Swift UI

A declarative user interface for developing apps on every Apple platform. SwiftUI provides views, controls, and layout structures for declaring your app user interface. The framework provides event handlers for delivering taps, gestures, and other types of input, and tools to manage the flow of data from app's models down to the views and controls. Apple Official Documnentation

Swift UI uses the Swift programming language Swift Notes

Your first view

Swift UI does a great job separating the interface from the logic. A View and most of the types we are gonna use in Swift UI are Struct. Struct are commons in other language programs are an abbreviation for Data Structure. A struct can be seen as a collections of variables, however we'll see they are powerful types on which most of the Swift UI elements are built up. Struct have variables but can also have functions. They are similar to classes but we'll see they lack inheritance. Struct are not object oriented things. Swift support also classes and supports both Object Oriented Programming [illegible] Functional Programming. We'll

Animation

Animation is very important in a mobile UI. Swift makes very easy to do. There are basically two ways to do animation: 1. by animating a Shape 2. by animating Views via their ViewModifiers

So what is a ViewModifier?

View modifiers are all those little functions that modified our Views (like aspectRatio, padding etc) They are (likely) turning right aroud and calling a function in View called Modifier.

e.g. .aspectRatio(2/3) is likely something like .modifier(AspectModifier(2/3)) AspectModifier can be anything that conforms to the ViewModifier protocol.

Apple Documentation - View Modifier A modifier that you apply to a view or another view modifier, producing a different version of the original value.

ViewModifier is a protocol that lets us create a reusable modifier that can be applied to any view.

Conceptually, this protocol is sort of like this...

```
protocol ViewModifier{
    typealias Content //the type of the View passed to body
```