NSMutableString

* The [NSMutableString](https://developer.apple.com/library/mac/#documentation/Cocoa/Reference/Foundation/Classes/NSMutableString_Class/Reference/Reference.html) class is a mutable version of NSString.
* Unlike immutable strings, it’s possible to alter individual characters of a mutable string without creating a brand new object.
* The NSString class and its mutable subclass, [NSMutableString](https://developer.apple.com/library/mac/documentation/Cocoa/Reference/Foundation/Classes/NSMutableString_Class/index.html#//apple_ref/occ/cl/NSMutableString), provide an extensive set of APIs for working with strings, including methods for comparing, searching, and modifying strings.
* This is preferred data structure when you’re performing several small edits on the same string.
* NSMutableString inherits from NSString, so aside from the ability to manipulate it in place, you can use a mutable string just like you would an immutable string.
* NSMutableString inherits from NSString, so aside from the ability to manipulate it in place, you can use a mutable string just like you would an immutable string.
* The fundamental workflow for mutable strings is different than that of immutable ones.
* Instead of creating a new object and replacing the old value, NSMutableString methods operate directly on the existing instance.
* Over distributed-object connections, mutable string objects are passed by-reference and immutable string objects are passed by-copy
* To construct and manage a string that can be changed after it has been created, use [NSMutableString](https://developer.apple.com/library/mac/documentation/Cocoa/Reference/Foundation/Classes/NSMutableString_Class/index.html#//apple_ref/occ/cl/NSMutableString).

## Creating Mutable Strings:

Mutable strings can be created through the stringWithString: class method, which turns a literal string or an existing NSString object into a mutable one:

NSMutableString \*car = [NSMutableString stringWithString:@"Porsche 911"];

After you’ve created a mutable string, the setString: method lets you assign a new value to the instance:

[car setString:@"Porsche Boxster"];

Compare this to NSString, where you re-assign a new value to the variable. With mutable strings, we don’t change the instance reference, but rather manipulate its contents through the mutable API.

## Replacing/Deleting Substrings

It’s possible to replace or delete substrings via thereplaceCharactersInRange:withString: anddeleteCharactersInRange: methods, as shown below.

NSMutableString \*car = [NSMutableString stringWithCapacity:20];

[car setString:@"Lotus Elise"];

[car replaceCharactersInRange:NSMakeRange(6, 5)

withString:@"Exige"];

NSLog(@"%@", car); // Lotus Exige

[car deleteCharactersInRange:NSMakeRange(5, 6)];

NSLog(@"%@", car);

## When to Use Mutable Strings:

* A good rule of thumb is to use a mutable string whenever you’re running any kind of algorithm that edits or assembles a string in several passes and to use an immutable string for everything else.
* This also applies to [sets](http://rypress.com/tutorials/objective-c/data-types/nsset.html), [arrays](http://rypress.com/tutorials/objective-c/data-types/nsarray.html), and [dictionaries](http://rypress.com/tutorials/objective-c/data-types/nsdictionary.html).

**Inherits From:**

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

NSString

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

**Import Statement:**

* @import Foundation;

**Availability:**

* Available in OS X v10.0 and later

**Advantages:**

* It stores any type of character.
* It has many built in methods.

**Disadvantage:**

* Any subclass of NSString must override the primitive instance methods [length](https://developer.apple.com/library/mac/documentation/Cocoa/Reference/Foundation/Classes/NSString_Class/index.html#//apple_ref/occ/instp/NSString/length) and [characterAtIndex:](https://developer.apple.com/library/mac/documentation/Cocoa/Reference/Foundation/Classes/NSString_Class/index.html#//apple_ref/occ/instm/NSString/characterAtIndex:)

**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>