

# SwiftUI Basics

Gary Tokman

# Agenda

- Benefits
- Syntax
- Examples



# WWDC 2019

**\* Apple releases SwiftUI \***

**iOS Developers:**



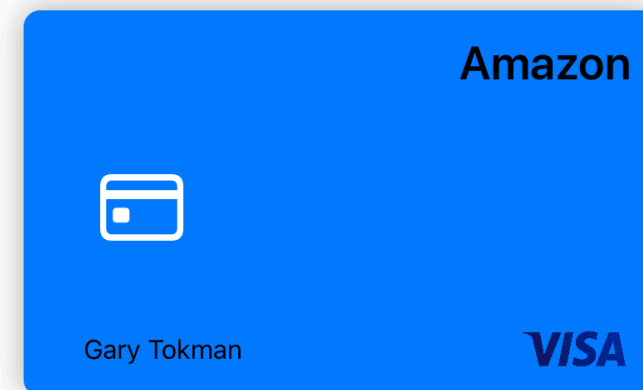
# What is it? Benefits?

- Declarative views library
  - 🎉 No more storyboards 🎉
    - Build views in code
  - No UIKit, brand new framework
  - Everything is declarative, building composable views
  - 🎉 No more view controllers (RIP MVC) 🎉
    - Lifecycle, less repetition, binding

```

9  import SwiftUI
10
11  struct ContentView: View {
12      var body: some View {
13          // Outmost container
14          VStack(alignment: .leading, spacing: 50) {
15              // Top Label
16              HStack {
17                  Spacer()
18                  Text("Amazon")
19                      .font(.title)
20                      .fontWeight(.bold)
21              }
22              .padding(.top, 20)
23
24              // Middle Image
25              Image(systemName: "creditcard")
26                  .resizable()
27                  .frame(width: 50, height: 40)
28                  .padding(.leading, 30)
29                  .foregroundColor(.white)
30
31              // Bottom Text and Image
32              HStack {
33                  Text("Gary Tokman")
34                      .font(.callout)
35                      .fontWeight(.medium)
36                  Spacer()
37                  Image("visa")
38                      .resizable()
39                      .frame(width: 70, height: 30)
40              }
41              .padding([.leading, .bottom], 20)
42          }
43          .frame(maxWidth: 350, maxHeight: 200)
44          .padding()
45          .background(Color.blue)
46          .cornerRadius(10)
47          .shadow(radius: 10)
48      }
49  }

```



- Canvas - visual builder and live preview
- Dark mode
- Localization and directional language support
- SF Symbols

- New view protocol
- Opaque Type
- Views
- View Modifiers

```
import SwiftUI

struct ContentView: View {
    var body: some View {
        // Outmost container
        VStack(alignment: .leading, spacing: 50) {
            // Top Label
            HStack {
                Spacer()
                Text("Amazon")
                    .font(.title)
                    .fontWeight(.bold)
            }
                .padding(.top, 20)

            // Middle Image
            Image(systemName: "creditcard")
                .resizable()
                .frame(width: 50, height: 40)
                .padding(.leading, 30)
                .foregroundColor(.white)

            // Bottom Text and Image
            HStack {
                Text("Gary Tokman")
                    .font(.callout)
                    .fontWeight(.medium)
                Spacer()
                Image("visa")
                    .resizable()
                    .frame(width: 70, height: 30)
            }
                .padding([.leading, .bottom], 20)
        }
        .frame(maxWidth: 350, maxHeight: 200)
        .padding()
        .background(Color.blue)
        .cornerRadius(10)
        .shadow(radius: 10)
    }
}
```

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

- View **protocol**
  - One requirement - **body**
  - Computed property - **generated** every time it's **read**
- Some keyword
  - Opaque type - Swift 5.1
  - Return any type of **View** regardless of behavior

```
VStack {  
    Text("Hello World!")  
    Text("Hello World!")  
    Text("Hello World!")  
    Text("Hello World!")  
    Text("Hello World!")  
    Text("Hello World!")  
    Text("Hello World!")  
    Text("Hello World!")  
    Text("Hello World!")  
    Text("Hello World!")  
  
    //      Text("Hello World!") // Cause Error  
}  

```

- 10 child view max
- @ViewBuilder limitation
- Need more child views?



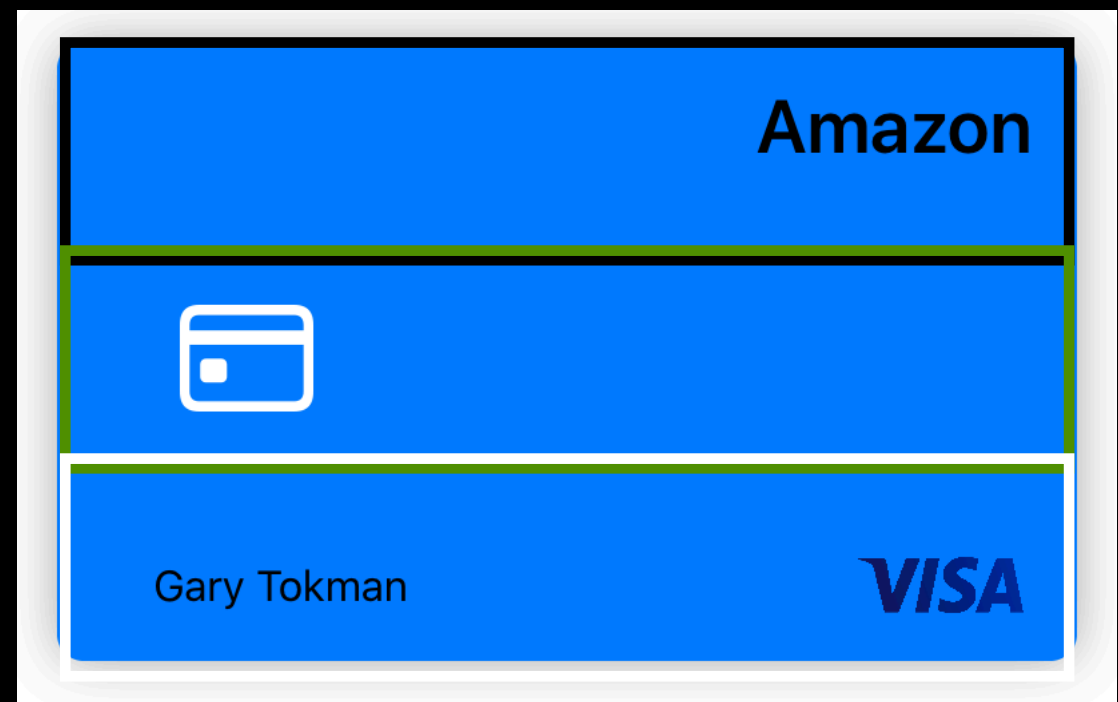
```
VStack {  
    Text("Hello World!")  
    Text("Hello World!")  
    Text("Hello World!")  
    Text("Hello World!")  
    Text("Hello World!")  
    Text("Hello World!")  
    Text("Hello World!")  
    Text("Hello World!")  
    VStack {  
        Text("Hello World!") // 9  
        Text("Hello World!") // 10  
        Text("Hello World!") // 11  
    }  
}
```

- Use another container
  - VStack, HStack, List, Group, ForEach, Section...

# VStack

```
VStack(alignment: .leading, spacing: 50) {  
    // ...  
}
```

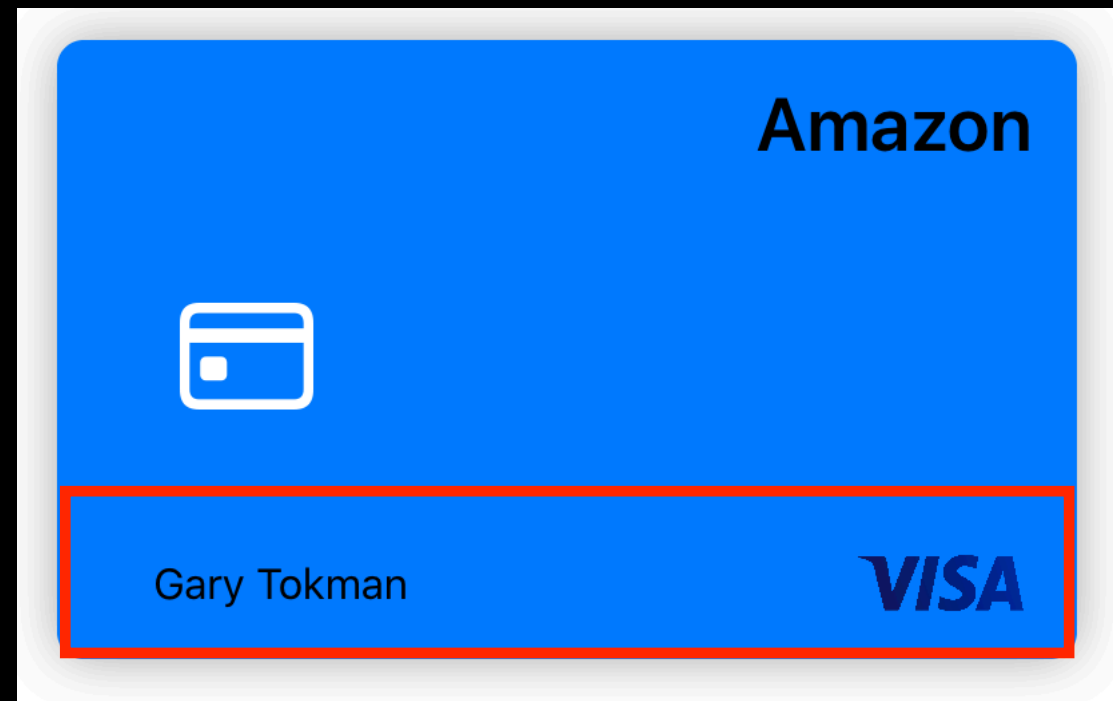
- Arrange views in vertical grid
- Spacing and alignment



# HStack

- Arrange views in horizontal grid
- Spacing and alignment

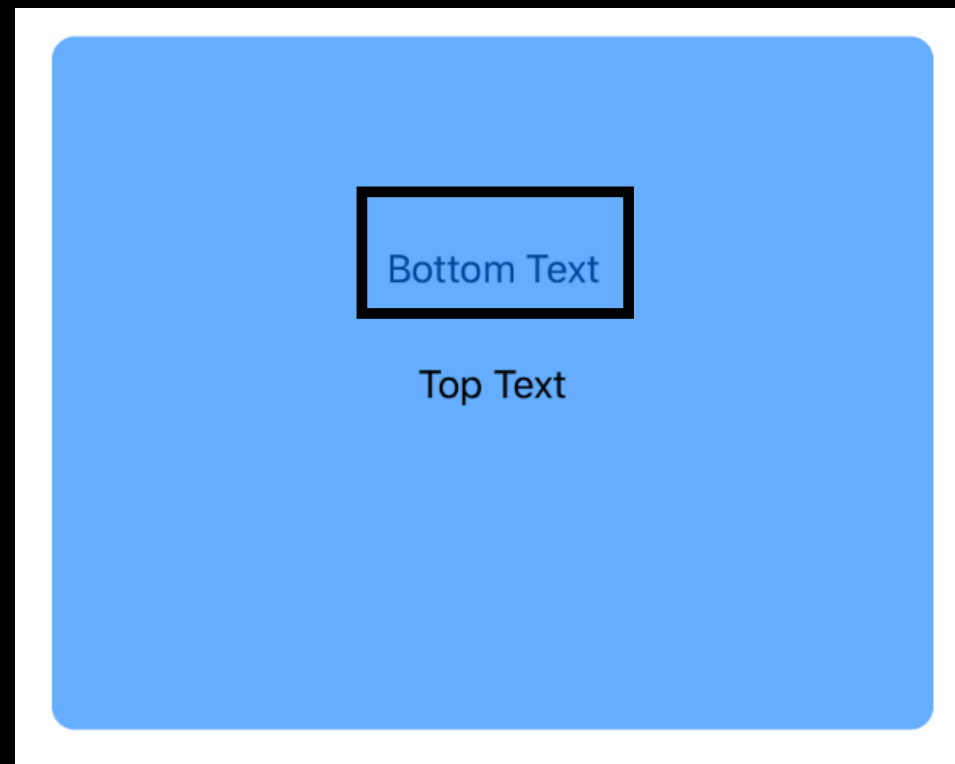
```
// Bottom Text and Image
HStack {
  Text("Gary Tokman")
    .font(.callout)
    .fontWeight(.medium)
  Spacer()
  Image("visa")
    .resizable()
    .frame(width: 70, height: 30)
}
.padding([.leading, .bottom], 20)
```



# ZStack

- Arrange views in overlapping (layering)
- Spacing and alignment

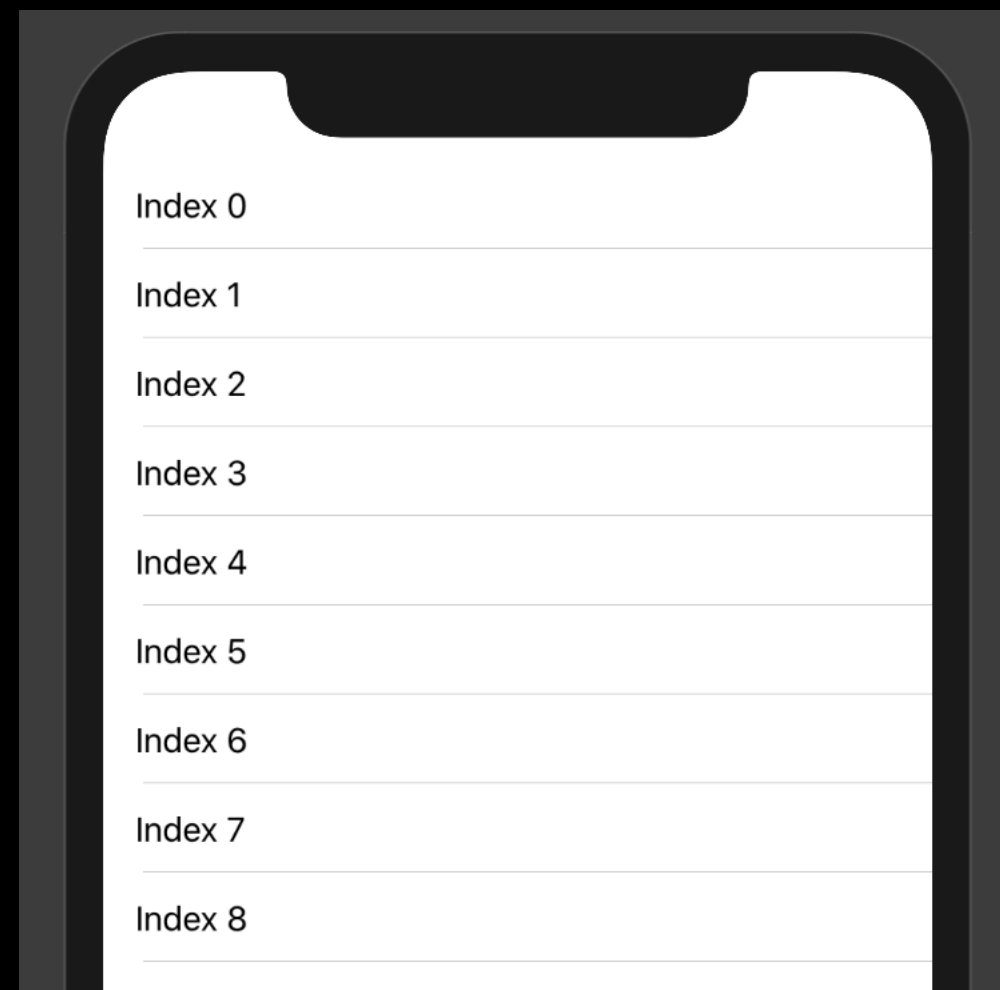
```
ZStack {  
    Text("Bottom Text")  
        .offset(y: -50)  
    RoundedRectangle(cornerRadius: 10)  
        .frame(  
            maxWidth: .infinity,  
            maxHeight: 300  
        )  
        .padding()  
        .foregroundColor(.blue)  
        .opacity(0.60)  
    Text("Top Text")  
}
```



# List

```
var body: some View {  
    List(0...10, id: \.description) { index in  
        Text("Index \("\(index)")")  
    }  
}
```

- Arrange views in vertical list
- A lot of different **inits**



# ForEach

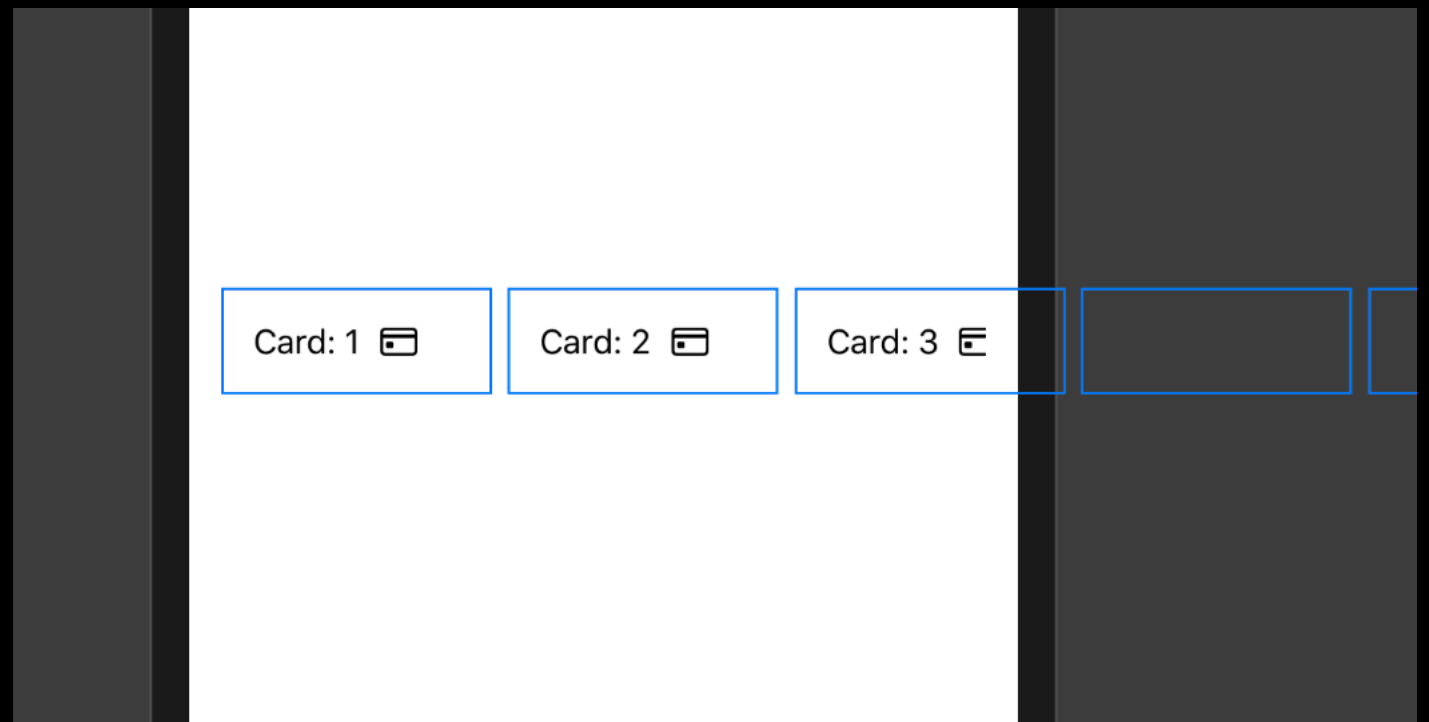
- Add views many times
- Don't **repeat** yourself

[illegible][illegible]

# ScrollView

```
ScrollView(.horizontal, showsIndicators: true) {  
    HStack {  
        ForEach(1..10, id: \.self) { index in  
            HStack(alignment: .center, spacing: 10.0) {  
                Text("Card: \(index)")  
                Image(systemName: "creditcard")  
                Spacer()  
            }  
        }.padding()  
    }  
}.padding()
```

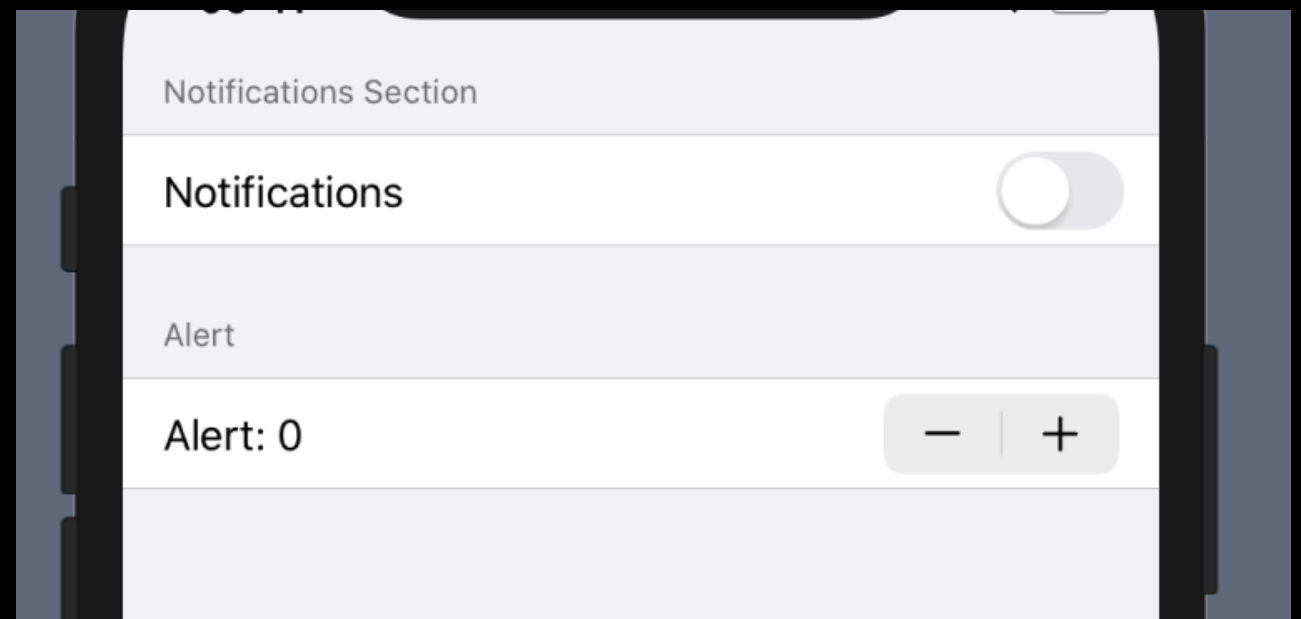
- Add views many times in vertical or horizontal axis
- Don't repeat yourself



# Form

```
Form {  
  Section(header: Text("Notifications  
Section")) {  
    Toggle(isOn: $isEnabled) {  
      Text("Notifications")  
    }  
  }  
}
```

- Add sections views
- Use for settings or static lists
- Stepper, Picker, Slider

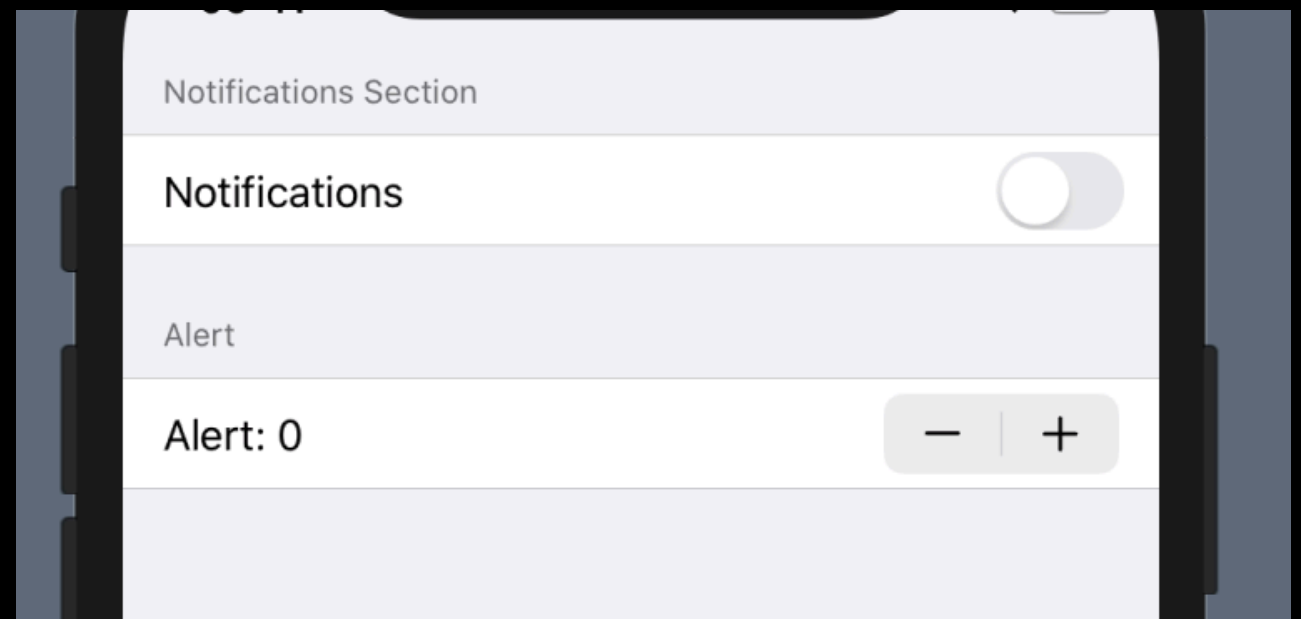




# Form

```
Form {  
  Section(header: Text("Notifications  
Section")) {  
    Toggle(isOn: $isEnabled) {  
      Text("Notifications")  
    }  
  }  
}
```

- Add sections views
- Use for settings or static lists
- Stepper, Picker, Slider



## Tap Gesture

- Use a binding to toggle an animation
- Lots of modifiers to transform
- Many Animations

```
@State private var flip = false
var body: some View {
    // Outmost container
    Card()
        .rotationEffect(Angle.init(degrees:
self.flip ? 90 : 0))
    // .animation(.default)
    .animation(.spring())
    .onTapGesture {
        self.flip.toggle()
    }
}
```



```

@State private var flip = false
@State private var position = CGSize.zero

```

```

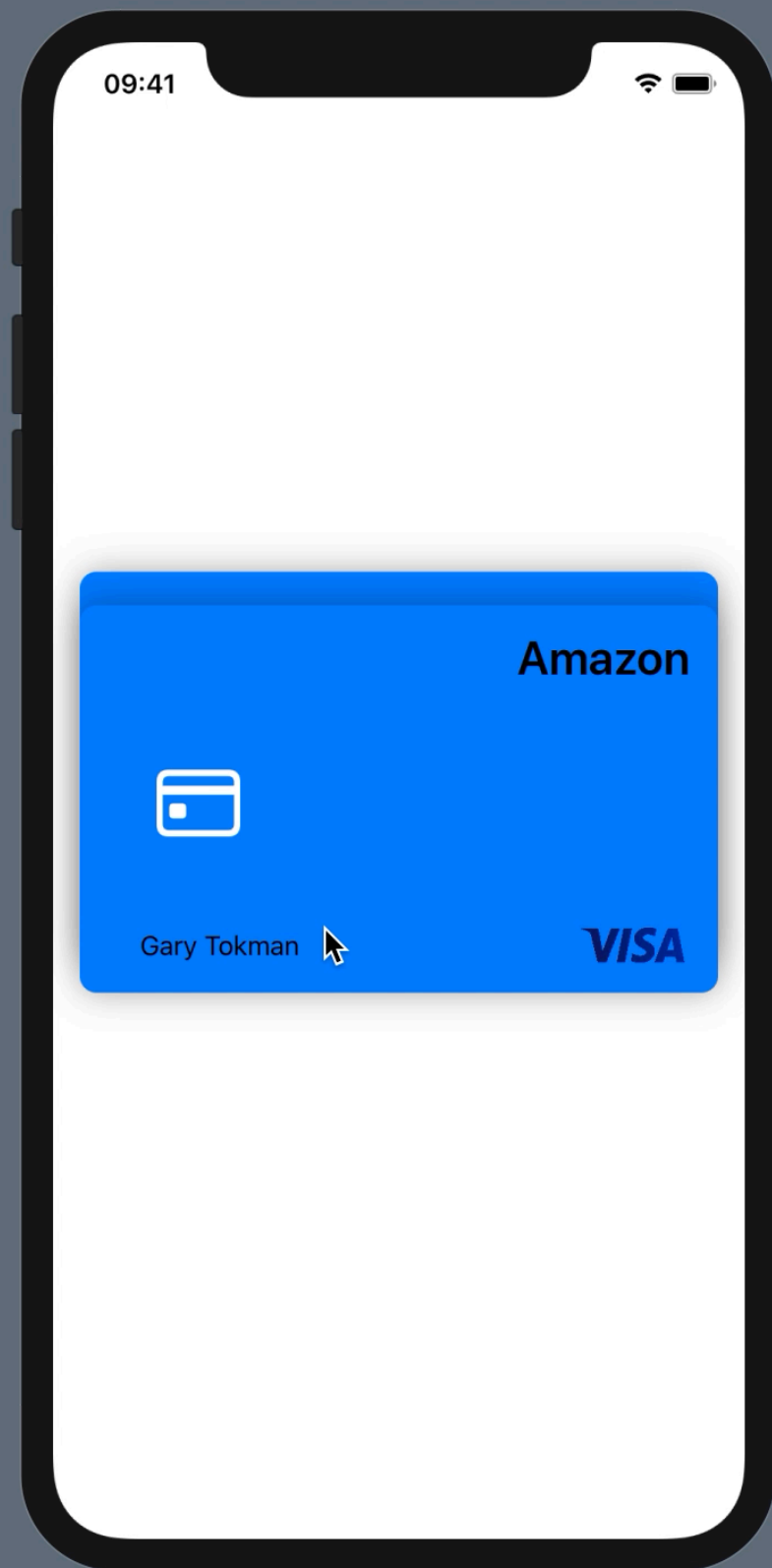
var body: some View {
    ZStack {
        Card()
            .animation(.spring())
            .offset(x: 0, y: flip ?
-100 : -20)
            .offset(x: position.width, y:
position.height)

```

```

        Card()
            .rotationEffect(Angle.init(degree
s: self.flip ? 20 : 0))
            .offset(x: position.width, y:
position.height)
            .animation(.spring())
            .onTapGesture {
                self.flip.toggle()
            }
            .gesture(DragGesture()
                .onChanged { (value) in
                    self.position =
value.translation
                    self.flip = true
                }.onEnded { _ in
                    self.position = .zero
                    self.flip = false
                } )
    }
}

```



Preview

# UIKit?

- UIViewControllerRepresentable & UIViewRepresentable

```
struct ActivityIndicator: UIViewRepresentable {
```

```
    func makeUIView(context: Context) -> UIActivityIndicatorView {  
        let v = UIActivityIndicatorView(style: .large)
```

```
        return v  
    }
```

```
    func updateUIView(_ activityIndicator: UIActivityIndicatorView, context:  
Context) {  
        activityIndicator.startAnimating()  
    }  
}
```

# Resources

- <https://www.hackingwithswift.com>
- <https://www.raywenderlich.com>
- <https://developer.apple.com/tutorials/swiftui>

**The End**

**Twitter:**  
**@bestgaryever**