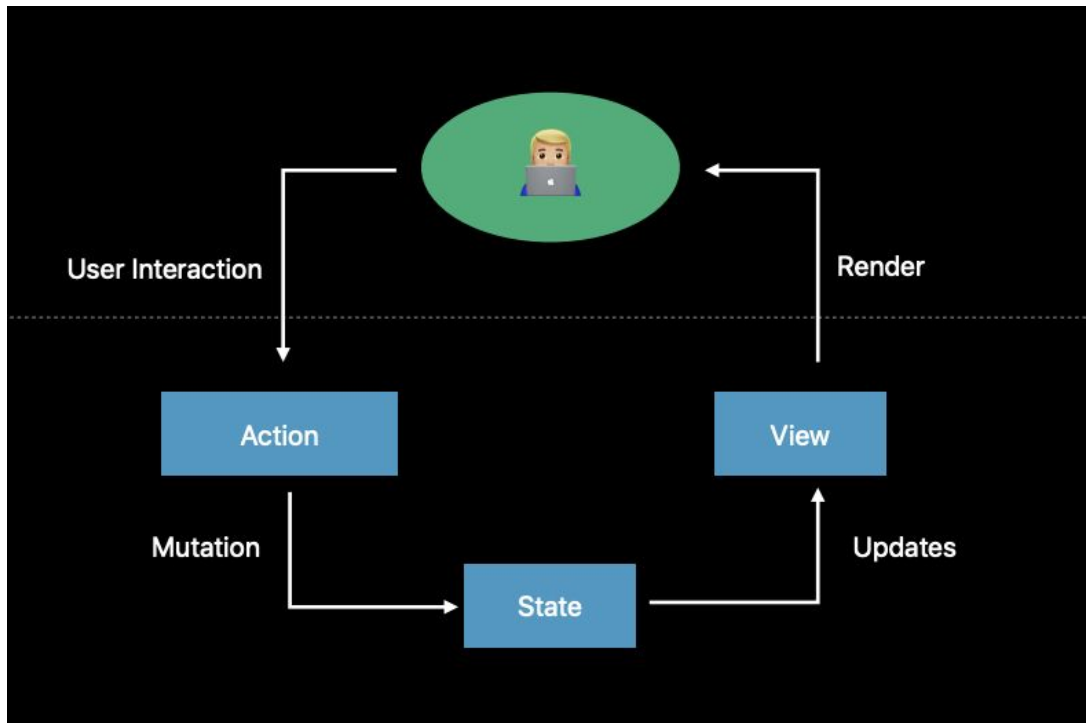
```
struct PlayButton: View {  
    @State private var isPlaying: Bool = false // Create the state.  
  
    var body: some View {  
        Button(isPlaying ? "Pause" : "Play") { // Read the state.  
            isPlaying.toggle() // Write the state.  
        }  
    }  
}
```

DATA FLOW

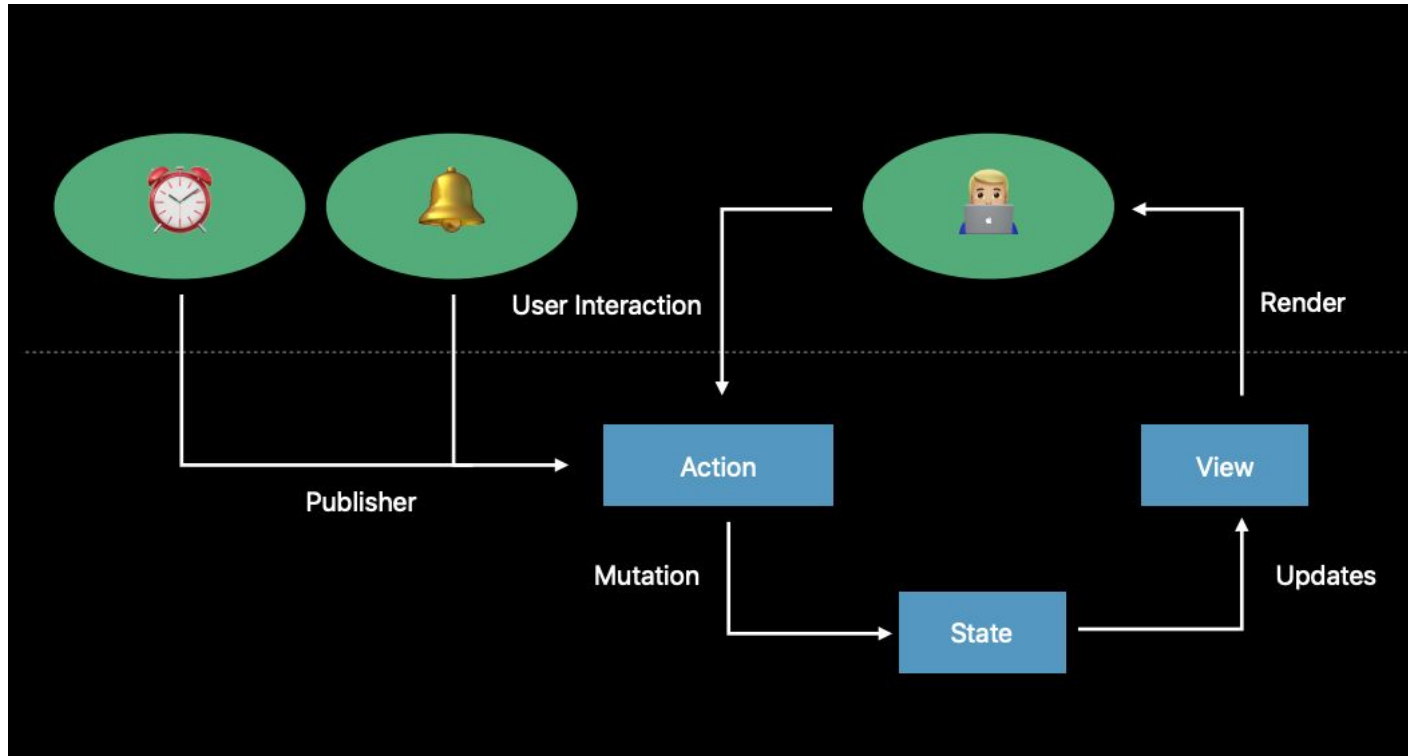
- Data flows differently in SwiftUI and UIKit applications
- Views in SwiftUI are state driven
- State is the only source of truth for the View
- By manipulating the state, we update the View



DATA FLOW



DATA FLOW - EXTERNAL EVENTS



DATA FLOW

- Data flows differently in SwiftUI and UIKit applications
- Views in SwiftUI are state driven
- State is the only source of truth for the View
- By manipulating the state, we update the View
- **Data flow is unidirectional**

ADDITIONAL RESOURCES

- [@State Property Wrapper Explained](#)
- WWDC Video by Apple - [Data Flow Through SwiftUI](#)
- [Intro to SwiftUI Data Flow](#)

LETS CODE





THANK YOU!

www.strv.com / [@strvcom](https://twitter.com/strvcom)

STRV