1. Strings

String is a class & They are immutable

- Useful string methods
- Converting a string to a number
- Converting a number to a string

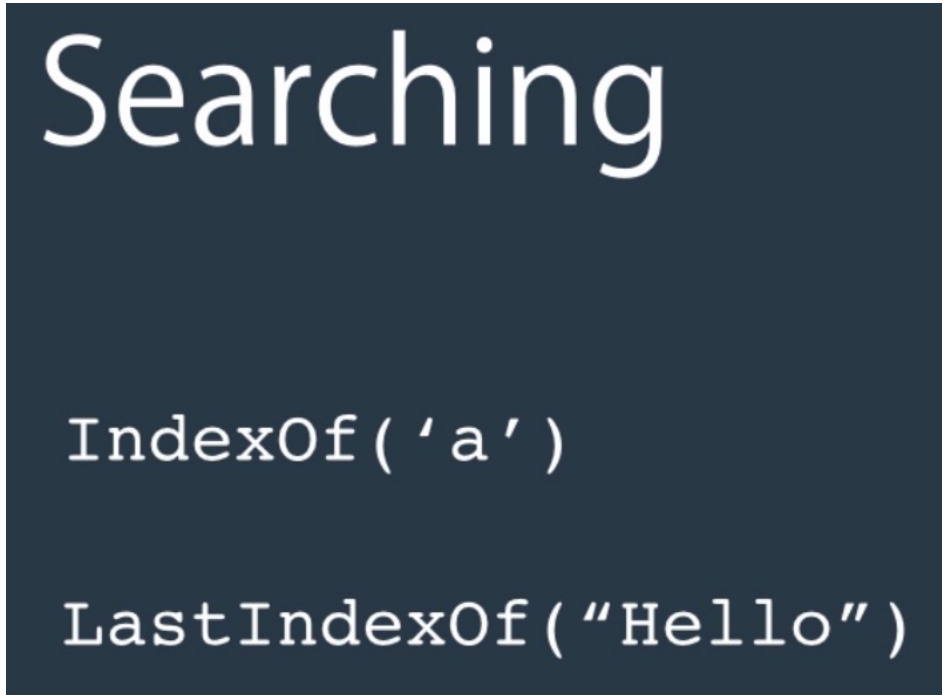
Formatting

```
ToLower()       // “hello world”

ToUpper()       // “HELLO WORLD”

Trim()
```

Trim - a useful method which gets rid of the white spaces around a string. This is important and when you capture user inputs in web forums or Windows Forms if you want to



Both of these methods have overload and they accept a character or a string and then return an index

Substrings

```
Substring(startIndex)

Substring(startIndex, length)
```

starIndex - Takes start the index and then retrieves all the characters from that point onwards to the end

Null checking

a common way to compare a string against now or empty string or whitespace.

```
String.IsNullOrEmpty(str)

String.IsNullOrWhiteSpace(str)
```

These are common ways to compare a string against null or empty string or whitespace.

It split a string by an empty character which means if we have a sentence with multiple words since there is a empty character between each of these words they will all be split.

And what will be returned from this method is an array of strings each element containing a word.

Converting Strings to Numbers

```
string s = “1234”;

int i = int.Parse(s);

int j = Convert.ToInt32(s);
```

Prefer to convert to Int32 because if the string is null or empty this method returns the default value for the integer which is zero. Whereas instead parse throws an exception. So it's easier and safer to work with convert to Int32.

Converting Numbers to Strings

```
int i = 1234;

string s = i.ToString();                    // “1234”

string t = i.ToString(“C”);                 // “$1,234.00”

string t = i.ToString(“C0”);                // “$1,234”
```

Format Strings

Format Specifier	Description	Example
c or **C**	Currency	123456 (C) -> $123,456
d or **D**	Decimal	1234 (D6) -> 001234
e or **E**	Exponential	1052.0329112756 (E) -> 1.052033E+003
f or **F**	Fixed Point	1234.567 (F1) -> 1234.5
x or **X**	Hexadecimal	255 (X) -> FF

```
1   using System;
2
3   namespace WorkingText
4   {
5       class Program
6       {
7           static void Main(string[] args)
8           {
9               var fullName = "Saqib Ali ";
10
11              Console.WriteLine("Trim: '{0}'", fullName.Trim());
12
13          }
14      }
15  }
16
```

Terminal – WorkingText

```
Trim: 'Saqib Ali'
```

Program ▸ Main(string[] args)

```
1   using System;
2
3   namespace WorkingText
4   {
5       class Program
6       {
7           static void Main(string[] args)
8           {
9               var fullName = "Saqib Ali ";
10
11              Console.WriteLine("Trim: '{0}'", fullName.Trim());
12              Console.WriteLine("ToUpper: '{0}'", fullName.Trim().ToUpper());
13
14          }
15      }
16  }
17
```

Terminal – WorkingText

```
Trim: 'Saqib Ali'
ToUpper: 'SAQIB ALI'
```

```
1   using System;
2
3   namespace WorkingText
4   {
5       class Program
6       {
7           static void Main(string[] args)
8           {
9               var fullName = "Saqib Ali ";
10
11              Console.WriteLine("Trim: '{0}'", fullName.Trim());
12              Console.WriteLine("ToUpper: '{0}'", fullName.Trim().ToUpper());
13
14              var index = fullName.IndexOf(' ');
15              Console.WriteLine("Index of ' ': {0} ", index);
16
17          }
18      }
19  }
```

Terminal – WorkingText

```
Trim: 'Saqib Ali'
ToUpper: 'SAQIB ALI'
Index of ' ': 5
```

```
using System;

namespace WorkingText
{
    class Program
    {
        static void Main(string[] args)
        {
            var fullName = "Saqib Ali ";

            Console.WriteLine("Trim: '{0}'", fullName.Trim());
            Console.WriteLine("ToUpper: '{0}'", fullName.Trim().ToUpper());

            var index = full
            Console.WriteLin

            var firstName = fullName.Substring()

        }
    }
}
```

1 of 2 string string.Substring(int **startIndex**)
Retrieves a substring from this instance. The substring starts at a specified character position and continues to the end of the string.
startIndex: *The zero-based starting character position of a substring in this instance.*

```
using System;

namespace WorkingText
{
    class Program
    {
        static void Main(string[] args)
        {
            var fullName = "Saqib Ali ";

            Console.WriteLine("Trim: '{0}'", fullName.Trim());
            Console.WriteLine("ToUpper: '{0}'", fullName.Trim().ToUpper());

            var index = full
            Console.WriteLin

            var firstName = fullName.Substring(0,)

        }
    }
}
```

2 of 2 string string.Substring(int startIndex, int length)
Retrieves a substring from this instance. The substring starts at a specified character position and has a specified length.
length: *The number of characters in the substring.*

```
using System;

namespace WorkingText
{
    class Program
    {
        static void Main(string[] args)
        {
            var fullName = "Saqib Ali ";

            Console.WriteLine("Trim: '{0}'", fullName.Trim());
            Console.WriteLine("ToUpper: '{0}'", fullName.Trim().ToUpper());

            var index = fullName.IndexOf(' ');
            Console.WriteLi

            var firstName = fullName.Substring(0, index);
            var lastName = fullName.Substring()

        }
    }
}
```

1 of 2 string string.Substring(int startIndex)
Retrieves a substring from this instance. The substring starts at a specified character position and continues to the end of the string.
startIndex: *The zero-based starting character position of a substring in this instance.*

```
using System;

namespace WorkingText
{
    class Program
    {
        static void Main(string[] args)
        {
            var fullName = "Saqib Ali ";

            Console.WriteLine("Trim: '{0}'", fullName.Trim());
            Console.WriteLine("ToUpper: '{0}'", fullName.Trim().ToUpper());

            var index = fullName.IndexOf(' ');
            Console.WriteLine("Index of ' ': {0} ", index);

            var firstName = fullName.Substring(0, index);
            var lastName = fullName.Substring(index + 1);

            Console.WriteLine("First Name: {0}", firstName);
            Console.WriteLine("Last Name: {0}", lastName);



        }
    }
}
```

Terminal – WorkingText

```
Trim: 'Saqib Ali'
ToUpper: 'SAQIB ALI'
Index of ' ': 5
First Name: Saqib
Last Name: Ali
```

```
    class Program
    {
        static void Main(string[] args)
        {
            var fullName = "Saqib Ali ";

            Console.WriteLine("Trim: '{0}'", fullName.Trim());
            Console.WriteLine("ToUpper: '{0}'", fullName.Trim().ToUpper());

            var index = fullName.IndexOf(' ');
            Console.WriteLine("Index of ' ': {0} ", index);

            var firstName = fullName.Substring(0, index);
            var lastName = fullName.Substring(index + 1);

            Console.WriteLine("First Name: {0}", firstName);
            Console.WriteLine("Last Name: {0}", lastName);

            var names = fullName.Split(' ');


        }
    }
}
```

```
M string[] string.Split(char separator, [StringSplitOptions options =
  StringSplitOptions.None]) (+ 9 overloads)

Splits a string into substrings based on the provided character separator.

Returns:

  An array whose elements contain the substrings from this instance that are
  delimited by separator.
```

```
    class Program
    {
        static void Main(string[] args)
        {
            var fullName = "Saqib Ali ";

            Console.WriteLine("Trim: '{0}'", fullName.Trim());
            Console.WriteLine("ToUpper: '{0}'", fullName.Trim().ToUpper());

            var index = fullName.IndexOf(' ');
            Console.WriteLine("Index of ' ': {0} ", index);

            var firstName = fullName.Substring(0, index);
            var lastName = fullName.Substring(index + 1);

            Console.WriteLine("First Name: {0}", firstName);
            Console.WriteLine("Last Name: {0}", lastName);

            va

        }
    }
}
```

class System.Console

Represents the standard input, output, and error streams for console applications. This class cannot be inherited.

```
            var fullName = "Saqib Ali ";

            Console.WriteLine("Trim: '{0}'", fullName.Trim(
            Console.WriteLine("ToUpper: '{0}'", fullName.Tr

            var index = fullName.IndexOf(' ');
            Console.WriteLine("Index of ' ': {0} ", index);

            var firstName = fullName.Substring(0, index);
            var lastName = fullName.Substring(index + 1);

            Console.WriteLine("First Name: {0}", firstName);
            Console.WriteLine("Last Name: {0}", lastName);

            var names = fullName.Split(' ');
            Console.WriteLine("First Name: {0}", names[0]);
            Console.WriteLine("Last Name: {0}", names[1]);

        }
    }
}
```

Terminal – WorkingText

```
Index of ' ': 5
First Name: Saqib
Last Name: Ali
First Name: Saqib
Last Name: Ali
```

```
13
14          var index = fullName.IndexOf(' ');
15          Console.WriteLine("Index of ' ': {0} ", index);
16
17          var firstName = fullName.Substring(0, index);
18          var lastName = fullName.Substring(index + 1);
19
20          Console.WriteLine("First Name: {0}", firstName);
21          Console.WriteLine("Last Name: {0}", lastName);
22
23          var names = fullName.Split(' ');
24          Console.WriteLine("First Name: {0}", names[0]);
25          Console.WriteLine("Last Name: {0}", names[1]);
26
27          var newName = fullName.Replace("Ali", "ALI");
28          Console.WriteLine("New Name is: {0}", newName);
29
30
```

Terminal – WorkingText

```
First Name: Saqib
Last Name: Ali
First Name: Saqib
Last Name: Ali
New Name is: Saqib ALI
```

```
26
27              var newName = fullName.Replace("Ali", "ALI");
28              Console.WriteLine("New Name is: {0}", newName);
29
30              if (String.IsNullOrEmpty(""))
31                  Console.WriteLine("Invalid");
32
33              if (String.IsNullOrEmpty(null))
34                  Console.WriteLine("Invalid");
35          }
36      }
37  }
38
```

Terminal – WorkingText

```
First Name: Saqib
Last Name: Ali
New Name is: Saqib ALI
Invalid
Invalid
```

```
29
30              if (String.IsNullOrEmpty(""))
31                  Console.WriteLine("Invalid");
32
33              if (String.IsNullOrEmpty(null))
34                  Console.WriteLine("Invalid");
35
36              if (String.IsNullOrEmpty(" "))
37                  Console.WriteLine("Invalid");
38          }
39      }
```

Terminal – WorkingText

```
First Name: Saqib
Last Name: Ali
New Name is: Saqib ALI
Invalid
Invalid
```

```
29
30              if (String.IsNullOrEmpty(""))
31                  Console.WriteLine("Invalid");
32
33              if (String.IsNullOrEmpty(null))
34                  Console.WriteLine("Invalid");
35
36              if (String.IsNullOrEmpty(" "))
37                  Console.WriteLine("Invalid");
38
39              if (String.IsNullOrEmpty(" ".Trim()))
40                  Console.WriteLine("Invalid");
41          }
42      }
43  }
```

Terminal – WorkingText

```
Last Name: Ali
New Name is: Saqib ALI
Invalid
Invalid
Invalid
```

```
32
33                  if (String.IsNullOrEmpty(null))
34                      Console.WriteLine("Invalid");
35
36                  if (String.IsNullOrEmpty(" "))
37                      Console.WriteLine("Invalid");
38
39                  if (String.IsNullOrEmpty(" ".Trim()))
40                      Console.WriteLine("Invalid");
41
42                  if (String.IsNullOrWhiteSpace(" "))
43                      Console.WriteLine("Invalid");
44              }
45          }
46      }
47
```

Terminal – WorkingText

```
New Name is: Saqib ALI
Invalid
Invalid
Invalid
Invalid
```

```
32
33              if (String.IsNullOrEmpty(null))
34                  Console.WriteLine("Invalid");
35
36              if (String.IsNullOrEmpty(" "))
37                  Console.WriteLine("Invalid");
38
39              if (String.IsNullOrEmpty(" ".Trim()))
40                  Console.WriteLine("Invalid");
41
42              if (String.IsNullOrWhiteSpace(" "))
43                  Console.WriteLine("Invalid");
44
45              var str = "25";
46              var age = Convert.ToByte(str);
47              Console.WriteLine(age);
```

Terminal – WorkingText

```
Invalid
Invalid
Invalid
Invalid
25
```

```
38
39              if (String.IsNullOrEmpty(" ".Trim()))
40                  Console.WriteLine("Invalid");
41
42              if (String.IsNullOrWhiteSpace(" "))
43                  Console.WriteLine("Invalid");
44
45              var str = "25";
46              var age = Convert.ToByte(str);
47              Console.WriteLine(age);
48
49              var price = 29.95f;
50              Console.WriteLine(price.ToString("C"));
51
52          }
```

Terminal – WorkingText

```
Invalid
Invalid
Invalid
25
$29.95
```

```
class Program
{
    static void Main(string[] args)
    {
        var sentence = "This is going to be a very long santance";

        if (sentence.Length < 20)
            Console.WriteLine(sentence);
        else
        {
            var words = sentence.Split(" ");
            var totalCharacters = 0;
            var summaryWords = new List<string>();

            foreach (var word in words)
            {
                summaryWords.Add(word);
                totalCharacters += word.Length + 1;
                if (totalCharacters > 20)
                    break;
            }
            var summary = String.Join(" ", summaryWords) + "... ";
            Console.WriteLine(summary);
        }
```

```
Terminal – SummaryText

This is going to be a...
```

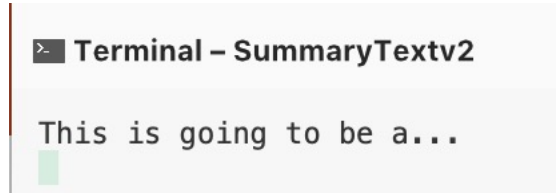
```cs
using System;
using System.Collections.Generic;

namespace SummaryTextv2
{
    class Program
    {
        static void Main(string[] args)
        {
            var sentence = "This is going to be a very long sentence";
            var summary = SummerizedText(sentence);
            Console.WriteLine(summary);
        }

        static string SummerizedText(string text, int maxLength = 20)
        {
            if (text.Length < maxLength)
                return (text);

            var words = text.Split(" ");
            var totalCharacters = 0;
            var summaryWords = new List<string>();

            foreach (var word in words)
            {
                summaryWords.Add(word);
                totalCharacters += word.Length + 1;
                if (totalCharacters > maxLength)
                    break;
            }
            return (String.Join(" ", summaryWords) + "... ");
        }
    }
}
```



```
using System;
using System.Collections.Generic;

namespace SummaryTextv3
{
    class Program
    {
        static void Main(string[] args)
        {
            var sentence = "This is going to be a very long sentence";
            var summary = StringUtility.SummerizedText(sentence);
            Console.WriteLine(summary);
        }


    }
}
```

```
using System;
using System.Collections.Generic;

namespace SummaryTextv3
{
    public class StringUtility
    {
        public static string SummerizedText(string text, int maxLength = 20)
        {
            if (text.Length < maxLength)
                return (text);

            var words = text.Split(" ");
            var totalCharacters = 0;
            var summaryWords = new List<string>();

            foreach (var word in words)
            {
                summaryWords.Add(word);
                totalCharacters += word.Length + 1;
                if (totalCharacters > maxLength)
                    break;
            }
            return (String.Join(" ", summaryWords) + "... ");
        }
    }
}
```