***NSString***

The Objective-C class for strings is NSString. Strings are typically created by direct assignment or by calling one of the NSString class methods. The NSString class provides an extensive set of APIs for working with strings, including methods for comparing, searching, and modifying strings. Just like other languages, strings are enclosed and defined by the use of “double quotes”, however in Objective-C an NSString also comes with the prefix of the @ sign.

Syntax:

NSString \*variable name = @”statement”;

E.g.:

NSString string1 \*stirng1 = @”This is NSString”;

Inherits from:

NSObject

NSString

**Comparing String:**

NSString uses the isEqualToString method to compare the two strings and also with this it uses hassuffix and hasprefix methods for partial comparison.

Eg:

NSString \*bus = @”KSRTC red bus”;

If ([bus isEqualToString:@”This is red bus”])

{

NSLog(@”The bus is red”);

}

If ([bus hasprefix:@”KSRTC”])

{

NSLog(@“it is the ksrtc bus of some color”);

}

If ([bus hassuffix:@”bus”])

{

NSLog(@“this is the bus”);

}

**Combining the String:**

NSString is an immutable type so whenever we concatenate the new strings will be created. It uses the two methods to combine the strings they are 1)stringByAppendingString and 2)stringByAppendingFormat.

Eg:

NSString \*fname = @”Harish”;

NSString \*lname = @”Bhardwaj”;

NSString \*name = [ fname stringByAppendingString lname ]

NSLog(@”%@”,name);//HarishBhardwaj

NSString \*name = [ fname stringByAppendingFormat lname ]

NSLog(@”%@”,name);//Harish Bharadwaj

**Searching String:**

The NSString search method returns the NSRange which defines a location and length field. The Location contains the beginning match of the string and length field has toatal number of characters present in the string. if no match was found, location will contain NSNotFound.

**Changing case:**

The NSString change case is used to convert the string from lower case to upper case and upper case to lower case.

Eg:

NSString \*name = @”NeoRays”;

NSLog(@”%@”, [name uppercaseStirng]);//NEORAYS

NSLog(@”%@”, [ name lowercaseString]);//neorays

**2>NSMutabaleString:**

NSObject

NSString

NSMutableString

The NSMutableString class is the mutable version of NSString. The NSMutableString will not create a new string for any changes made for the string. There are so many methods are there to support for the NSMutableString. The mutable string is created with StringewithString Class method. Which turns the NSString object into mutableString. The NSMutableString class declares the programmatic interface to an object that manages a mutable string—that is, a string whose contents can be edited—that conceptually represents an array of Unicode characters. To construct and manage an immutable string—or a string that cannot be changed after it has been created—use an object of the [NSString](https://developer.apple.com/library/mac/documentation/Cocoa/Reference/Foundation/Classes/NSString_Class/index.html#//apple_ref/occ/cl/NSString) class.

Eg:

NSMutableString \*name = [NSMutableString StringwithString :@”Harish”];

After created a mutable string we can assign the new value for the string using setString method.

The NSMutableString class adds one primitive method

[replaceCharactersInRange:withString:](https://developer.apple.com/library/mac/documentation/Cocoa/Reference/Foundation/Classes/NSMutableString_Class/index.html#//apple_ref/occ/instm/NSMutableString/replaceCharactersInRange:withString:)—to the basic string-handling behavior inherited from NSString. All other methods that modify a string work through this method. For example, [insertString:atIndex:](https://developer.apple.com/library/mac/documentation/Cocoa/Reference/Foundation/Classes/NSMutableString_Class/index.html#//apple_ref/occ/instm/NSMutableString/insertString:atIndex:) simply replaces the characters in a range of 0 length, while [deleteCharactersInRange:](https://developer.apple.com/library/mac/documentation/Cocoa/Reference/Foundation/Classes/NSMutableString_Class/index.html#//apple_ref/occ/instm/NSMutableString/deleteCharactersInRange:) replaces the characters in a given range with no characters.