1.0 Lesson - iPhone App Data Basics

Paul Solt - Paul@SuperEasyApps.com
SuperEasyApps.com

- 1.0 Lesson - iPhone App Data Basics

1.1 Lecture - iPhone App Data Basics

You will learn more about how data can be used inside your apps in this lesson. Data can come from distant servers, local databases inside your app, or from user input.

1.2 Tutorial - Swift Dictionary

The `Dictionary` type is a collection class that provides storage for related data using key and value pairs. Much like an English dictionary, each word in the dictionary is associated with a meaning, synonyms, and supportive text on using the word.

```
// Swift Dictionary [key : value]

let place = [ "city" : "Rochester",
    "state" : "NY",
    "population" : 210000]

place["city"]
print("place[city] = \(place["city"])")
place["state"]
place["population"]

place["best food"] // nil
```

Links

- Collection Types - Dictionary

1.3 Tutorial - Swift Array

The Swift `Array` collection allows you to store multiple objects or values. It is a building block combined with the Dictionary type for storing multiple pieces of information.

```
// Array [value, value, value]

let favoriteFood = ["pizza", "thai", "tacos", "vegetables", "smoothies"]

favoriteFood.first
favoriteFood.last

// 0, 1, 2, ...

favoriteFood[0]
favoriteFood[1]
favoriteFood[2]
favoriteFood[3]
//favoriteFood[4] // error if 4 elements in array!

// Check the number of elements before trying to access
//  the 5th element (0-based counting)
if favoriteFood.count > 4 {
    favoriteFood[4]
}
```

Links

- Collection Types - Array

1.4 Tutorial - JSON and Web Services

1. JSON stands for Javascript Object Notation
    1. http://www.json.org
2. JSON is common web data format with server APIs.
    1. The top 5 paid apps on iTunes:
    2. https://itunes.apple.com/us/rss/toppaidapplications/limit=5/json
3. JSON viewer
    1. http://jsonviewer.stack.hu

1.5 Tutorial - Loading JSON Files and Error Handling

Given a `sample.json` file you can load the data into your iPhone app or a Playground.

```
do {
    // Simple json file loading
    let fileURL = NSBundle.mainBundle().URLForResource("sample", withExtension: "json")
    let jsonData = NSData(contentsOfURL: fileURL!)
    let json = try NSJSONSerialization.JSONObjectWithData(jsonData!, options: [])

    print(json) // display the data AnyObject (NSDictionary or NSArray)

} catch let error as NSError {
    print("Error loading JSON data:", error.localizedDescription)
}
```

Links

- NSJSONSerialization - apple.com
- NSBundle URLForResource - apple.com
- Error Handling in Swift

1.6 Parsing JSON Data With Swift

As you work with web data, you will have to write code that is more robust that can handle situations when data is missing. Optional types allow you to capture information when it exists as well as when it doesn't.

```
// Convert the data to a Dictionary [String : AnyObject]

if let personData = json as? [String : AnyObject] { // Swift Dictionary

    personData["FirstName"]
    personData["LastName"]
    personData["lastName"] // nil
    personData["FavoriteFoods"]

    if let foodArray = personData["FavoriteFoods"] as? [String] {

        // loop
        for item in foodArray {
            print(item)
```

```
        }

    } else {
        print("Invalid food array")
    }
} else {
    print("Invalid Swift dictionary")
}
```

1.7 Quiz - Data and JSON

Take the quiz:

- 1.7 Quiz - Data and JSON