Working with Text

Introduction

These notes look at:

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

Case and Triming

ToLower()

ToUpper()

Trim()

Searching

IndexOF(‘a’)

LastIndexOf(‘hello’)

Substrings

Substring(startIndex)

Substring(startIndex, Length)

Replacing

Replace(‘a’, ‘!’)

Replace(“hello”,”goodbye”)

Null checking

String.IsNullOrEmpty(str)

String.IsNullOrWhiteSpace(str)

Splitting

str.Split(‘ ‘)

Converting Strings to Numbers

```
string s = “1234”

int i = int.Parse(s);

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

Converting Number to Strings

```
int i = 1234;

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

string t = i.ToString(“C”)                  //”$1,2,3,4.00”

string t = i.ToString(“C”)                  //”$1,2,3,4”
```

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

Exercise String

Type in the code →

```
using System;

namespace DemoString
{
    class Program
    {
        static void Main(string[] args)
        {
            //Trim
            var fullName = "John Smith ";
            Console.WriteLine("Trim: '{0}'", fullName.Trim());
            Console.WriteLine("Trim: '{0}'", fullName.Trim().ToUpper());

            //SubString
            var index = fullName.IndexOf(' ');
            var firstName = fullName.Substring(0, index);
            var lastName = fullName.Substring(index +1);
            Console.WriteLine("FirstName: " + firstName);
            Console.WriteLine("LastName: " + lastName);

            //Split
            var names = fullName.Split(' ');
            Console.WriteLine("FirstName: " +names[0]);
            Console.WriteLine("LastName: " + names[1]);

        }
    }
}
```

Exercise String

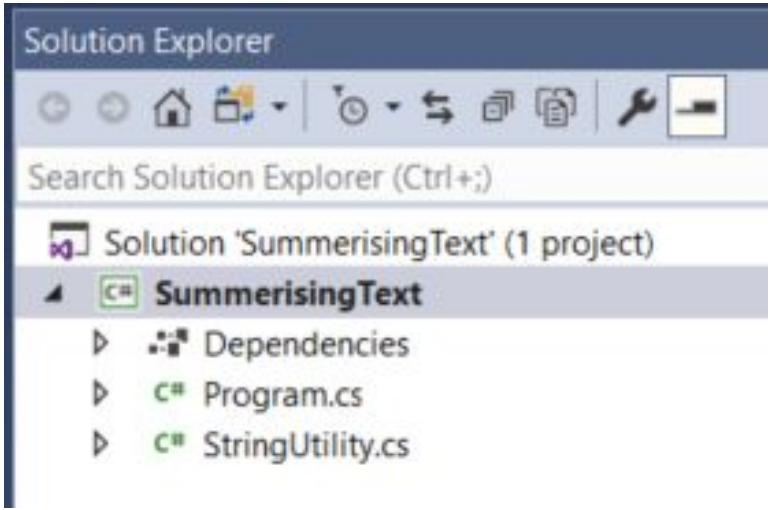
Type in the code →

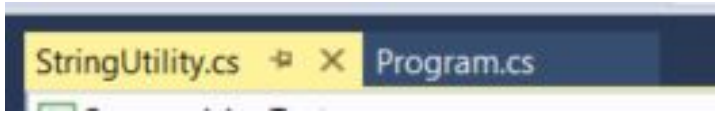
```
using System;

namespace DemoString
{
    class Program
    {
        static void Main(string[] args)
        {
            //Trim
            var fullName = "John Smith ";

            //Replace
            Console.WriteLine(fullName.Replace("John", "Johnothan"));

            //White space
            if (String.IsNullOrWhiteSpace(" "))
                Console.WriteLine("Invalid");

            //Covert string to Number
            var str = "25";
            var age = Convert.ToByte(str);

            //convert number to string
            float price = 29.95f;
            Console.WriteLine(price.ToString("C"));      //£0,000.00
            Console.WriteLine(price.ToString("C0"));     //£0,000

        }
    }
}
```

Mini Project - Exercise

This program will use two classes:

Create a Second Class called StringUtility





String Utility Class

Create the String Utility Class →

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


namespace SummerisingText
{
    public class StringUtility
    {
        public static string SummerizeText(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) + "...";
        }
    }
}
```

Program Class

Create the Program Class →

```
1   using SummerisingText;
2   using System;
3   using System.Collections.Generic;
4
5   namespace SummarisingText
6   {
7       class Program
8       {
9           static void Main(string[] args)
10          {
11              var sentence = "This is going to be a really really really really really long text.";
12              var summary = StringUtility.SummerizeText(sentence, 25);
13              Console.WriteLine(summary);
14
15          }
16
17      }
18  }
```

Stringbuilder

StringBuilder

StringBuilder is:

- Defined in System.Text
- A mutable string
- Easy and fast to create and manipulate strings

Not for Searching

It does not have:

IndexOf()

LastIndexOf()

Contains()

StartsWith()

...

String Manipulation Methods

It does have:

Append()

Insert()

Remove()

Replace()

Clear()

Exercise - StringBuilder

Type in the Code -->

```
using System;
using System.Text;

namespace StringyBuilder
{
    class Program
    {
        static void Main(string[] args)
        {
            var builder = new StringBuilder("Hello World");
            builder
                .Append('_', 10)
                .Append("Header")
                .AppendLine()
                .Append('_', 10)
                .Replace('-', '+')
                .Remove(0, 10)
                .Insert(0, new string('-', 10));

            Console.WriteLine(builder);
            Console.WriteLine("First Character: " + builder[0]);

        }
    }
}
```

Do Exercises

Working With Strings