# Data Persistence

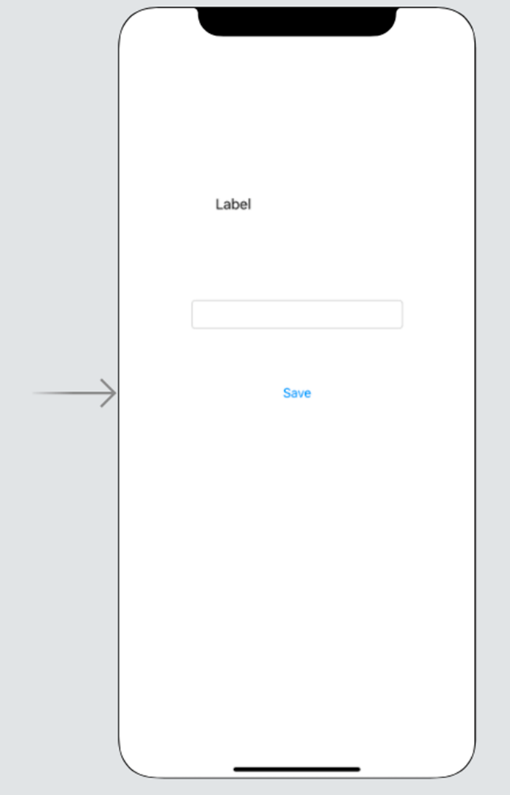
The Game Base we developed previously would benefit from a memory of some type. For example, this could be used to save a game part-way through and continue it at a later date. This is referred to as "data persistence" and is one of the elements expected in the coursework.

There are several different ways of achieving data persistence in Swift, namely UserDefaults, Keychain, Saving files to disk, Core Data, SQLite, and Property Lists.

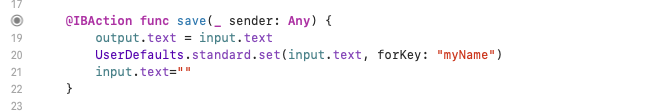
In this workbook we will be looking at UserDefaults. UserDefaults are easy to use and have a simple API. They are best used for small pieces of data.

## Simple UserDefaults Example

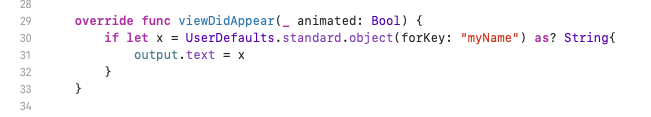
1. Create a new Single View Application
2. Add a Label to the Main.storyboard
3. Add an Outlet for the label called output
4. Add a Text Field to the Main.storyboard
5. Add an Outlet for the Text Field called input
6. Add a Button to the Main.storyboard
7. Name the Button save



1. Add an Action to the save Button on Touch Up Inside
2. Add the following code to the Action

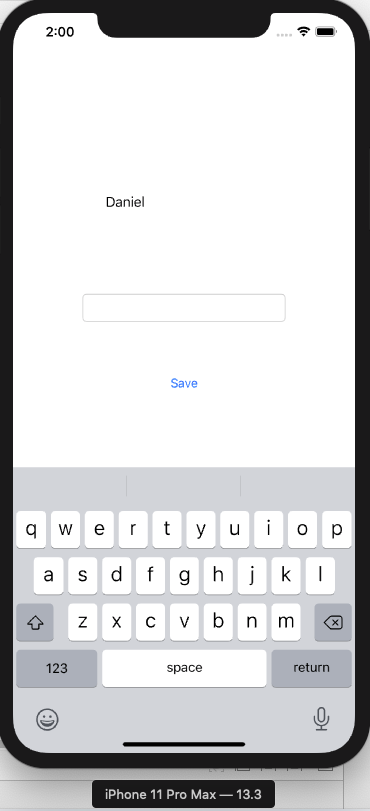
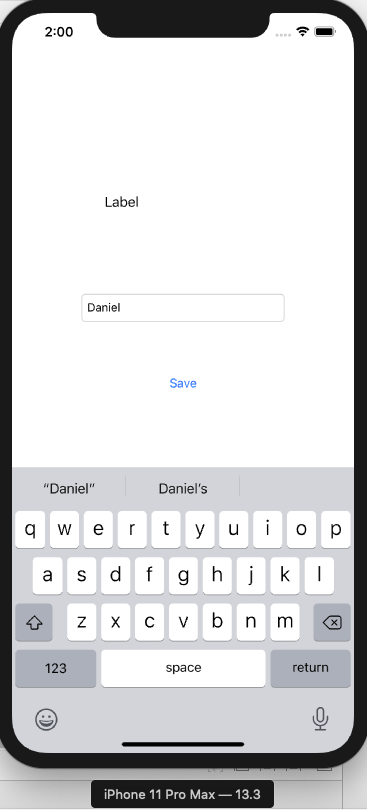
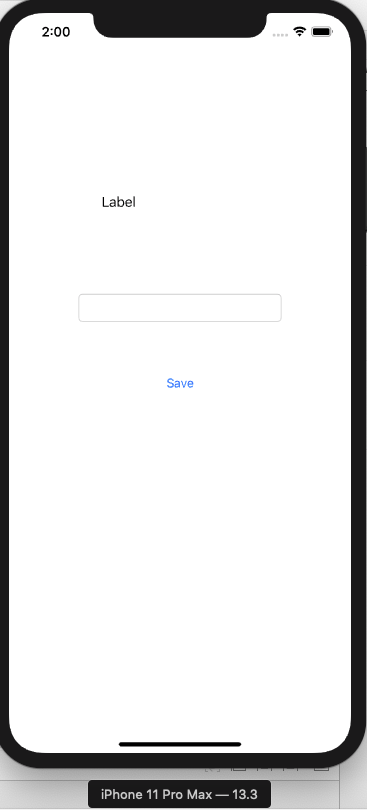


1. Add the following function. This will run when the view appears.



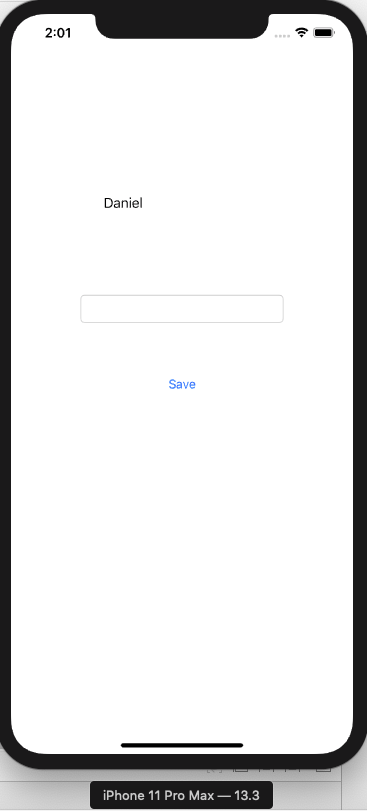
1. Run the app in the simulator

When we type some text into the input Text Field and press the save Button, the text is displayed in the output Label.



1. Stop the simulator.
2. Run the simulator again.

We will see that the text we entered last time we ran the app is displayed in the output Label – it has persisted when the app was closed.

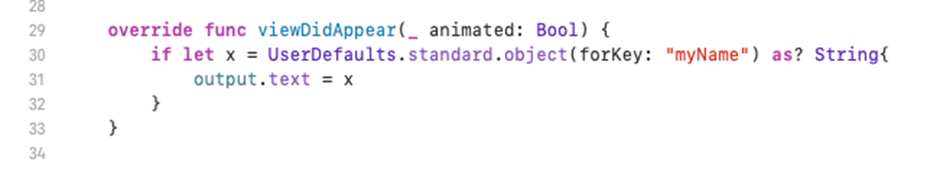


The underlying data structure we are using is a property list file (a "plist"). We can think of it as essentially being a Dictionary.

If we look at the code we can see on line 20 (my image line numbers) that we are setting the value of the myName key within UserDefaults to be equal to the text from input (the Text Field).



If we look at line 30, we are retrieving an object from UserDefaults, using the myName key and expecting an optional String. If we successfully retrieve a value, it is assigned to the text of output (the Label).



Obviously, we can define whatever keys we need. We can also use other data types, so we can store a Bool, or an Array for example.