**NSArray**

It represents an ordered collection of objects, and it provides a high-level interface for sorting and otherwise manipulating lists of data. Immutable arrays can be defined as literals using the @[ ] syntax. objectAtIndex: method is used to access the element at a particular index.

**Comparing Arrays:**

Arrays can be compared for equality using the method namedisEqualToArray: , which returns YES when both arrays have the same number of elements and every pair pass an isEqual: comparison. NSArray does not offer the same subset and intersection comparisons as NSSet.

**Sorting Arrays:**

Sorting is one of the main advantages of arrays. thesortedArrayUsingComparator:  method. It is used to sort the array. This accepts an NSComparisonResult(id obj1, id obj2) block, which should return one of the following enumerators depending on the relationship between obj1 and obj2:

| Return Value | Description |
| --- | --- |
| NSOrderedAscending | obj1 comes before obj2 |
| NSOrderedSame | obj1 and obj2 have no order |
| NSOrderedDescending | obj1 comes after obj2 |

**Filtering Array:**

Array can be filtered with the filteredArrayUsingPredicate:method.

**Subdividing Array:**

Subdividing an array is essentially the same as extracting substrings from an NSString, but instead of substringWithRange:, we can use subarrayWithRange:.

**Combining Arrays:**

Arrays can be combined via arrayByAddingObjectsFromArray:.

**NSMutableArray:**

In The NSMutableArray class dynamically items can be added or removed from arbitrary locations in the collection. it’s slower to insert or delete elements from a mutable array than a set or a dictionary.

**Creating Mutable Array:**

To create mutable arrays is still through the arrayWithObjects: method. You can create empty mutable arrays using the array orarrayWithCapacity: class methods. Or, if you already have an immutable array that you want to convert to a mutable one, you can pass it to the arrayWithArray: class method.

**Adding and Removing objects:**

The two basic methods for manipulating the contents of an array are the addObject: and removeLastObject methods. The former adds an object to the end of the array, and the latter is pretty self-documenting. Insert or delete objects at arbitrary locations usinginsertObject:atIndex: and removeObjectAtIndex:. It’s also possible to replace the contents of an index with thereplaceObjectAtIndex:withObject: method.