

SwiftUI Basics

Cody Henrichsen

May 2020

SwiftUI Overview

- ▶ Designed to make Views reusable across platforms
- ▶ Simplify layout and animation
- ▶ Component names reflect purpose
- ▶ Simple initialization based on String for most
- ▶ Event handlers attached directly

ContentView.swift

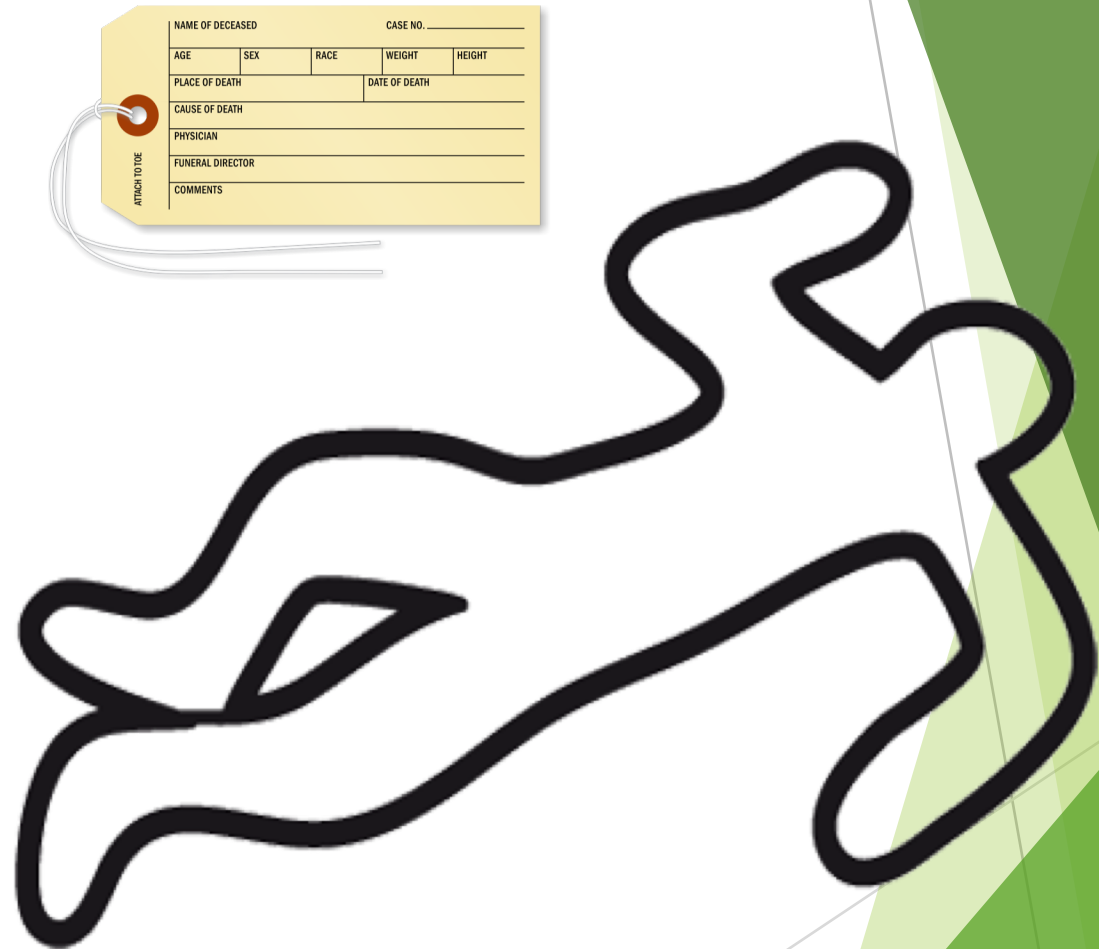
- ▶ Default SwiftUI file when creating an app
- ▶ Composed of two structs
 - ▶ ContentView
 - ▶ Holds the contents of the screen
 - ▶ ContentView_Previews
 - ▶ Displays contents in the Canvas Preview
 - ▶ Not required

How do I make a screen?

- ▶ Make a .swift file
- ▶ Import SwiftUI
- ▶ Make a struct that conforms to the View protocol
 - ▶ Required var
 - ▶ body : some View
 - ▶ No required methods

The body?

- ▶ A computed property (var)
- ▶ Must have something in it
 - ▶ A single built in component
 - ▶ A Stack of components
 - ▶ Custom component



Basic SwiftUI Components

- ▶ Label
 - ▶ `Text("What it should say")`
- ▶ Picture
 - ▶ `Image("Name seen in xcAssets")`
 - ▶ No GIF!
- ▶ Button
 - ▶ `Button("Text"){ //Handler code }`

Stacks

- ▶ Vertical
 - ▶ A
 - ▶ B
 - ▶ C
- ▶ Horizontal
 - ▶ A B C
- ▶ Depth (Z-Order)
 - ▶ Layered on top of each other AKA bad for text
 - ▶ Cool with images/colors

Simple VStack

```
var body: some View
{
    VStack
    {
        Text("Demo")
            .padding(.all, 3.0)
        Image("mini-octo")
        Text("more text here")
        Button("No real stuff")
        {
            //Button code should be
            here
        }
    }
}
```

Demo



more text here
No real stuff

Simple HStack

```
var body: some View
{
    HStack
    {
        Text("Demo")
            .padding(.all, 3.0)
        Image("mini-octo")
        Text("more text here")
        Button("No real stuff")
        {
            //Button code should be
            here
        }
    }
}
```

Demo



more text here No real stuff

Simple ZStack

```
var body: some View
{
    ZStack
    {
        Text("Demo")
            .padding(.all, 3.0)
        Image("mini-octo")
        Text("more text here")
        Button("No real stuff")
        {
            //Button code should be
            here
        }
    }
}
```



Attributes

- ▶ Each method in the component returns `self` so attribute modifications stack easily
- ▶ `Component("blah")`
 - ▶ `.modify()`
 - ▶ `.otherChange(.typeName, varName: value, other: otherVal)`

```
Text("Demo")  
    .fontWeight(.heavy)  
    .foregroundColor(Color.green)  
    .padding(.all, 3.0)  
    .blur(radius: CGFloat(0.93))  
Image("mini-octo")
```

Each effect stacks!!

