Data Persistence in IOS - User Defaults

Jump to a Section

Data Persistence Options in IOS

In IOS, we can persist data to:

- User Defaults
- Core Data
- Cloud based storage (example: Firebase)

1/ User Defaults

- This is key-value storage
- Offline storage (saved data is local to the device and cannot be accessed outside the device)
- Data is stored as key-value pairs
- Best for saving simple data, like numbers, strings, simple arrays, etc
- Similar to localStorage in web programming

2/ Core Data

- Interface for saving data to an relational database (SQLite)
- Offline storage (saved data is local to the device and cannot be accessed outside the device)
- Used for modelling data as tables; and, describing relationships between data

3/ Cloud based service (Firebase)

- Advantages: Data is saved to the cloud, so it can be accessed anywhere and on any device.
- There are many service providers of cloud based data, including Parse, Realm, CloudKit, and Firebase
- Firebase is a no-sql database that is accessible cross platform (mobile and web)

What is User Defaults?

- User Defaults is a key-value storage provided by the IOS framework.
- Used to persist *simple* data to the app: strings, numbers, the **Date** object.
- “Persisting data” means that the data is remembered even after the app closes.

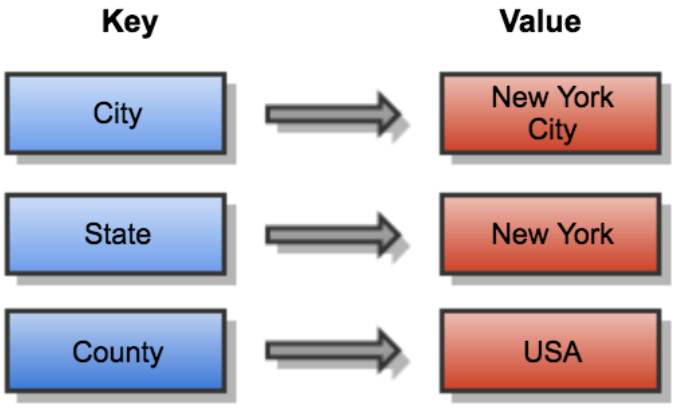
- Can store dictionaries and arrays, but only if the dictionary/array contains “simple data”
- Structs can be stored if they conform to the **Codable** protocol and converted to JSON format

If you come from a web-programming background, UserDefaults is similar to **localStorage**.

What is key-value storage?

In a key-value data store, information is stored as *key-value* pairs.

- The “key” is a unique identifier
- The “value” is the data you want to store



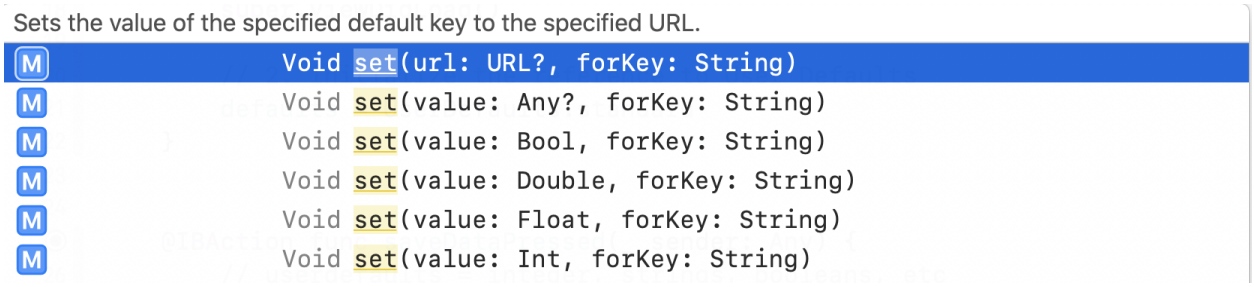
- Data is retrieved using the *key*

Initializing User Defaults, Saving and Retrieving Items

Creating an instance of UserDefaults:

```
var defaults:UserDefaults!
self.defaults = UserDefaults.standard
```

Adding an item to UserDefaults:




```
self.defaults.set("Peter", forKey: "name")    // string
self.defaults.set(77, forKey: "age")    // integer
self.defaults.set(true, forKey: "hasPet")    // boolean
```

Retrieving an item from User Defaults

```
let x = self.defaults.string(forKey: "name") // gets a string out of the userdefaults
let y = self.defaults.integer(forKey: "age") // get a int out of user defaults
let z = self.defaults.bool(forKey: "hasPet") // get a boolean out of user defaults
```

Code Example: Saving and Retrieving Data from User Defaults

Test:

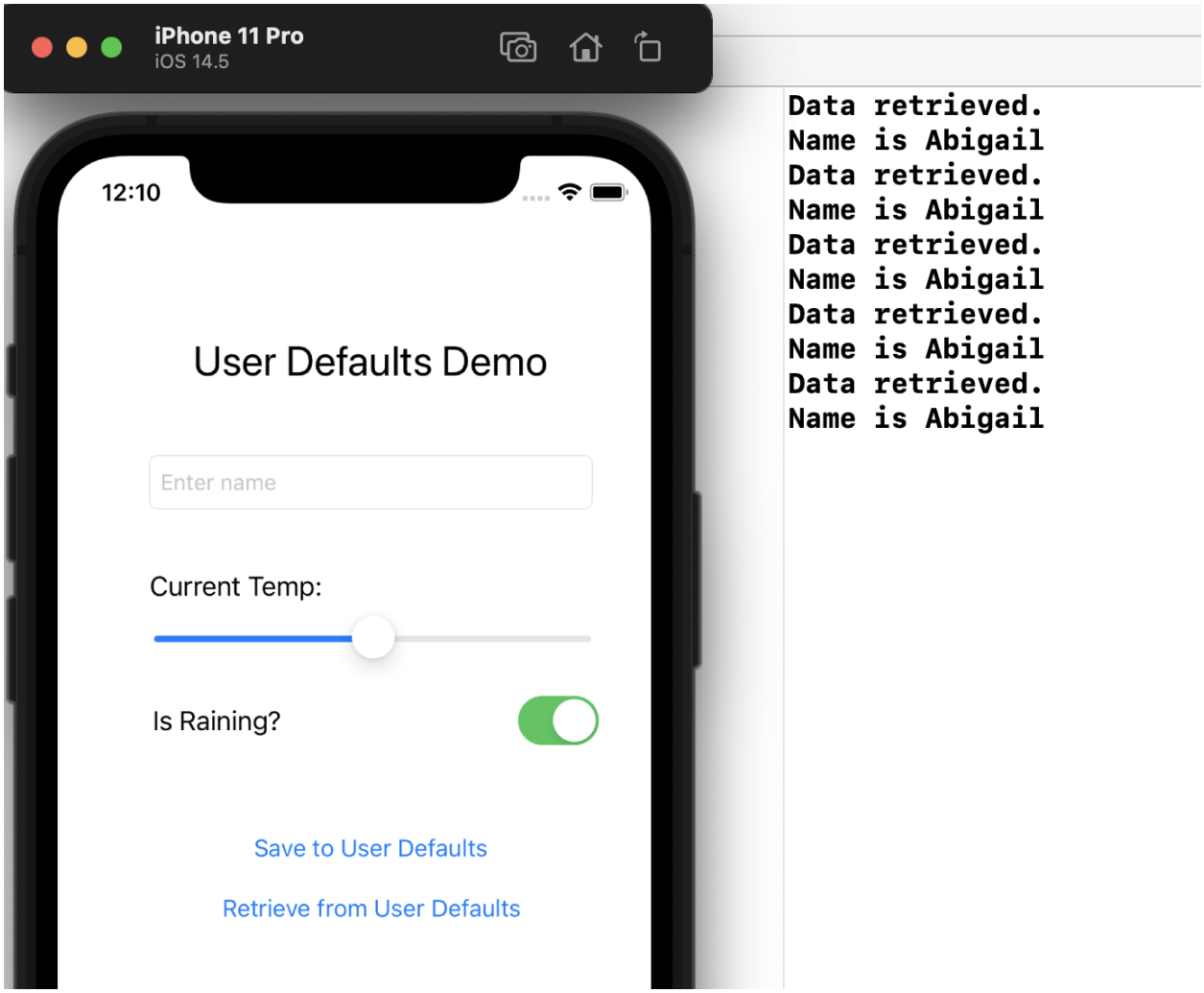
- Save data
- Retrieve data
- Close the app & reopen it, and press the RETRIEVE button, your app remembers the data it saved.

```
import UIKit

class ViewController: UIViewController {

    // OUTLETS:
    @IBOutlet weak var txtName: UITextField!
    @IBOutlet weak var lblCurrTemp: UILabel!
    @IBOutlet weak var sliderTemp: UISlider!
    @IBOutlet weak var switchIsRaining: UISwitch!

    // MARK: USER DEFAULTS
    var defaults:UserDefaults = UserDefaults.standard

    override func viewDidLoad() {
        super.viewDidLoad()
    }

    @IBAction func saveButtonPressed(_ sender: Any) {
        // 1. get person person name
        let name:String = txtName.text!
        print("Saving \(name) to user defaults")
        self.defaults.set(name, forKey:"KEY_NAME") // string
        print("    Save finished.")

        // 2. Clear the UI and prepare for new input
        txtName.text = ""

    }

    @IBAction func retrieveButtonPressed(_ sender: Any) {

//        let result = self.defaults.string(forKey: "KEY_ABC") // nil
//        let result2 = self.defaults.string(forKey: "KEY_NAME") // whatever the person typed in the
textbox
//
        // nil coealscing to deal with the possible nil case
        let nameFromUserDefaults:String = self.defaults.string(forKey: "KEY_NAME") ?? "N/A"

        print("Data retrieved.")
        print("Name is \(nameFromUserDefaults)")
    }
```

```
}
```

Expected result:

- Enter a name and press save to user defaults
- Restart application and press Retrieve from User defaults
- Result will be output to console



Full Example #2 -

- Demonstrates how to save and retrieve and integer & boolean

```
import UIKit

class ViewController: UIViewController {

    // OUTLETS:
    @IBOutlet weak var txtName: UITextField!
    @IBOutlet weak var lblCurrTemp: UILabel!
    @IBOutlet weak var sliderTemp: UISlider!
    @IBOutlet weak var switchIsRaining: UISwitch!
    @IBOutlet weak var tvResults: UITextView!

    // MARK: USER DEFAULTS
    var defaults:UserDefaults = UserDefaults.standard

    override func viewDidLoad() {
        super.viewDidLoad()

        // initialze user defaults
        //self.defaults = UserDefaults.standard
    }

    @IBAction func saveButtonPressed(_ sender: Any) {
        // 1. get person person name
        let name:String = txtName.text!
        print("Saving \(name) to user defaults")

        // 2. Save the data
        self.defaults.set(name, forKey:"KEY_NAME") // string
        self.defaults.set(45, forKey:"KEY_AGE")    // integer
        self.defaults.set(false, forKey:"KEY_HAS_PET") //boolean

        print("   Save finished.")

        // 2. Clear the UI and prepare for new input
        txtName.text = ""
    }

    @IBAction func retrieveButtonPressed(_ sender: Any) {

//        let result = self.defaults.string(forKey: "KEY_ABC") // nil
//        let result2 = self.defaults.string(forKey: "KEY_NAME") // whatever the person typed in the
textbox
//
        // nil coealscing to deal with the possible nil case
        let nameFromUserDefaults:String = self.defaults.string(forKey: "KEY_NAME") ?? "N/A"
        let ageFromUserDefaults = self.defaults.integer(forKey: "KEY_AGE")          // default to 0
        let hasPetFromUserDefaults = self.defaults.bool(forKey: "KEY_HAS_PET")
```

```
        print("Data retrieved.")
        print("Name is \(nameFromUserDefaults)")
        print("Age is \(ageFromUserDefaults)")
        print("Has Pet?  \(hasPetFromUserDefaults)")


        tvResults.text = "Name is \(nameFromUserDefaults) \n Age is \(ageFromUserDefaults) \n
Has Pet?  \(hasPetFromUserDefaults)"
    }
}
```

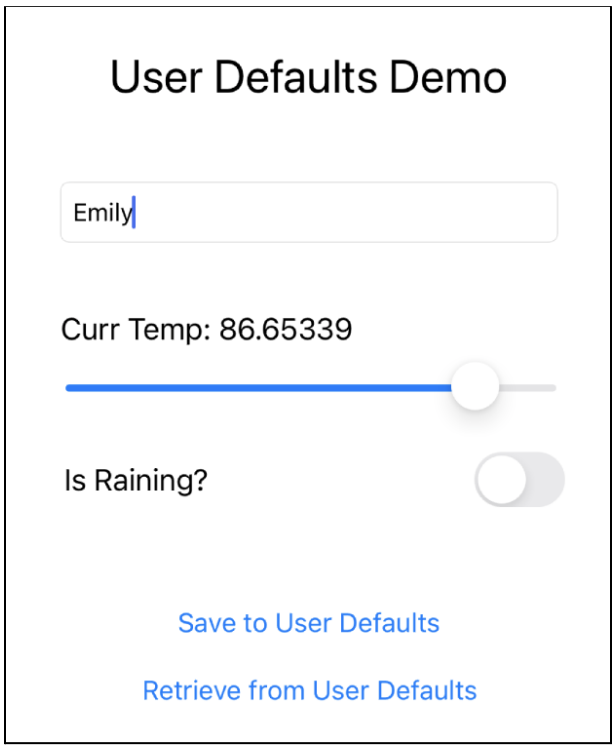
Repopulate the UI based on the values saved in User Defaults

```
import UIKit

class ViewController: UIViewController {

    // OUTLETS:
    @IBOutlet weak var txtName: UITextField!
    @IBOutlet weak var lblCurrTemp: UILabel!
    @IBOutlet weak var sliderTemp: UISlider!
    @IBOutlet weak var switchIsRaining: UISwitch!
    @IBOutlet weak var tvResults: UITextView!

    // MARK: USER DEFAULTS
    var defaults:UserDefaults = UserDefaults.standard

    override func viewDidLoad() {
        super.viewDidLoad()
     }

    @IBAction func saveButtonPressed(_ sender: Any) {
        // 1. get person person name
        let name:String = txtName.text!
        print("Saving \(name) to user defaults")


        // 2. retrieve the value in the slider
        let temp:Float = sliderTemp.value * 100.0

        // 3. retrieve the switch value
        let isRaining:Bool = switchIsRaining.isOn

        // 4. Save the data
        self.defaults.set(name, forKey:"KEY_NAME") // string
        self.defaults.set(45, forKey:"KEY_AGE")    // integer
        self.defaults.set(false, forKey:"KEY_HAS_PET") //boolean

        self.defaults.set(temp, forKey: "KEY_TEMP")   //double (float)
        self.defaults.set(isRaining, forKey:"KEY_IS_RAINING")

        print("   Save finished.")

        // 5. Clear the UI and prepare for new input
        txtName.text = ""
```

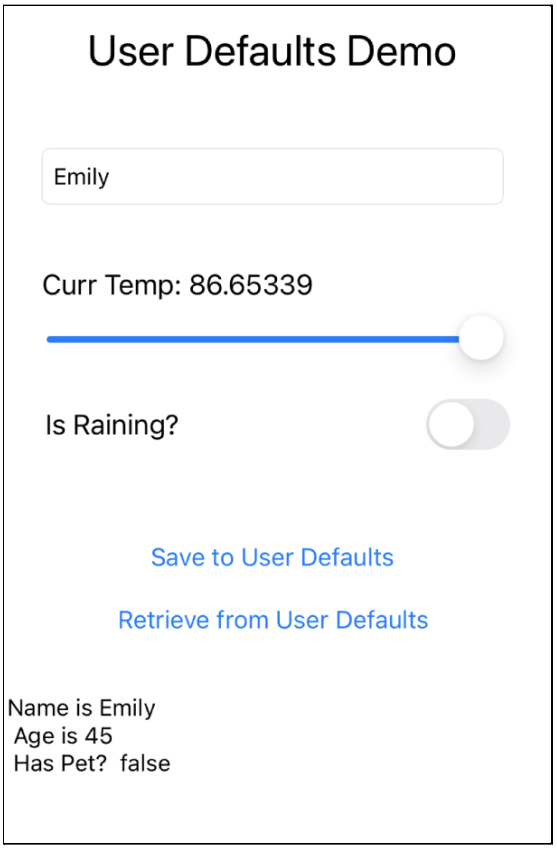
```
        sliderTemp.value = 0.5
        switchIsRaining.isOn = true
        lblCurrTemp.text = "Curr Temp: "

    }

    @IBAction func retrieveButtonPressed(_ sender: Any) {

//        let result = self.defaults.string(forKey: "KEY_ABC") // nil
//        let result2 = self.defaults.string(forKey: "KEY_NAME") // whatever the person typed in the
textbox
//
        // nil coealscing to deal with the possible nil case
        let nameFromUserDefaults:String = self.defaults.string(forKey: "KEY_NAME") ?? "N/A"
        let ageFromUserDefaults = self.defaults.integer(forKey: "KEY_AGE")          // default to 0
        let hasPetFromUserDefaults = self.defaults.bool(forKey: "KEY_HAS_PET")

        print("Data retrieved.")
        print("Name is \(nameFromUserDefaults)")
        print("Age is \(ageFromUserDefaults)")
        print("Has Pet?  \(hasPetFromUserDefaults)")

        // repopulate the UI with the values from user defaults
        txtName.text = nameFromUserDefaults
        switchIsRaining.isOn = self.defaults.bool(forKey: "KEY_IS_RAINING")
        sliderTemp.value = self.defaults.float(forKey:"KEY_TEMP")
        lblCurrTemp.text = "Curr Temp: \(self.defaults.float(forKey:"KEY_TEMP"))"

        tvResults.text = "Name is \(nameFromUserDefaults) \n Age is \(ageFromUserDefaults) \n
Has Pet?  \(hasPetFromUserDefaults)"
    }


    @IBAction func sliderChanged(_ sender: Any) {
        lblCurrTemp.text = "Curr Temp: \(sliderTemp.value * 100)"
    }
}
```

Expected Result

Saved data:



Retrieved Data



Save An Array

- You save an array of strings, array of numbers, array of booleans

Save:

```
 // save an array
let friendsList = ["Emily", "Abigail", "Alex"]
defaults.set(friendsList, forKey:"KEY_FRIENDS_LIST")



// save an array of numbers
let testScores = [100, 50, 75, 90]
defaults.set(testScores, forKey:"KEY_TEST_SCORES_LIST")
```

Retrieval:

```
// cast this generic "Any" object to the specific data type of the array
// - friends = Array of Strings  ---> [String]

let friendsFromUserDefaults:[String] =
        defaults.object(forKey: "KEY_FRIENDS_LIST") as? [String]  ?? []

let testScoresFromUserDefaults:[Int] =
        defaults.object(forKey: "KEY_TEST_SCORES_LIST") as? [Int]  ?? []
```

Full Example:

```
import UIKit

class ViewController: UIViewController {

    // OUTLETS:
    @IBOutlet weak var txtName: UITextField!
    @IBOutlet weak var lblCurrTemp: UILabel!
    @IBOutlet weak var sliderTemp: UISlider!
    @IBOutlet weak var switchIsRaining: UISwitch!
    @IBOutlet weak var tvResults: UITextView!

    // MARK: USER DEFAULTS
    var defaults:UserDefaults = UserDefaults.standard

    override func viewDidLoad() {
        super.viewDidLoad()

        // initialze user defaults
        //self.defaults = UserDefaults.standard
    }

    @IBAction func saveButtonPressed(_ sender: Any) {
        // 1. get person person name
        let name:String = txtName.text!
        print("Saving \(name) to user defaults")


        // 2. retrieve the value in the slider
        let temp:Float = sliderTemp.value * 100.0

        // 3. retrieve the switch value
        let isRaining:Bool = switchIsRaining.isOn

        // 4. Save the data
        self.defaults.set(name, forKey:"KEY_NAME") // string
        self.defaults.set(45, forKey:"KEY_AGE")    // integer
        self.defaults.set(false, forKey:"KEY_HAS_PET") //boolean

        self.defaults.set(temp, forKey: "KEY_TEMP")   //double (float)
        self.defaults.set(isRaining, forKey:"KEY_IS_RAINING")

        // 4b. save an array of data
        let friendsList = ["Emily", "Abigail", "Alex"]
        defaults.set(friendsList, forKey:"KEY_FRIENDS_LIST")

        let testScores = [100, 50, 75, 90]
        defaults.set(testScores, forKey:"KEY_TEST_SCORES_LIST")


        print("   Save finished.")

        // 5. Clear the UI and prepare for new input
```

```
        txtName.text = ""
        sliderTemp.value = 0.5
        switchIsRaining.isOn = true
        lblCurrTemp.text = "Curr Temp: "

    }

    @IBAction func retrieveButtonPressed(_ sender: Any) {

//        let result = self.defaults.string(forKey: "KEY_ABC") // nil
//        let result2 = self.defaults.string(forKey: "KEY_NAME") // whatever the person typed in the textbox
//
        // nil coealscing to deal with the possible nil case
        let nameFromUserDefaults:String = self.defaults.string(forKey: "KEY_NAME") ?? "N/A"
        let ageFromUserDefaults = self.defaults.integer(forKey: "KEY_AGE")          // default to 0
        let hasPetFromUserDefaults = self.defaults.bool(forKey: "KEY_HAS_PET")

        print("Data retrieved.")
        print("Name is \(nameFromUserDefaults)")
        print("Age is \(ageFromUserDefaults)")
        print("Has Pet?  \(hasPetFromUserDefaults)")

        // repopulate the UI with the values from user defaults
        txtName.text = nameFromUserDefaults
        switchIsRaining.isOn = self.defaults.bool(forKey: "KEY_IS_RAINING")
        sliderTemp.value = self.defaults.float(forKey:"KEY_TEMP")
        lblCurrTemp.text = "Curr Temp: \(self.defaults.float(forKey:"KEY_TEMP"))"

        // retrieve the array value
        let friendsFromUserDefaults:[String] =
            defaults.object(forKey: "KEY_FRIENDS_LIST") as? [String]  ?? []
        let testScoresFromUserDefaults:[Int] =
            defaults.object(forKey: "KEY_TEST_SCORES_LIST") as? [Int]  ?? []


        tvResults.text = "Name is \(nameFromUserDefaults) \n Age is \(ageFromUserDefaults) \n Has Pet?
\(hasPetFromUserDefaults)\n"
        tvResults.text = tvResults.text + "Friends: \(friendsFromUserDefaults)\nTest
Scores:\(testScoresFromUserDefaults)"

        print(friendsFromUserDefaults)
        print(friendsFromUserDefaults[0])          // Emily
        print(testScoresFromUserDefaults)
        print(testScoresFromUserDefaults[0] + 20)   // 120
    }


    @IBAction func sliderChanged(_ sender: Any) {
        lblCurrTemp.text = "Curr Temp: \(sliderTemp.value * 100)"
    }
}
```

Expected Result:

- Arrays are retrieved
- Individual data from the array can be accessed and manipulated

```
Data retrieved.
Name is
Age is 45
Has Pet?  false
["Emily", "Abigail", "Alex"]
Emily
[100, 50, 75, 90]
120
```

Save A Dictionary

Save:

```
// save a dictionary -> airports
let airports:[String:String]
        = ["YYZ":"Toronto International Airport", "ORD": "Chicago O'Hare",
"LHR":"London Heathrow"]

defaults.set(airports, forKey: "KEY_AIRPORTS_DICT")
```

Retrieve:

```
// get the dictionary

// option 1
      let airportsFromUserDefaults:[String:String] = defaults.dictionary(forKey:
"KEY_AIRPORTS_DICT")
          as? [String:String] ?? [:]

      // option 2
      let airportsFromUserDefaults2:[String:String] = defaults.object(forKey:
"KEY_AIRPORTS_DICT")
          as? [String:String] ?? [:]
```

Save a Struct to the UserDefaults

UserDefaults can save a struct, provided that the struct conforms to the Codable protocol.

0/ Define the struct

1/ Make the struct conform to the Codable protocol

2/ Using JSONEncoder, convert the struct to a JSON object

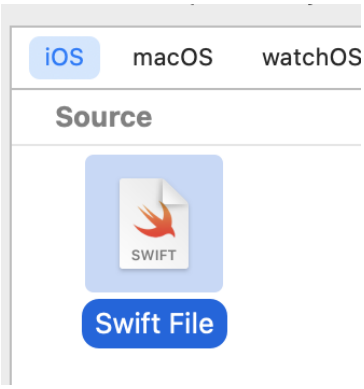
3/ Retrieve the struct using .object(forKey:), then convert it to a struct using JSONDecoder

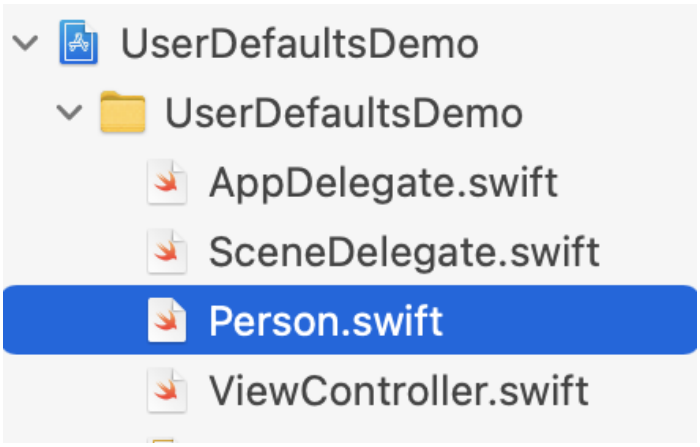
Examples:
https://www.hackingwithswift.com/example-code/system/how-to-load-and-save-a-struct-in-userdefaults-using-codable

Code Along - Structs and User Defaults

0/ Define the struct

Create a file called Person.struct





Define the properties of the struct

```
struct Person {
    var name:String
    var age:Int
    var hasPet:Bool

    // struct provides a implicit initializer for all the properties
    // so therefore, we dont need to define an explicit initializer
}
```

1/ Make the struct conform to the **Codable** protocol

```
// Ensure struct conforms with Codable protocol
struct Person:Codable {
    var name:String
    var age:Int
    var hasPet:Bool

    // struct provides a implicit initializer for all the properties
    // so therefore, we dont need to define an explicit initializer
}
```

2/ Using **JSONEncoder**, convert the struct to a JSON object

- To save, convert the struct to a JSON object

```
// 4d. save a Person struct
let personToSave = Person(name: "Zayne", age: 55, hasPet: true)

// convert the struct to a JSON object
let encoder = JSONEncoder()
if let personAsJSONObj = try? encoder.encode(personToSave) {
    // save the JSON object to user defaults
    defaults.set(personAsJSONObj, forKey:"KEY_PERSON")
}
else {
    print("Could not encode the person as a JSON object")
}
```

3/ Retrieve the struct using .object(forKey:), then convert it to a struct using JSONDecoder

- To retrieve: get the JSON object out of the user defaults
- Convert (cast) that json object BACK to the struct

```
// retrieve the person object
        if let personFromUserDefaults = self.defaults.object(forKey: "KEY_PERSON") as?
Data {
            // convert the data to a Person struct
            let decoder = JSONDecoder()
            if let person = try? decoder.decode(Person.self, from: personFromUserDefaults)
{
                print("Person conversion success")
                print("Person properties are: ")
                print(person.name)
                print(person.age)
                print(person.hasPet)

                person.sayHello()
            }
            else {
                print("Could not convert the data back to a Person struct")
            }
        }
        else {
            print("Retrieval failed or casting to type Data failed")
        }
```

```
Data retrieved.
Name is
Age is 45
Has Pet?  false
Person conversion success
Person properties are:
Zayne
55
true
Zayne is 55 years old and says HELLO to you!
```

Code: Completed functions for saving structs

Person.swift:

```
// Ensure struct conforms with Codable protocol
struct Person:Codable {
    var name:String
    var age:Int
    var hasPet:Bool

    // struct provides a implicit initializer for all the properties
    // so therefore, we dont need to define an explicit initializer

    func sayHello() {
        print("\(name) is \(age) years old and says HELLO to you!")
    }
}
```

ViewController.swift:

```
import UIKit

class ViewController: UIViewController {

    // OUTLETS:
    @IBOutlet weak var txtName: UITextField!
    @IBOutlet weak var lblCurrTemp: UILabel!
    @IBOutlet weak var sliderTemp: UISlider!
    @IBOutlet weak var switchIsRaining: UISwitch!
    @IBOutlet weak var tvResults: UITextView!

    // MARK: USER DEFAULTS
    var defaults:UserDefaults = UserDefaults.standard

    override func viewDidLoad() {
        super.viewDidLoad()

        // initialze user defaults
        //self.defaults = UserDefaults.standard
    }

    @IBAction func saveButtonPressed(_ sender: Any) {
        // 1. get person person name
        let name:String = txtName.text!
        print("Saving \(name) to user defaults")


        // 2. retrieve the value in the slider
        let temp:Float = sliderTemp.value * 100.0

        // 3. retrieve the switch value
        let isRaining:Bool = switchIsRaining.isOn

        // 4. Save the data
        self.defaults.set(name, forKey:"KEY_NAME") // string
        self.defaults.set(45, forKey:"KEY_AGE")    // integer
        self.defaults.set(false, forKey:"KEY_HAS_PET") //boolean

        self.defaults.set(temp, forKey: "KEY_TEMP")   //double (float)
        self.defaults.set(isRaining, forKey:"KEY_IS_RAINING")

        // 4b. save an array of data
        let friendsList = ["Emily", "Abigail", "Alex"]
        defaults.set(friendsList, forKey:"KEY_FRIENDS_LIST")

        let testScores = [100, 50, 75, 90]
        defaults.set(testScores, forKey:"KEY_TEST_SCORES_LIST")


        // 4c. save a dictionary
        let airports:[String:String] = ["YYZ":"Toronto International Airport", "ORD": "Chicago O'Hare",
"LHR":"London Heathrow"]
```

```
        defaults.set(airports, forKey: "KEY_AIRPORTS_DICT")


        // 4d. save a Person struct
        let personToSave = Person(name: "Zayne", age: 55, hasPet: true)

        // convert the struct to a JSON object
        let encoder = JSONEncoder()
        if let personAsJSONObj = try? encoder.encode(personToSave) {
            // save the JSON object to user defaults
            defaults.set(personAsJSONObj, forKey:"KEY_PERSON")
        }
        else {
            print("Could not encode the person as a JSON object")
        }

        print("   Save finished.")

        // 5. Clear the UI and prepare for new input
        txtName.text = ""
        sliderTemp.value = 0.5
        switchIsRaining.isOn = true
        lblCurrTemp.text = "Curr Temp: "

    }

    @IBAction func retrieveButtonPressed(_ sender: Any) {

//        let result = self.defaults.string(forKey: "KEY_ABC") // nil
//        let result2 = self.defaults.string(forKey: "KEY_NAME") // whatever the person typed in the textbox
//
        // nil coealscing to deal with the possible nil case
        let nameFromUserDefaults:String = self.defaults.string(forKey: "KEY_NAME") ?? "N/A"
        let ageFromUserDefaults = self.defaults.integer(forKey: "KEY_AGE")          // default to 0
        let hasPetFromUserDefaults = self.defaults.bool(forKey: "KEY_HAS_PET")

        print("Data retrieved.")
        print("Name is \(nameFromUserDefaults)")
        print("Age is \(ageFromUserDefaults)")
        print("Has Pet?  \(hasPetFromUserDefaults)")

        // repopulate the UI with the values from user defaults
        txtName.text = nameFromUserDefaults
        switchIsRaining.isOn = self.defaults.bool(forKey: "KEY_IS_RAINING")
        sliderTemp.value = self.defaults.float(forKey:"KEY_TEMP")
        lblCurrTemp.text = "Curr Temp: \(self.defaults.float(forKey:"KEY_TEMP"))"

        // retrieve the array value
        let friendsFromUserDefaults:[String] =
            defaults.object(forKey: "KEY_FRIENDS_LIST") as? [String]  ?? []
        let testScoresFromUserDefaults:[Int] =
            defaults.object(forKey: "KEY_TEST_SCORES_LIST") as? [Int]  ?? []

        // retrieve the dictionray value
        // option 1
        let airportsFromUserDefaults:[String:String] = defaults.dictionary(forKey: "KEY_AIRPORTS_DICT")
```

```
            as? [String:String] ?? [:]

        // option 2
        let airportsFromUserDefaults2:[String:String] = defaults.object(forKey: "KEY_AIRPORTS_DICT")
            as? [String:String] ?? [:]

        tvResults.text = "Name is \(nameFromUserDefaults) \n Age is \(ageFromUserDefaults) \n Has Pet?
\(hasPetFromUserDefaults)\n"
        tvResults.text = tvResults.text + "Friends: \(friendsFromUserDefaults)\nTest
Scores:\(testScoresFromUserDefaults)"

        print(friendsFromUserDefaults)
        print(friendsFromUserDefaults[0])          // Emily
        print(testScoresFromUserDefaults)
        print(testScoresFromUserDefaults[0] + 20)   // 120

        // demoing option 1 & option 2
        print(airportsFromUserDefaults["LHR"])
        print(airportsFromUserDefaults2["LHR"])

        // retrieve the person object
        if let personFromUserDefaults = self.defaults.object(forKey: "KEY_PERSON") as? Data {
            // convert the data to a Person struct
            let decoder = JSONDecoder()
            if let person = try? decoder.decode(Person.self, from: personFromUserDefaults) {
                print("Person conversion success")
                print("Person properties are: ")
                print(person.name)
                print(person.age)
                print(person.hasPet)

                person.sayHello()
            }
            else {
                print("Could not convert the data back to a Person struct")
            }
        }
        else {
            print("Retrieval failed or casting to type Data failed")
        }
        }


    @IBAction func sliderChanged(_ sender: Any) {
        lblCurrTemp.text = "Curr Temp: \(sliderTemp.value * 100)"
    }
}
```

User Defaults are Shared Between Screens

1/ Each screen creates its own instance of UserDefaults
2/ Use the screen's UserDefaults variable to store or retrieve values
3/ All screens store and retrieve from the **same** UserDefaults

Code Example: Sharing Data between Screens using UserDefaults

Screen2ViewController.swift

Screen1ViewController.swift

Return values for non-existent keys

From HackingWithSwift.com:

- `integer(forKey:)` returns an integer if the key existed, or 0 if not.
- `bool(forKey:)` returns a boolean if the key existed, or false if not.
- `float(forKey:)` returns a float if the key existed, or 0.0 if not.
- `double(forKey:)` returns a double if the key existed, or 0.0 if not.
- `object(forKey:)` returns `Any?` so you need to conditionally typecast it to your data type.

Examples:

Strings return nil

```
let a = self.defaults.string(forKey:"licensePlate")
```

Numbers return 0

```
let b = self.defaults.integer(forKey: "salary") // get a int out of user defaults
```

Booleans return false

```
let c = self.defaults.bool(forKey:"likesCoconut")
```