TEXT AND STRINGS IN OBJECTIVE C

NSString is one the most commonly used classes that is used for storing strings and texts. If you want to know more about NSString, please refer NSString in Objective-C strings.

As mentioned earlier, NSCharacterSet represents various groupings of characters that are used by the NSString and NSScanner classes.

NSCharacterSet

Here is the set of methods available in NSCharacterSet which represent the various character sets.

- alphanumericCharacterSet - Returns a character set containing the characters in the categories Letters, Marks, and Numbers.

- capitalizedLetterCharacterSet - Returns a character set containing the characters in the category of Titlecase Letters.

- characterSetWithCharactersInString - Returns a character set containing the characters in a given string.

- characterSetWithRange: - Returns a character set containing characters with Unicode values in a given range.

- illegalCharacterSet - Returns a character set containing values in the category of Non-Characters or that have not yet been defined in version 3.2 of the Unicode standard.

- letterCharacterSet - Returns a character set containing the characters in the categories Letters and Marks.

- lowercaseLetterCharacterSet - Returns a character set containing the characters in the category of Lowercase Letters.

- newlineCharacterSet - Returns a character set containing the newline characters.

- punctuationCharacterSet - Returns a character set containing the characters in the category of Punctuation.

- symbolCharacterSet - Returns a character set containing the characters in the category of Symbols.

- uppercaseLetterCharacterSet - Returns a character set containing the characters in the categories of Uppercase Letters and Titlecase Letters.

- whitespaceAndNewlineCharacterSet - Returns a character set containing Unicode General Category Z*, U000A ~ U000D, and U0085.

- whitespaceCharacterSet - Returns a character set containing only the in-line whitespace characters space $U+0020$ and tab $U+0009$.

```
#import <Foundation/Foundation.h>

int main()
{
   NSAutoreleasePool * pool = [[NSAutoreleasePool alloc] init];
   NSString *string = @"....Tutorials Point.com.....";
   NSLog(@"Initial String :%@", string);
   NSCharacterSet *characterset = [NSCharacterSet punctuationCharacterSet];
   string = [string stringByTrimmingCharactersInSet:characterset];
   NSLog(@"Final String :%@", string);
   [pool drain];
   return 0;
}
```

Now when we compile and run the program, we will get the following result.

```
2013-09-29 14:19:27.328 demo[687]  Initial String :....Tutorials Point.com.....
2013-09-29 14:19:27.328 demo[687 Final String :Tutorials Point.com
```

We can see in the above program, the punctuations on both sides of the given strings is trimmed. It's just an example of using NSCharacterSet.