Strings, characters, Regular expression,

Structures and numerations

Introduction to String

This chapter introduce the .Net framework class library's string and character – processing.

Fundamentals of characters and string

Characters are established according to the Unicode character set, an intranational characters set the ASCII.

Examples of string

" C# programming "

" (201) 555-21-21"

Input and read string

هناك طريق عديدة لقراءة واسناد القيم الى السرتنغ في المثال التالي عملية الادخال مباشرة من قبل المستخدم باستخدام (console.readline) وفي هذه الحالة لا نحتاج الى اي عملية تحويل (convert)

```
using System;
public class Exercise1
{
    public static void Main()
    {
        string str;

        Console.Write("\n\nAccept a string from keyboard :\n");
        Console.Write("-----------------------------------\n");
        Console.Write("Input the string : ");
        str = Console.ReadLine();
        Console.Write("The string you entered is : {0}\n", str);
        Console.ReadKey();
    }
}
```

Different ways to input an string

من الطرق الاخرى للاسناد : اما باسناد string مباشرة او عن طريق مسواة two string بحيث تاخذ الاولى قيمة الثانية او باستخدام المصفوفة وكما موضح في المثال التالي وكل طريقة تستخدم بحسب ما يحتاج اليه البرنامج.

```
using System;
namespace ConsoleApplication9
{
    class StringConstructore
    {
        static void Main(string[] args)
        {
            char[] charcterArray = { 'b', 'i', 'r', 't', 'h', ' ',
'd', 'a', 'y' };
            string origanalstring = "welcome to C# programming";
            string string1 = origanalstring ;
            string string2= new string (charcterArray);
            string string3 = new string(charcterArray, 6, 3);
            string string4 = new string('c', 5);
            Console.WriteLine("string 1 = " + string1 +"\n" +"string2
= "+ string2 +"\n" + "string3 = "+ string3 +"\n"+"string4 = "+
string4);
            Console.ReadKey();
        }
    }
}
```

String indexer, length property and copy to methods

استخدام ال stringاشبه باستخدام المصفوفة فلها نفس صفات المصفوفة من استخدام ال indexer وطول المصفوفة.

عند استخدام ال string نحتاج الى العديد من ال methodsمما يسهل علينا العمل والتعامل معها ومن هذه ال methods :

Copyto تستخدم لاخذ نسخة من string او مصفوفة ونسخها في string جديدة تتطلب تحديد موقع ومكان النسخ واين يتم النسخ ومن اي موقع كما في المثال.

```
using System;

namespace ConsoleApplication
{
    class StringMethods
    {
        static void Main(string[] args)
        {
            string string1 = "Hello there";
            char[] characterArray = new char[5];
            Console.WriteLine("string is"+ string1);
            Console.WriteLine("Length of string1:" + string1.Length);
            Console.Write("The string reversed is:");
            for (int i=string1.Length -1; i>=0; i--)
                Console.Write(string1[i]);
            string1.CopyTo(0, characterArray,
0,characterArray.Length);
            Console.Write ("\n The charcter array is:");
            for (int i = 0; i<characterArray.Length; i++)
                Console.Write (characterArray [i]);
            Console.WriteLine("\n");
            Console.ReadKey();
        }
    }
}
```

Comparing Strings

لنقوم بعملية المقاربة بين two string نستطيع استخدام احدى الطرق التالية:

1- equals وتستخدم لمعرفة هل ال two string متساويات ام لا ويكون ناتج المقارنة اما true or false.

2- "==" نفس عمل ال methods السابقة.

3- compare to تقارن بين two string لكن في هذه الحالة يقارن اما اكبر او اصغر او مستاويات ويكون الناتج اما -1 في حالة الاصغر او 0 في حالة المساواة و 1 كون القيمة الثانية اكبر.

```
using System;

namespace ConsoleApplication
{
    class StringCompare
    {
        static void Main(string[] args)
        {
            string string1 = "Hello ";
```

```
            string string2 = "good bye";
            string string3 = "Happy Birthday";
            string string4 = "happy birthday";

            Console.WriteLine("string1 is=" + string1 +"\n" + "string2
is=" + string2 + "\n" + "string3 is=" + string3 + "\n" + "string4 is="
+ string4);
            if (string1.Equals("Hello"))
                Console.Write("string 1 equals Hello");
            else
                Console.WriteLine("string 1 does not equal Hello");
            if (string1 == "hello")
                Console.Write("string 1 equals Hello");
            else
                Console.WriteLine("string 1 does not equal hello");
            if (string.Equals(string3,string4))
                Console.Write("string 3 equals string4");
            else
                Console.WriteLine("string 3 does not equal string4");
            Console.WriteLine("string1 compare to string2 is= "+
string1.CompareTo(string2)+ "\n"+
                "string2 compare to string3 is= "+
string2.CompareTo(string1)+ "\n"+
                "string1 compare to string1 is= "+
string1.CompareTo(string1)+ "\n"+
                "string3 compare to string4 is= "+
string3.CompareTo(string4)+ "\n"+
                "string4 compare to string3 is= "+
string4.CompareTo(string3)+"\n");
            Console.WriteLine("\n");
            Console.ReadKey();
        }
    }
}
```

Start with and End with

تسخدم هذه الدوال لمقارنة هل ال stringتبدأ بـ او تنهي بقيمة معينة وكما موضح في المثال:

```
using System;

namespace ConsoleApplication
{
    class StringCompare
    {
        static void Main(string[] args)
        {
            string [] strings ={ "started", "starting", "ended",
"ending"};
            for (int i = 0; i < strings.Length; i++)
                if (strings[i].StartsWith("st"))
                    Console.WriteLine(strings[i] +" start with st");
            for (int i = 0; i < strings.Length; i++)
                if (strings[i].EndsWith("ed"))
                    Console.WriteLine(strings[i] + " Ends with ed");


                Console.WriteLine("\n");
            Console.ReadKey();
        }
    }
}
```