C# - STRINGS

In C#, you can use strings as array of characters, However, more common practice is to use the **string** keyword to declare a string variable. The string keyword is an alias for the **System.String** class.

Creating a String Object

You can create string object using one of the following methods –

- By assigning a string literal to a String variable
- By using a String class constructor
- By using the string concatenation operator +
- By retrieving a property or calling a method that returns a string
- By calling a formatting method to convert a value or an object to its string representation

The following example demonstrates this –

```
using System;

namespace StringApplication {

   class Program {

      static void Main(string[] args) {
         //from string literal and string concatenation
         string fname, lname;
         fname = "Rowan";
         lname = "Atkinson";

         char []letters= { 'H', 'e', 'l', 'l','o' };
         string [] sarray={ "Hello", "From", "Tutorials", "Point" };

         string fullname = fname + lname;
         Console.WriteLine("Full Name: {0}", fullname);

         //by using string constructor { 'H', 'e', 'l', 'l','o' };
         string greetings = new string(letters);
         Console.WriteLine("Greetings: {0}", greetings);

         //methods returning string { "Hello", "From", "Tutorials", "Point" };
         string message = String.Join(" ", sarray);
         Console.WriteLine("Message: {0}", message);

         //formatting method to convert a value
         DateTime waiting = new DateTime(2012, 10, 10, 17, 58, 1);
         string chat = String.Format("Message sent at {0:t} on {0:D}", waiting);
         Console.WriteLine("Message: {0}", chat);
```

```
      }
   }
}
```

When the above code is compiled and executed, it produces the following result –

```
Full Name: RowanAtkinson
Greetings: Hello
Message: Hello From Tutorials Point
Message: Message sent at 5:58 PM on Wednesday, October 10, 2012
```

Properties of the String Class

The String class has the following two properties –

Sr.No.	Property & Description
1	**Chars** Gets the *Char* object at a specified position in the current *String* object.
2	**Length** Gets the number of characters in the current String object.

Methods of the String Class

The String class has numerous methods that help you in working with the string objects. The following table provides some of the most commonly used methods –

Given below is the list of methods of the String class.

Sr.No.	Methods & Description
1	**public static int Compare**$stringstrA, stringstrB$ Compares two specified string objects and returns an integer that indicates their relative position in the sort order.
2	**public static int Compare**$stringstrA, stringstrB, boolignoreCase$

	Compares two specified string objects and returns an integer that indicates their relative position in the sort order. However, it ignores case if the Boolean parameter is true.
3	**public static string Concat***stringstr*0, *stringstr*1 Concatenates two string objects.
4	**public static string Concat***stringstr*0, *stringstr*1, *stringstr*2 Concatenates three string objects.
5	**public static string Concat***stringstr*0, *stringstr*1, *stringstr*2, *stringstr*3 Concatenates four string objects.
6	**public bool Contains***stringvalue* Returns a value indicating whether the specified String object occurs within this string.
7	**public static string Copy***stringstr* Creates a new String object with the same value as the specified string.
8	**public void CopyTo***intsourceIndex*, *char*[]*destination*, *intdestinationIndex*, *intcount* Copies a specified number of characters from a specified position of the String object to a specified position in an array of Unicode characters.
9	**public bool EndsWith***stringvalue* Determines whether the end of the string object matches the specified string.
10	**public bool Equals***stringvalue* Determines whether the current String object and the specified String object have the same value.

11	**public static bool Equals***stringa*, *stringb* Determines whether two specified String objects have the same value.
12	**public static string Format***stringformat*, *Objectarg*0 Replaces one or more format items in a specified string with the string representation of a specified object.
13	**public int IndexOf***charvalue* Returns the zero-based index of the first occurrence of the specified Unicode character in the current string.
14	**public int IndexOf***stringvalue* Returns the zero-based index of the first occurrence of the specified string in this instance.
15	**public int IndexOf***charvalue*, *intstartIndex* Returns the zero-based index of the first occurrence of the specified Unicode character in this string, starting search at the specified character position.
16	**public int IndexOf***stringvalue*, *intstartIndex* Returns the zero-based index of the first occurrence of the specified string in this instance, starting search at the specified character position.
17	**public int IndexOfAny***char*[]*anyOf* Returns the zero-based index of the first occurrence in this instance of any character in a specified array of Unicode characters.
18	**public int IndexOfAny***char*[]*anyOf*, *intstartIndex*

	Returns the zero-based index of the first occurrence in this instance of any character in a specified array of Unicode characters, starting search at the specified character position.
19	**public string Insert***intstartIndex*, *stringvalue* Returns a new string in which a specified string is inserted at a specified index position in the current string object.
20	**public static bool IsNullOrEmpty***stringvalue* Indicates whether the specified string is null or an Empty string.
21	**public static string Join***stringseparator*, *paramsstring*[]*value* Concatenates all the elements of a string array, using the specified separator between each element.
22	**public static string Join***stringseparator*, *string*[]*value*, *intstartIndex*, *intcount* Concatenates the specified elements of a string array, using the specified separator between each element.
23	**public int LastIndexOf***charvalue* Returns the zero-based index position of the last occurrence of the specified Unicode character within the current string object.
24	**public int LastIndexOf***stringvalue* Returns the zero-based index position of the last occurrence of a specified string within the current string object.
25	**public string Remove***intstartIndex* Removes all the characters in the current instance, beginning at a specified position and continuing through the last position, and returns the string.

26	**public string Remove**$intstartIndex, intcount$ Removes the specified number of characters in the current string beginning at a specified position and returns the string.
27	**public string Replace**$charoldChar, charnewChar$ Replaces all occurrences of a specified Unicode character in the current string object with the specified Unicode character and returns the new string.
28	**public string Replace**$stringoldValue, stringnewValue$ Replaces all occurrences of a specified string in the current string object with the specified string and returns the new string.
29	**public string[] Split**$paramschar[]separator$ Returns a string array that contains the substrings in the current string object, delimited by elements of a specified Unicode character array.
30	**public string[] Split**$char[]separator, intcount$ Returns a string array that contains the substrings in the current string object, delimited by elements of a specified Unicode character array. The int parameter specifies the maximum number of substrings to return.
31	**public bool StartsWith**$stringvalue$ Determines whether the beginning of this string instance matches the specified string.
32	**public char[] ToCharArray** Returns a Unicode character array with all the characters in the current string object.
33	**public char[] ToCharArray**$intstartIndex, intlength$

	Returns a Unicode character array with all the characters in the current string object, starting from the specified index and up to the specified length.
34	**public string ToLower** Returns a copy of this string converted to lowercase.
35	**public string ToUpper** Returns a copy of this string converted to uppercase.
36	**public string Trim** Removes all leading and trailing white-space characters from the current String object.

You can visit MSDN library for the complete list of methods and String class constructors.

Examples

The following example demonstrates some of the methods mentioned above –

Comparing Strings

```
using System;

namespace StringApplication {

   class StringProg {

      static void Main(string[] args) {
         string str1 = "This is test";
         string str2 = "This is text";

         if (String.Compare(str1, str2) == 0) {
            Console.WriteLine(str1 + " and " + str2 +  " are equal.");
         } else {
            Console.WriteLine(str1 + " and " + str2 + " are not equal.");
         }
         Console.ReadKey() ;
      }
   }
}
```

When the above code is compiled and executed, it produces the following result –

```
This is test and This is text are not equal.
```

String Contains String

```
using System;

namespace StringApplication {

   class StringProg {

      static void Main(string[] args) {
         string str = "This is test";

         if (str.Contains("test")) {
            Console.WriteLine("The sequence 'test' was found.");
         }
         Console.ReadKey() ;
      }
   }
}
```

When the above code is compiled and executed, it produces the following result –

```
The sequence 'test' was found.
```

Getting a Substring

```
using System;

namespace StringApplication {

   class StringProg {

      static void Main(string[] args) {
         string str = "Last night I dreamt of San Pedro";
         Console.WriteLine(str);
         string substr = str.Substring(23);
         Console.WriteLine(substr);
      }
   }
}
```

When the above code is compiled and executed, it produces the following result –

```
San Pedro
```

Joining Strings

```
using System;

namespace StringApplication {
```

```
   class StringProg {

      static void Main(string[] args) {
         string[] starray = new string[]{"Down the way nights are dark",
            "And the sun shines daily on the mountain top",
            "I took a trip on a sailing ship",
            "And when I reached Jamaica",
            "I made a stop"};

         string str = String.Join("\n", starray);
         Console.WriteLine(str);
      }
   }
}
```

When the above code is compiled and executed, it produces the following result –

```
Down the way nights are dark
And the sun shines daily on the mountain top
I took a trip on a sailing ship
And when I reached Jamaica
I made a stop
```