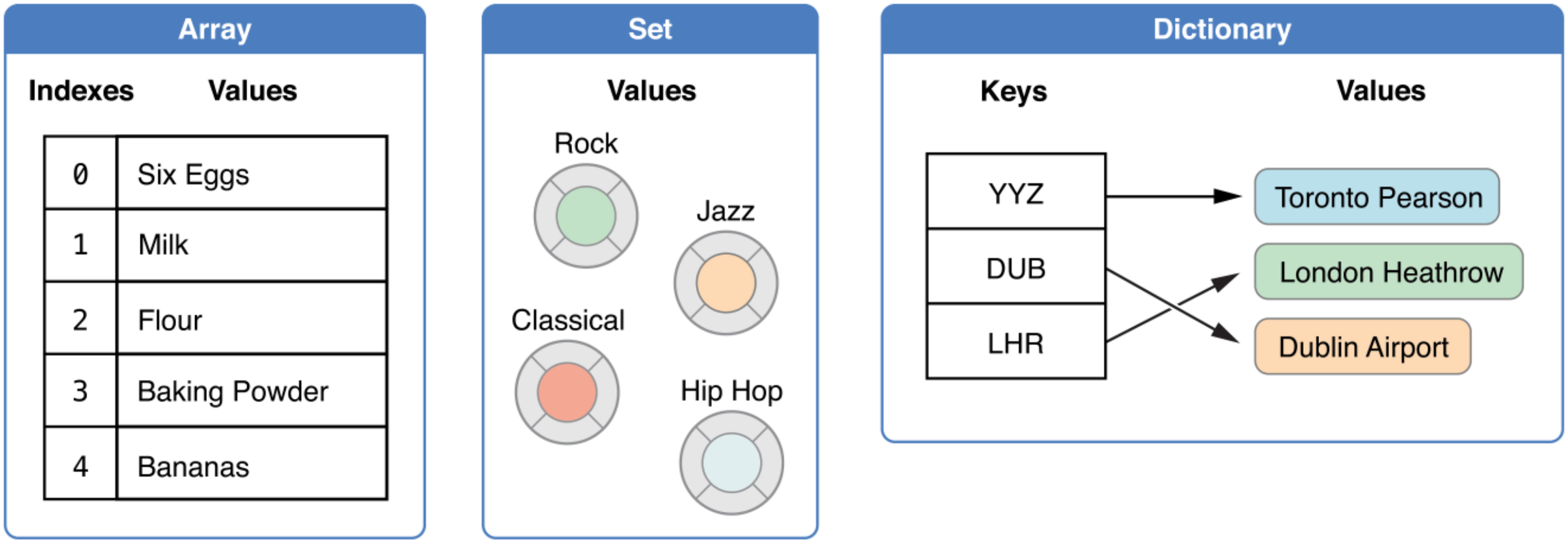
Collection Types

Arrays, Sets and Dictionaries

-Devendranath

Swift provides three kinds of collections like

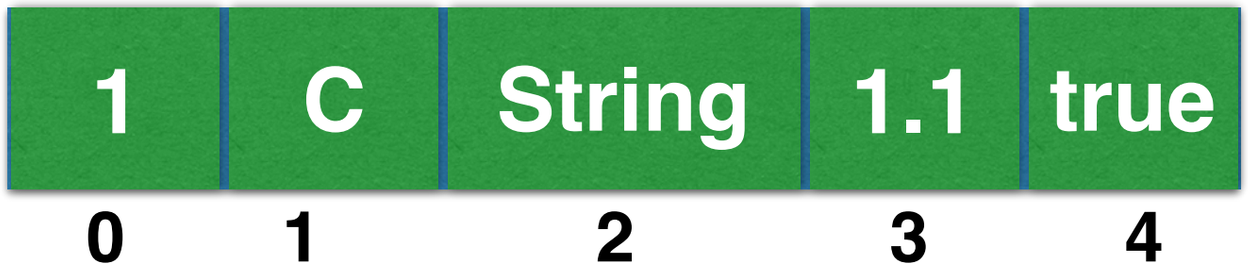
- **Array (Collection of similar or dissimilar elements)**
- **Dictionary (Collection of Key-Value pairs)**
- **Set (Collection of unique values)**



Set is a collection of Similar datatype values without duplicate values.

Array

let aArray = [1 , “C”, “String”, 1.1, true]



Array is collection of Similar / Dissimilar datatype values stored as Ordered collection.

Access values using index which starts from 0 to N-1.

Array Operations

- Creating Arrays
- Counting number of elements in an array
- Finding the object at given Index
- Comparing two arrays
- Iterating through Array Elements
- Adding elements to an array
- Inserting an Element at given index
- Adding two arrays
- Check object existence in an array
- Replace object at given index
- Accessing last object of the array
- Remove Object at given index
- Remove Objects in Range
- Remove all Objects
- Reversing an Array
- Finding the index of Element

```
// Creating Arrays
// Declaring an empty array which stores all Integer
values
   var anIntArray: [Int]
   var anIntArray = Array<Int>();
   var anIntArray = [Int]()

// Defining Arrays with similar type elements
    anIntArray = [1,2,3,1,2,3]
    anIntArray = [1,2,3,"A","B","C"] // ERROR: Can not
 store Non-Integer values integer array [Int]

// Defining arrays which stores different kind of
values

var multitypeArray:[Any] = [1,2,3,"One","Two","Three"]

// Accessing Array Elements
print(anIntArray[3]) // One
```

```swift
// Comparing two arrays
// NOTE: The arrays must be of same type (i.e: [Int], [Int])
if anIntArray == multitypeArray // compiler Error
{
    print("Two array contents are same")
}

let anotherIntArray = [1,2,3,1,2,3]

if anIntArray == anotherIntArray
{
    print("Two array elements are equal")
}
// O/P: Two array elements are equal

// Finding number of elements in an Array
print("Number of elements: \(anIntArray.count)") // 6
```

```swift
// Iterating through Array
for item in multitypeArray
{
    print(item)
}
// O/P: 1, 2, 3, One, Two, Three

// Adding an Element to an Array
multitypeArray.append("Four")
multitypeArray.insert(4, at: 3)
print(multitypeArray) // [1, 2, 3, 4, "One", "Two",
"Three", "Four"]

// Adding another array elements
anIntArray.append(contentsOf: anotherIntArray)
print(anIntArray)
// [1, 2, 3, 1, 2, 3, 1, 2, 3, 1, 2, 3]

// Adding elements of an array
anIntArray += anotherIntArray
print(anIntArray) // [1, 2, 3, 1, 2, 3, 1, 2, 3, 1, 2,
3, 1, 2, 3, 1, 2, 3]
```

```
print(multitypeArray) //[1, 2, 3, 4, "One", "Two", "Three",
“Four"]

// Replacing objects in range
multitypeArray.replaceSubrange(4...7, with: ["1", "2", "3", "4"])
print(multitypeArray) // [1, 2, 3, 4, "1", "2", "3", "4"]

// Removing array element at given index
multitypeArray.remove(at: 3)
print(multitypeArray) // [1, 2, 3, "1", "2", "3", "4"]

// Removing last n elements
multitypeArray.removeLast(3)
print(multitypeArray) // [1, 2, 3, "1"]

// Removing 1st element
multitypeArray.removeFirst()
print(multitypeArray) // [2, 3, "1"]

// Removing 1st n elements
multitypeArray.removeFirst(2)
print(multitypeArray) // ["1"]

// Removing all elements
multitypeArray.removeAll() // []
```

Dictionary

Unordered collection of Key-Value pairs

```swift
// Creating Dictionaries
var stringIntDict = ["One":1, "Two":2, "Three" : 3]
// Dictionaries with different kind of values

var multiValueDict: [String : Any] = ["One": 1, "OneAndHalf" :
1.50, "true" : true]

//Accessing Dictionary values
print(stringIntDict["One"]) // 1
print(stringIntDict["Ten"] ?? 0) //0

// Adding a pair to the Dictionary
stringIntDict["Four"] = 4
print(stringIntDict) //["Three": 3, "Two": 2, "Four": 4,
"One": 1]

// Updating Value for the given key
stringIntDict["Four"] = 44
print(stringIntDict) // ["Three": 3, "Two": 2, "Four": 44,
"One": 1]
stringIntDict.updateValue(4, forKey: "Four")
print(stringIntDict) ////["Three": 3, "Two": 2, "Four": 4,
"One": 1]
```

```swift
// Finding number of key-value pairs
print(stringIntDict.count) // 4

// Iterating through collection

for (key,value) in stringIntDict
{
    print("\(key):\(value)")
}
/*
    Three:3
    Two:2
    Four:4
    One:1
 */
```

```swift
// Comparing two dictionaries

let bDict = ["Ten": 10, "Twelve": 12]

if stringIntDict != bDict
{
    print("Two dictionaries are not equal")
}
// O/P: Two dictionaries are not equal
// Deleting elements
stringIntDict.removeValue(forKey: "Three")
print(stringIntDict) //["Two": 2, "Four": 4, "One": 1]

print(stringIntDict.removeAll())
```

Sets

Unordered collection of similar datatype elements

Operations

Definition: Set is a collection of Similar datatype unique values and these are stored in undefined order

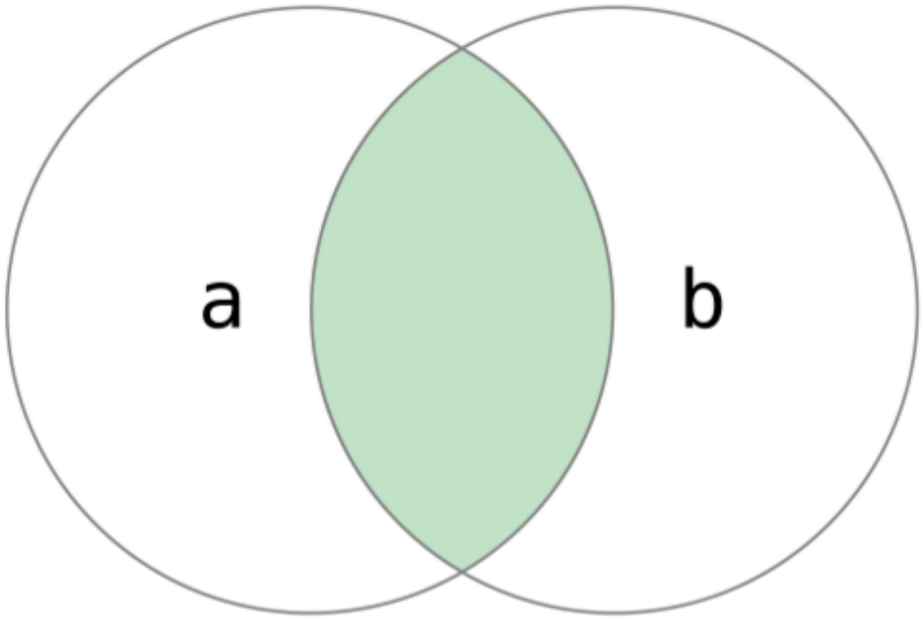
A set stores distinct values of the same type in a collection with no defined ordering. You can use a set instead of an array when the order of items is not important, or when you need to ensure that an item only appears once.

- **Creating Sets**
- **Finding number of Elements in a Set**
- **Comparing Two Sets**
- **Accessing Set's Elements**
- **Checking the existence of an Element**
- **Creating a Set from Array**
- **Empty Check**
- **Inserting an element in a Set**
- **Removing an Element from a Set**
- **Performing Set operations**
- **Removing all Objects**

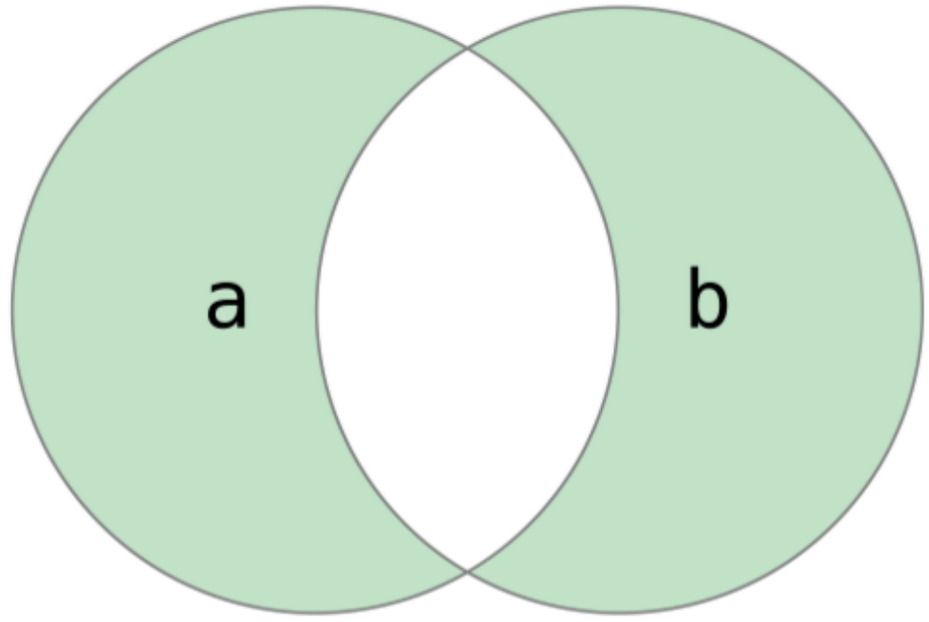
```
var aSet: Set = ["One", "Two", "Three", "Four", "Five", "z"];
let bSet = aSet;
print("Number of elements in set : \(aSet.count)"); // 5
// Comparison
if aSet == bSet
{
    print("aSet and bSet are equal");
}

// Existence check
let isExisting = aSet.contains("Four");
if isExisting
{
    print("Value Five is existing in aSet");
}
else
{
    print("Value Five is not existing in a set");
}
```

```
// Inserting an element
aSet.insert("New Element")
print(aSet); // ["Five", "Two", "One", "Three", "Four", "New Element”]

// Empty Check
if aSet.isEmpty
{
    print("aSet is empty”);
}
// Joining all elements into a string
print(aSet.joinWithSeparator(" ")); // Five Two One Three Four New
Element

// Removing particular element
aSet.remove(“One");

// Removing all objects
aSet.removeAll()

// Sorting
print(aSet.sort());

// Enumerating through collections
for item in aSet
{
    print(item);
}
```

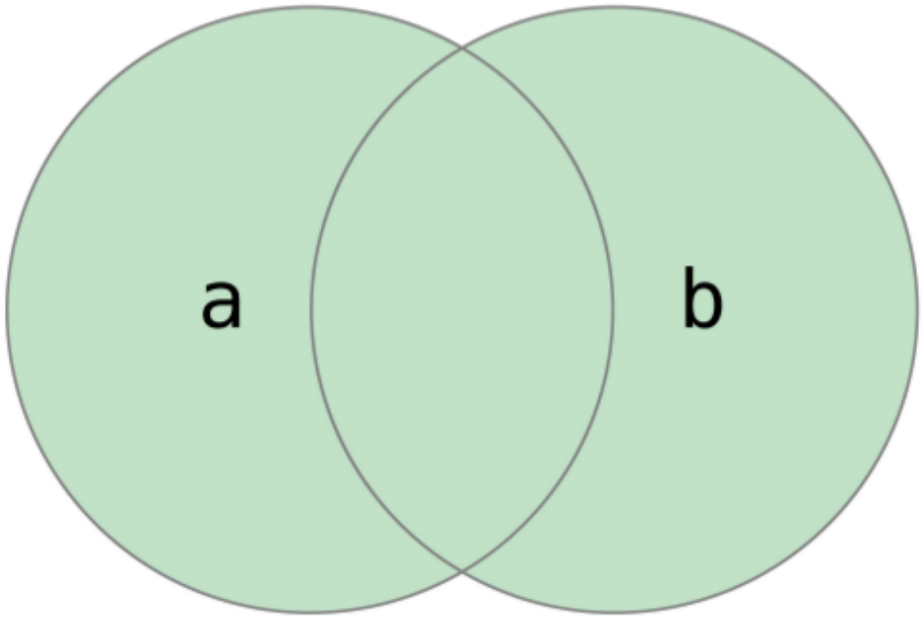
a.intersect(b)



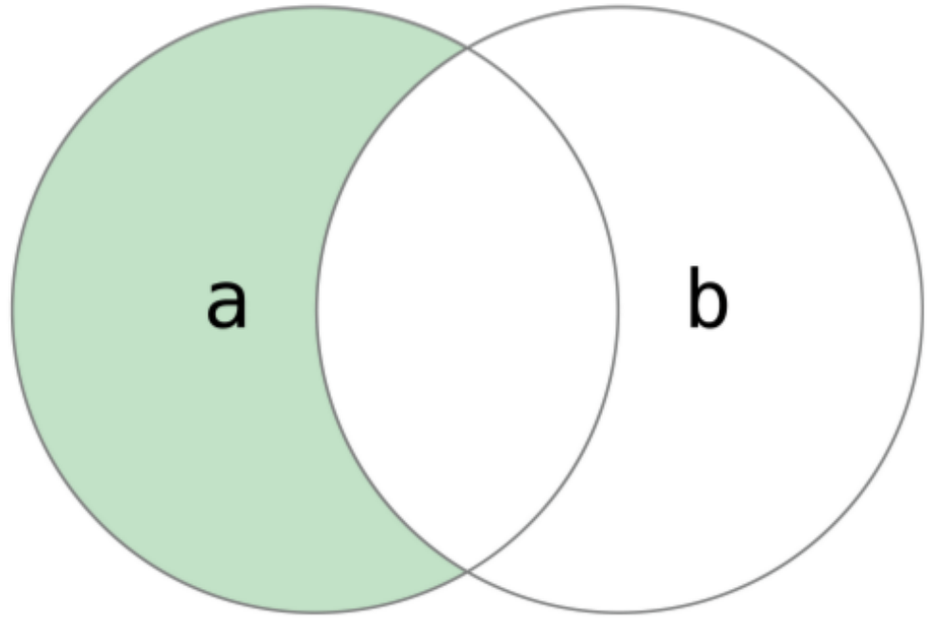
a.exclusiveOr(b)



a.union(b)



a.subtract(b)



Set Operations

```
let cSet: Set<String> = ["1", "2", "3", "4", "5", "6"];
let dSet: Set<String> = [ "4", "5", "6", "7", "8", “9"];

print(cSet.subtract(dSet)); // 1, 2, 3
print(cSet.union(dSet)); // 1, 2, 3, 4, 5, 6, 7, 8, 9
print(cSet.intersect(dSet)); // 4, 5, 6
print(cSet.exclusiveOr(dSet)); //1, 2, 3, 7, 8, 9
```

Thank you

For Offline / Online Training, reach me@ iPhoneDev1990@gmail.com