NSMutableDictionary

* The NSMutableDictionary class declares the programmatic interface to objects that manage mutable associations of keys and values.
* It is subclass of NSDictionary.
* It adds modification operations to the basic operations it inherits from [NSDictionary](https://developer.apple.com/library/mac/documentation/Cocoa/Reference/Foundation/Classes/NSDictionary_Class/index.html#//apple_ref/occ/cl/NSDictionary).
* [NSMutableDictionary](http://rypress.com/tutorials/objective-c/data-types/nsdictionary#nsmutabledictionary) data structure lets you dynamically add and remove entries as necessary.
* The NSMutableDictionary class lets you add new key-value pairs dynamically.
* Mutable collections in general lend themselves to representing system states, and mutable dictionaries are no different.
* NSMutableDictionary is inherited from NSDictionary and hence all instance methods of NSDictionary are available in NSMutableDictionary.

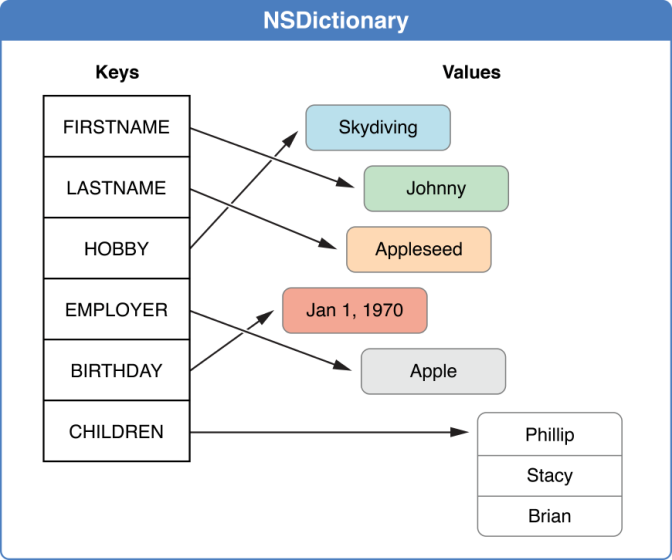


Fig a): NSDictionary collection classes of the Foundation Framework.

Courtesy: Apple Documentation

**Important methods of NSMutableDictionary are as follows:**

* removeAllObjects: Empties the dictionary of its entries.
* removeObjectForKey: Removes a given key and its associated value from the dictionary.
* setValue:forKey: Adds a given key-value pair to the dictionary.

## Creating Mutable Dictionaries:

Mutable dictionaries can be created by calling any of the factory methods defined by NSDictionary on the NSMutableDictionary class.

NSMutableDictionary \*jobs = [NSMutableDictionary

dictionaryWithDictionary:@{

@"Audi TT" : @"John",

@"Audi Quattro (Black)" : @"Mary",

@"Audi Quattro (Silver)" : @"Bill",

@"Audi A7" : @"Bill"

}];

NSLog(@"%@", jobs);

## Adding and Removing Entries:

The setObject:forKey: and removeObjectForKey: methods are the significant additions contributed by NSMutableDictionary.

NSMutableDictionary \*jobs = [NSMutableDictionary

dictionaryWithDictionary:@{

@"Audi TT" : @"John",

@"Audi Quattro (Black)" : @"Mary",

@"Audi Quattro (Silver)" : @"Bill",

@"Audi A7" : @"Bill"

}];

// Transfer an existing job to Mary

[jobs setObject:@"Mary" forKey:@"Audi TT"];

// Finish a job

[jobs removeObjectForKey:@"Audi A7"];

// Add a new job

jobs[@"Audi R8 GT"] = @"Jack";

## Enumeration Considerations:

* Dictionaries should not be mutated while you’re iterating over them.
* dictionaryWithDictionary: class method to create a shallow copy of the entire dictionary.

**Inherits From:**

[NSObject](https://developer.apple.com/library/mac/documentation/Cocoa/Reference/Foundation/Classes/NSObject_Class/index.html#//apple_ref/occ/cl/NSObject)

NSDictionary

[NSMutableDictionary](https://developer.apple.com/library/mac/documentation/Cocoa/Reference/Foundation/Classes/NSMutableDictionary_Class/index.html#//apple_ref/occ/cl/NSMutableDictionary)

**Import Statement:**

* @import Foundation;

**Availability:**

* Available in OS X v10.0 and later

**Advantages:**

* It stores two objects one is key and another one is value dynamically.
* It is better choice than mutable arrays.
* A dictionary is also a form of hash table.

**Disadvantage:**

* In a subclass, you must override both of its primitive methods:
* setObject:forKey
* removeObjectForKey
* You must also override the primitive methods of the [NSDictionary](https://developer.apple.com/library/mac/documentation/Cocoa/Reference/Foundation/Classes/NSDictionary_Class/index.html#//apple_ref/occ/cl/NSDictionary) class.

**References:**

* <https://developer.apple.com/library/mac/documentation/Cocoa/Reference/Foundation/Classes/NSArray_Class/index.html#//apple_ref/occ/cl/NSArray>
* <http://rypress.com/tutorials/objective-c/data-types/nsarray>
* <http://as.wiley.com/WileyCDA/Section/id-400181.html>
* <http://www.tutorialspoint.com/objective_c/objective_c_data_storage.htm>