# C# Working with Strings

In programming a String is a specialised type of array. The name String is derived from a string of characters. A String can be very long or a short as “”. An entire book could be placed in a single String. This is convenient but of limited use if we cannot search through the book to find specific parts. As moving through a String to rearrange the contents or find specific Strings within other Strings are common programming problems there are a number of useful solutions built into C#.

**Comparing Strings**

The following code will change the case of a String, the new value is assigned to the same value as the old String.

```
String data = "upAndDown";
data = data.ToUpper();
Console.WriteLine(data);
data = data.ToLower();
Console.WriteLine(data);
```

Strings can be compared to see if they are the same.

```
String otherData = "upAndDown";
if(data == otherData)
         Console.WriteLine("The same");
else
         Console.WriteLine("Not the same");
```

The `CompareTo` method does the same as == but returns 0, 1 or -1. If the Strings are the same then 0 is returned. If 1 is returned the String in ( )s is before the other in the alphabet. If -1 is returned it is after the other in the alphabet.

```
int num = data.CompareTo(otherData);
if(num == 0)
        Console.WriteLine("The same " + num);
else
        Console.WriteLine("Not the same " + num);
```

**Searching Strings**

A String can be searched to see if another String is present within it. The method `Contains` will return if a char is present within a String.

```
bool num = data.Contains('p');
```

To find the exact position of a char within a String the method `IndexOf` is used. The 1st position in a String will be 0. This method will not identify subsequent occurances of the same char within that String

```
int num = data.IndexOf('p');
```

There is no specific method to find a String within another String but `IndexOfAny` will return the initial position of an array of chars within a String. An array of chars is good enough to hold a word.

```
char [] letts = {'A','n','d'};
int num = data.IndexOfAny(letts);
```

To get away with finding multiple occurances of a character or String within another String we are going to need arrays of chars. Luckily the method `ToCharArray` will convert a String to an array of chars. There is no easy way to get the array back into a String. A for loop would be needed to build up a new String char by char.

```
String lettsAsString = "And";
char[] letts = lettsAsString.ToCharArray();
int num = data.IndexOfAny(letts);
```

We know that there are 2 n's in the String variable data but C# can prove this with a char array and a for loop.

```
char find = 'n';
int count = 0;
char[] words = data.ToCharArray();
for (int x = 0; x < words.Length; x++)
{
      if (find == words[x])
            count++;
}
Console.WriteLine("count is " + count);
```

**Changing (manipulating) Strings**

It is not always necessary to spilt a String apart and put it back together to change the value of a String

The length of a String is returned by `Length` this ignores any whitespace at the beginning or end of the String.

```
num = data.Length;
```

`Remove` will remove all characters in a String starting at the specified index.

```
data = data.Remove(5);
```

Replace will work with Strings or char variables and replaces all instances of the specified variable.

```
data = data.Replace("And", "OR");
data = data.Replace('A', 'O');
```

These methods can be used to assign the changed value of 1 String to another

```
otherData = data.Replace('A', 'O');
```

As used below the new String has the same length as the 1st but all elements are replaced by '?'.

```
otherData = data.Replace("upAndDown", "?????????");
```

`Insert` can be used to insert 1 String into another. This is not a case of overtyping; the new String is pushed into the old at the specified position and any following characters are moved along.

```
otherData = otherData.Insert(2, temp);
```

To have the new characters overwrite the old the previous character will need to be removed.

```
otherData = otherData.Remove(3, 1);
```

In the above example the String variable temp has a length of 1 (it is a 1 letter String). This is inserted at position 2 into the String otherData. The single letter now at position 3 of otherData used to be at position 2 and has been passed to `Remove`.

**Hangman example**

There are enough code algorithms above to play hangman. The nature of the methods used means that some swapping between Strings, chars and arrays of chars is required. The solution outlined below does not include any looping to check for more than 1 character. It will work with multiple instances of the same letter (as in initiative which has a good supply of 'i's). If this feature were not required the solution could be simplified.

1. Create 2 Strings, the word to guess and another string that is the same length as the hidden word but consists of a placeholder character such as ?.
2. Create an array of chars that contains all the letters in the same order as within the word to be guessed (not the placeholder String).
3. Ask the user for a letter and convert the input to a char.
4. Use a for loop and run through all letters within the char array (see 2)
    a. Check if the char entered matches each letter in the array
    b. If it does convert the input to a String (or use the initial String input)
    c. Use Insert to put that String (letter) in the placeholder word at the correct position (this will be found from the counter of the for loop, unfortunately Insert works with Strings not chars).

    d. Use Remove to remove the ? that has been moved 1 along by the Insert method
5. Output the new value of the placeholder String. If a letter has been matched then the value of that String will have changed.

The following shows some sample output, in this run the letters compared are case sensitive:

```
????????? is now
Please enter a letter
n
2 letters found
???n????n is now
Please enter a letter
A
1 letters found
??An????n is now
Please enter a letter
x
0 letters found
??An????n is now
Please enter a letter
```