

## Swift - Sets

Swift 4 **sets** are used to store distinct values of same types but they don't have definite ordering as arrays have.

You can use sets instead of arrays if ordering of elements is not an issue, or if you want to ensure that there are no duplicate values. (sets allow only distinct values.)

A type must be hashable to be stored in a set. A hash value is a Int value that is equal for equal objects. For example, if `x == y`, then **`x.hashvalue == y.hashvalue`**.

All the basic swift values are of hashable type by default and may be used as set values.

### Creating Sets

You can create an empty set of a certain type using the following initializer syntax –

```
var someSet = Set<Character>()    //Character can be replaced by data type
```

### Accessing and modifying Sets

You can access or modify a set using its methods and properties –

"count" method can be used to show the number of elements in the set.

```
someSet.count    // prints the number of elements
```

"insert" method can be used to insert values in set.

```
someSet.insert("c")    // adds the element to Set.
```

Similarly, isEmpty can be used to check if set is empty.

```
someSet.isEmpty    // returns true or false depending on the set Element
```

"remove" method can be used to remove value in set.

```
someSet.remove("c")    // removes a element , removeAll() can be used to r
```

"contains" method can be used to check existence of value in a set.

```
someSet.contains("c")    // to check if set contains this value.
```

## Iterating over a Set

You can iterate over a set using for-in loop –

```
for items in someSet {  
    print(someSet)  
}  
  
//Swift sets are not in an ordered way, to iterate over a set in ordered way  
  
for items in someSet.sorted() {  
    print(someSet)  
}
```

## Performing Set Operations

You can perform basic set operations on swift Sets.

Following are the methods for performing set operations –

- Intersection
- Union
- subtracting

```
let evens: Set = [10,12,14,16,18]  
let odds: Set = [5,7,9,11,13]  
let primes = [2,3,5,7]  
odds.union(evens).sorted()  
// [5,7,9,10,11,12,13,14,16,18]  
odds.intersection(evens).sorted()  
//[ ]  
odds.subtracting(primes).sorted()  
//[9, 11, 13]
```