String in Apex, as in any other programming language, is any set of characters with no character limit.

Example:

```
String companyName = 'Abc International';
System.debug('Value companyName variable'+companyName);
```

String Methods

String class in Salesforce has many methods. We will take a look at some of the most important and frequently used string methods in this chapter.

contains

This method will return true if the given string contains the substring mentioned.

Syntax:

```
public Boolean contains(String substring)
```

Example:

```
String myProductName1 = 'HCL';
String myProductName2 = 'NAHCL';
Boolean result = myProductName1.contains(myProductName2);
System.debug('O/p will be true as it contains the String and Output is: '+result );
```

equals

This method will return true if the given string and the string passed in the method have the same binary sequence of characters and they are not null. You could compare the SFDC record id as well using this method. This method is case sensitive.

Syntax:

```
public Boolean equals(Object string)
```

Example:

```
String myString1 = 'MyString';
String myString2 = 'MyString';
Boolean result = myString2.equals(myString1);
System.debug('Value of Result will be true as they are same and Result is:'+result);
```

equalsIngoreCase

This method will return true if stringtoCompare has the same sequence of characters as the given string. However, this method is not case sensitive.

Syntax:

```
public Boolean equalsIgnoreCase(String stringtoCompare)
```

Example:

Below code will return true as string characters and sequence are same, ignoring the case sensitivity.

```
String myString1 = 'MySTRING';
String myString2 = 'MyString';
Boolean result = myString2.equalsIgnoreCase(myString1);
System.debug('Value of Result will be true as they are same and Result is:'+result);
```

remove

This method removes the string provided in stringToRemove from given string. This is useful when you want to remove some specific characters from string and don't know the exact index of the characters to remove. This method is case sensitive and will not work if the same character sequence occurs but case is different.

Syntax:

```
public String remove(String stringToRemove)
```

Example:

```
String myString1 = 'This Is MyString Example';
String stringToRemove = 'MyString';
String result = myString1.remove(stringToRemove);
System.debug('Value of Result will be 'This Is Example' as we have removed the MyString
and Result is :'+result);
```

removeEndIgnoreCase

This method removes the string procvided in stringToRemove from the given string but only if it occurs at the end. This method is not case sensitive.

Syntax:

```
public String removeEndIgnoreCase(String stringToRemove)
```

Example:

```
String myString1 = 'This Is MyString EXAMPLE';
String stringToRemove = 'Example';
String result = myString1.removeEndIgnoreCase(stringToRemove);
System.debug('Value of Result will be 'This Is MyString' as we have removed the 'Example'
and Result is :'+result);
```

startsWith

This method will return true if the given string starts with the prefix provided in the method.

Syntax:

```
public Boolean startsWith(String prefix)
```

Example:

```
String myString1 = 'This Is MyString EXAMPLE';
String prefix = 'This';
Boolean result = myString1.startsWith(prefix);
System.debug(' This will return true as our String starts with string 'This' and the
Result is :'+result);
```