Collections & App Delegates

PROG31975 – Week 3 Part 2

Jawaad Sheikh

Jawaad.Sheikh@SheridanCollege.ca

Outline

- Introduction
- Arrays
- Dictionaries
- Sets
- Tuples
- The AppDelegate
- Exercise

Introduction

- Collections are used to organize data.
- We are used to using arrays and dictionaries and in Swift this is no different.
- In Objective-C, Collections were not as simple implement. Especially with numbers.

Introduction

- In Swift and Objective-C, the languages make a big distinction between something that is mutable or immutable:
  - Mutable = changeable (using var or NSMutableArray)
  - Immutable = read only (using let or NSArray)

Arrays

- Arrays are an ordered list of items.
- Typically used for numbers but can be used for objects.
- Three ways to define an array:
  - var array1: Array<String>
  - var array2: [String]
  - var array3 = ["Apple", "Pear", "Orange"]

Arrays

- Immutable (read-only) version
  - let array4 = ["Apple", "Pear", "Orange"]

Arrays

- Adding to the array **(defined using var):**
  - array3.append(“Bananas”)
- Deleting:
  - array3.remove(at: 1) // deletes Pear

Arrays

- Sorting and printing:

```
var listOfNumbers = [3, 2, 100, 10, 1]

listOfNumbers.sort(by: <)

print(listOfNumbers)
// [1, 2, 3, 10, 100]

listOfNumbers.sort(by: >)

print(listOfNumbers)
// [100, 10, 3, 2, 1]
```

Dictionaries

- Work on a key-value pair approach.
- Allow you to give a label to an index.
- Unordered
- Three ways to define a dictionary:
  - var dictionary1: Dictionary<String, Int>
  - var dictionary2: [String: Int]
  - var dictionary3 = ["Apple": 3, "Pear": 8, "Orange": 11]

Dictionaries

- Immutable (read only) version:
  - let dictionary4 = ["Apple": 3, "Pear": 8, "Orange": 11]

Dictionaries

- Deleting an object:
  - dictionary4.removeValue(forKey: 0)
- Another example:

```
var dictionary = [String: Int]()

dictionary["Oranges"] = 2
dictionary["Apples"] = 10
dictionary["Pears"] = 5

dictionary = [:]
```

Sets

- Similar to arrays but unordered.
- Items can only appear once in a set.
- Lightweight.
- Three ways to define a set:
  - var set1: Set<String>
  - var set2 = Set<String>()
  - var set3: Set<String> = ["Apple", "Pear", "Orange"]

Sets

- Operations:
  - set3.count
  - set3.insert(“Prune”)
  - set3.remove(“Orange”)
  - set3.contains(“Apple”)

Tuples

- Tuples are not a collection.
- They are a pairing of two items not necessarily of the same data type.
- Used conveniently in loops to loop through 2 items at the same time.

Tuples

- Examples:
  - var currency = ("EUR", 0.81)
  - var time = (Date(), "This is my message.")
  - var email = ("Bart Jacobs", "bart@example.com")
- Accessing values:
  - var rate = currency.1
  - var message = time.1
  - var name = email.0

Tuples

- Example using a for loop:

```
let intNums = [ “Prime” : [2,3,5,7,11, 13],
        “Fibonacci” : [1, 1, 2, 3, 5, 8],
        “Square” : [1, 4, 9, 16, 25] ]
var largest = 0
for (kind, numbers) in intNums{
        for number in numbers {
                if number > largest
                        largest = number
        } } }
```

The AppDelegate

- Every app has a file called AppDelegate.swift
- It serves as the manager of the lifecycle for the app.
- It receives messages about certain states of the app.
- Such as when it launches, gets minimized and so on.

The AppDelegate

- It is a singleton class.
  - This means that it is instantiated once and only once.
  - This can be useful for sharing data between objects.

The AppDelegate

- One method stands out from the rest:
  - didFinishLaunching() – similar concept to viewDidLoad() in a ViewController except it applies to the entire app.
    - Launches at the start of the app launch (don’t mix this up with an app already open and you are bringing it into the forground)

The AppDelegate

- Accessing the AppDelegate
  - you can’t instantiate the AppDelegate, that is done internally.
  - Instead, you request access to it.

```
let mainDelegate = UIApplication.shared.delegate as! AppDelegate
```

Exercise

- Create an iPhone app that will have a home page and a sub page.
- In the app delegate define
  - An array to hold options to play.
  - A dictionary to hold game information (scores, time left, high score, level)
  - A string to hold the title of the app
- On the home page display the title and options.
- On the sub page display the title and the game information.