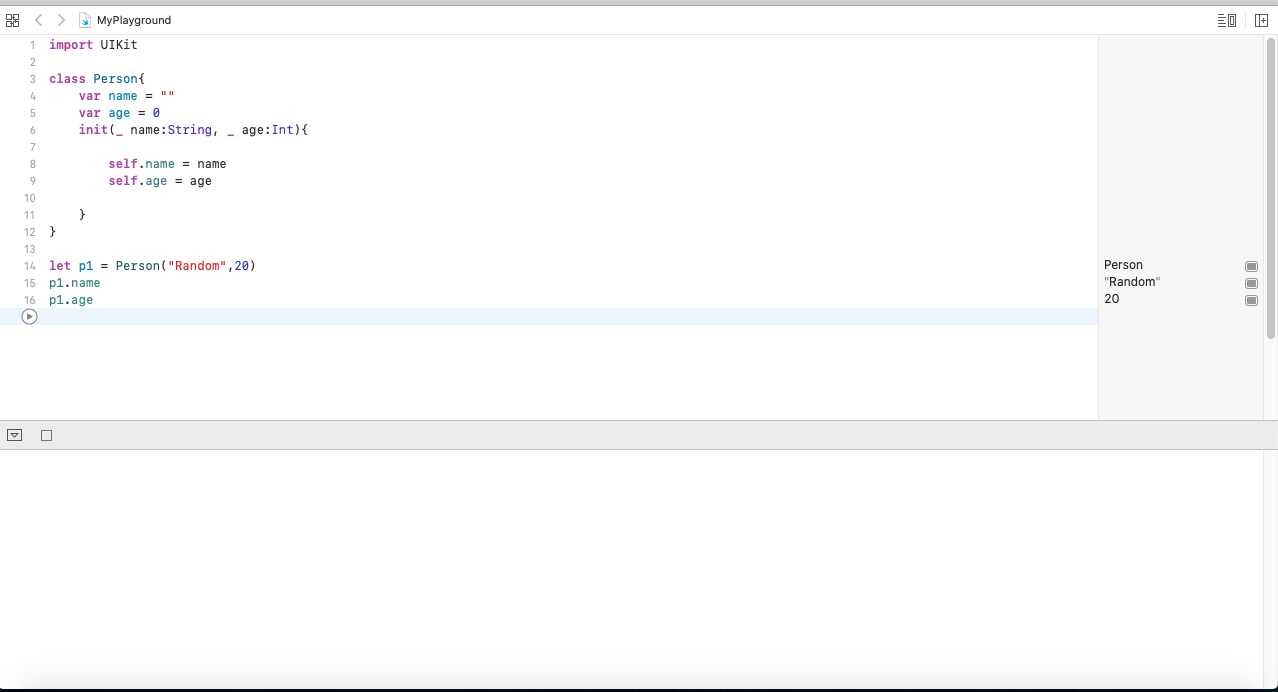
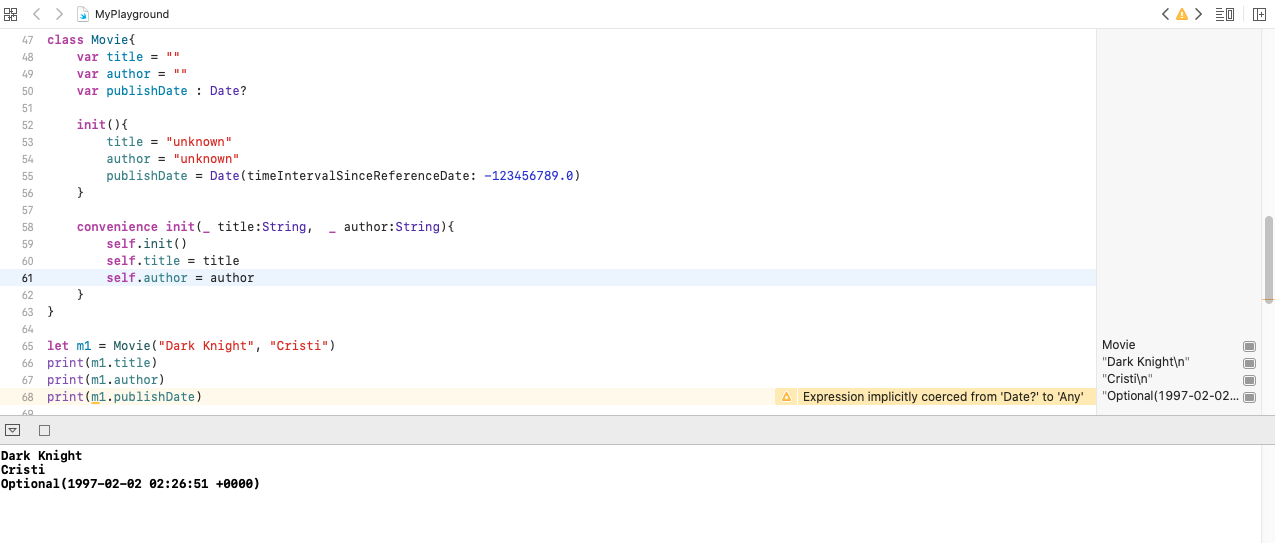
1. **Implement the parameterised initialisation with class or struct.**

****

1. **Write all the Rules of initialiser in Inheritance**

* **Rule 1:** A designated initializer must call a designated initializer from its immediate superclass.
* **Rule 2:** A convenience initializer must call another initializer from the *same* class.
* **Rule 3:** A convenience initializer must ultimately call a designated initializer

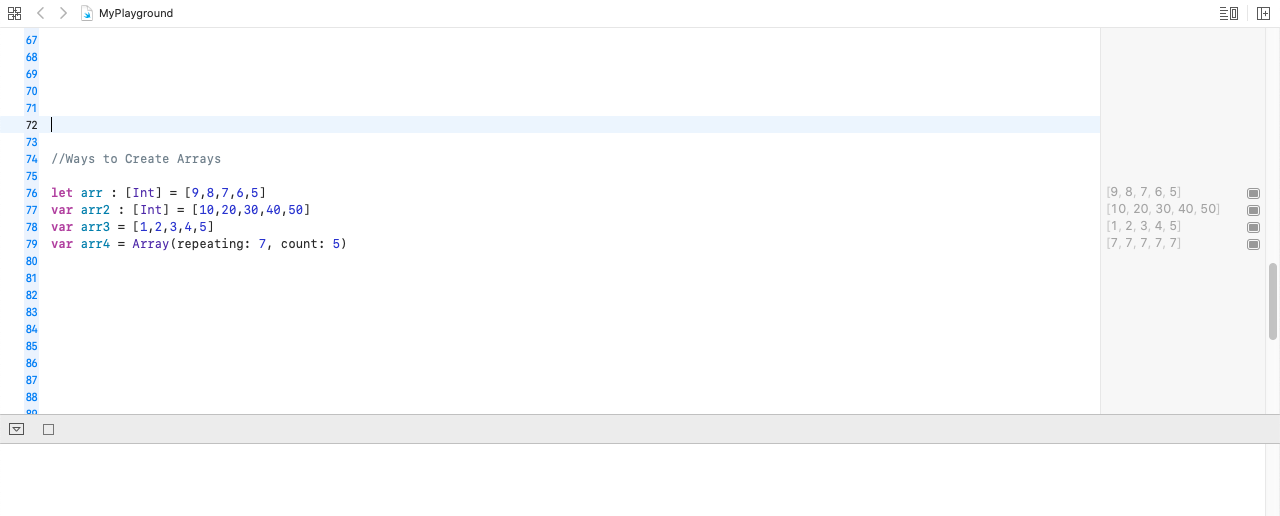
1. **Using convenience Initializers, write-down the Initializers for MOVIE class having basic attributes like title, author, publish\_date, etc.**



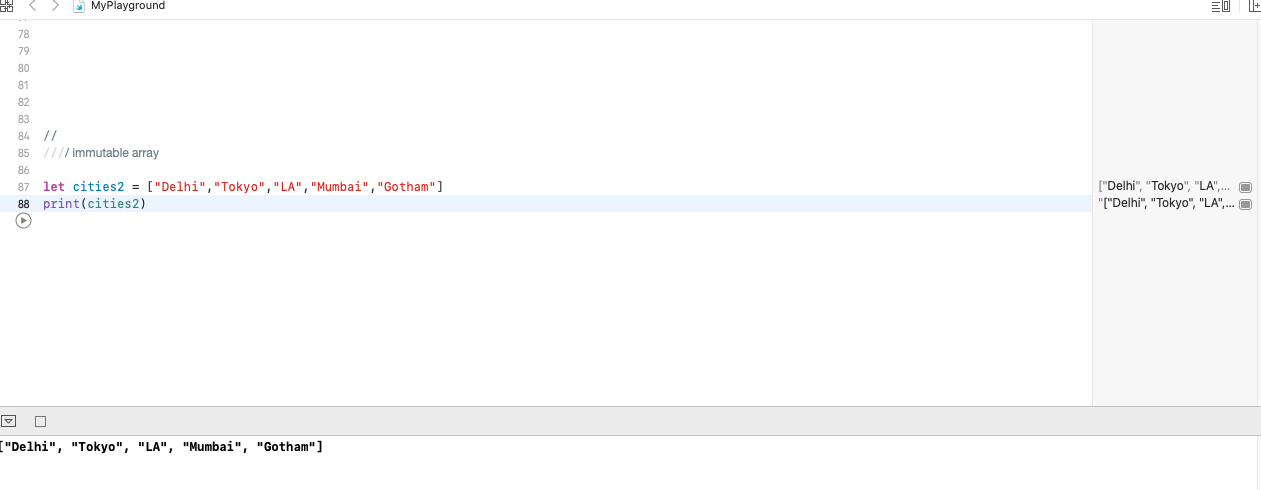
1. **Declare a structure which can demonstrate the throwable Initializer**

**Array**

1. **Create an array containing the 5 different integer values. Write are at least 4 ways to do this.**

****

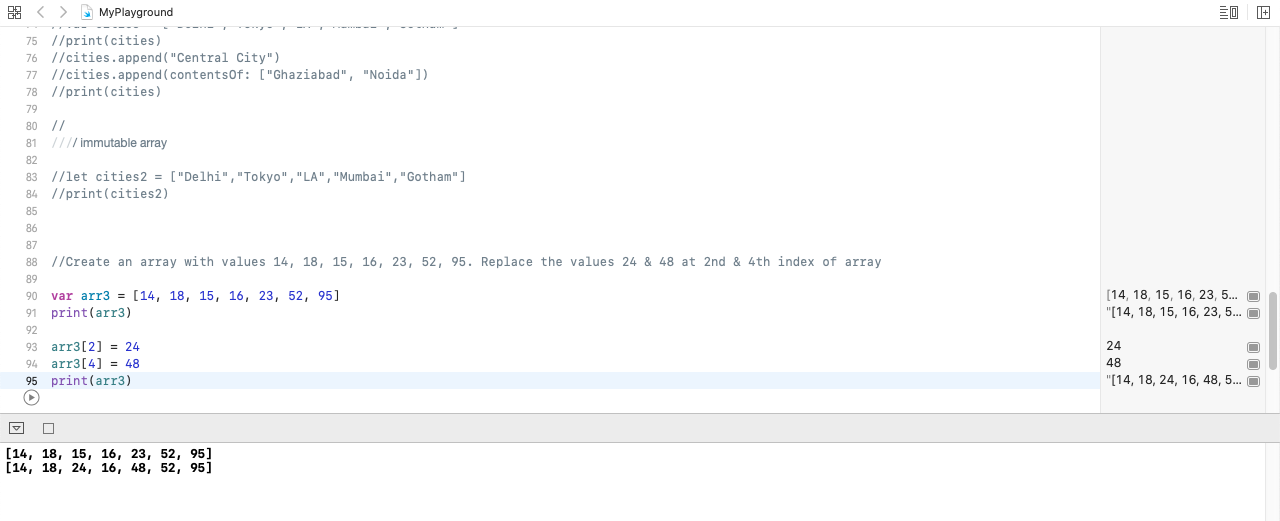
1. **Create an immutable array containing 5 city names.**

****

1. **Create an array with city 5 city names. Later add other names like Canada, Switzerland, Spain to the end of the array in at least 2 possible ways.**

****

1. **Create an array with values 14, 18, 15, 16, 23, 52, 95. Replace the values 24 & 48 at 2nd & 4th index of array**

****

**Set**

1. Given the following sets:

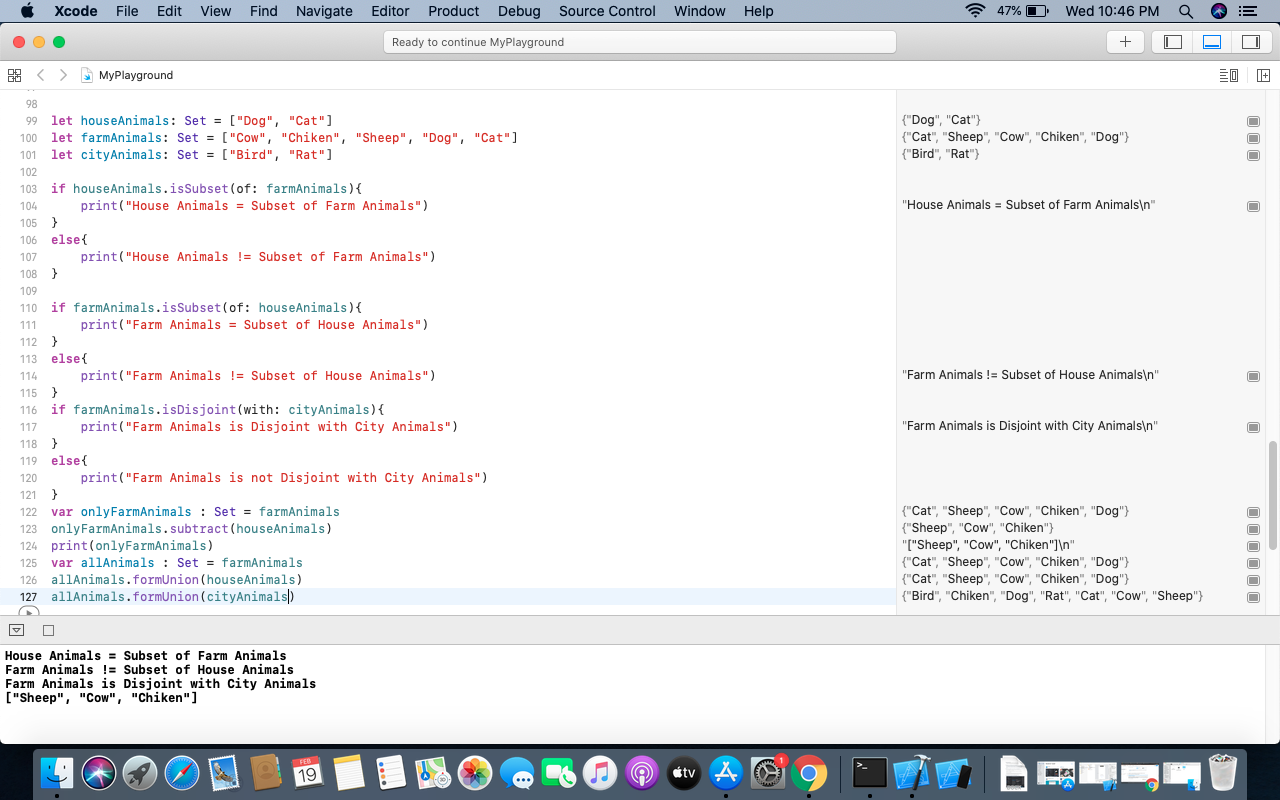
let houseAnimals: Set = ["🐶", "🐱"]

let farmAnimals: Set = ["🐮", "🐔", "🐑", "🐶", "🐱"]

let cityAnimals: Set = ["🐦", "🐭"]

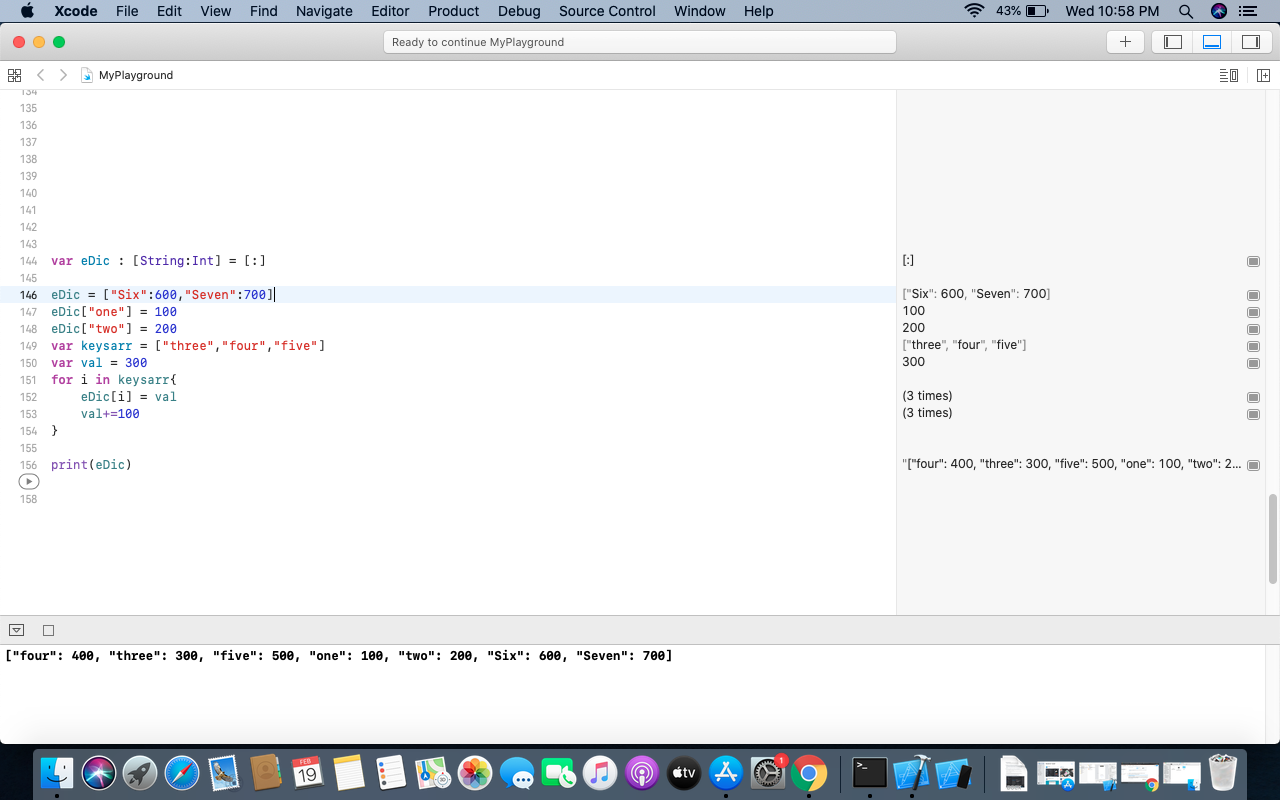
**Use set operations to...**

1. Determine whether the set of house animals is a subset of farm animals.
2. Determine whether the set of farm animals is a superset of house animals.
3. Determine if the set of farm animals is disjoint with city animals.
4. Create a set that only contains farm animals that are not also house animals.
5. Create a set that contains all the animals from all sets.

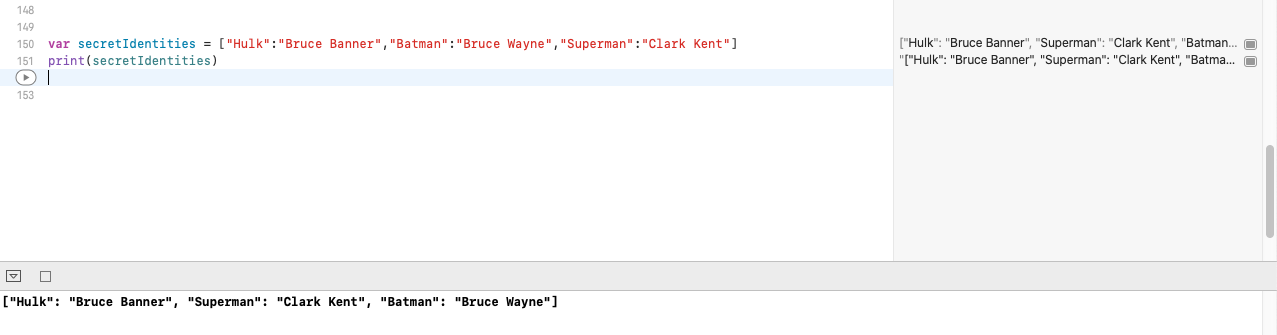


**Dictionary**

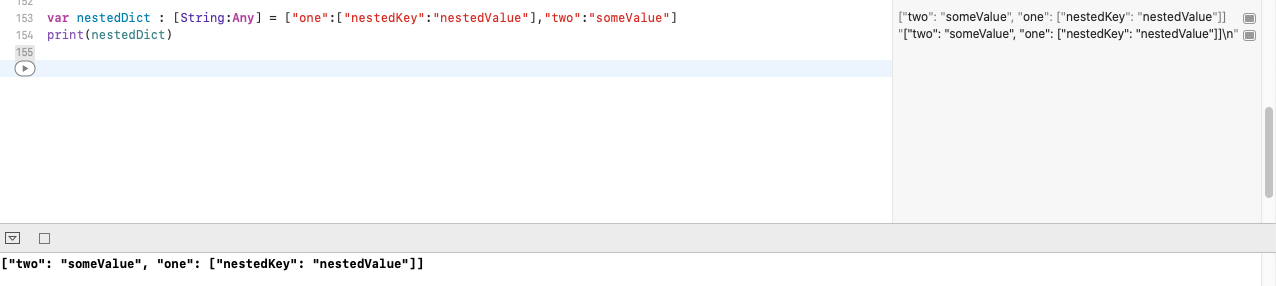
1. **Create an empty dictionary with keys of type String and values of type Int and assign it to a variable in as many ways as you can think of (there's at least 4 ways).**

****

1. **Create a mutable dictionary named secretIdentities where the key value pairs are "Hulk" -> "Bruce Banner", "Batman" -> "Bruce Wayne", and "Superman" -> "Clark Kent".**

****

1. **Create a nesters structure of Key-value pair.**

****

1. **Print all the keys in the dic**

****

**Subscript :**

1. **What is subscript ? Write down the declaration syntax.**

Subscripts are used to access information from a collection, sequence and a list in Classes, Structures and Enumerations without using a method. These subscripts are used to store and retrieve the values with the help of index without the use of separate method. To access elements via subscripts write one or more values between square brackets after the instance name.

**Syntax:**

**subscript(index: Int) −> Int {**

**get {**

**// used for subscript value declarations**

**}**

**set(newValue) {**

**// definitions are written here**

**}**

**}**

**2. Create a simple subscript that outputs true if a string contains a substring and false otherwise.**

**extension String{**

**subscript(subString : String) -> Bool {**

**get{**

**let parentString = self.range(of: subString)**

**return !parentString!.isEmpty**

**}**

**}**

**}**

**var someString = "This is Something"**

**print(someString["Something"])**