

C# More fun with Strings

A long piece of text can be treated as a single String variable. This is handy for copying or writing data to file but if some useful processing is to be done on the String it may be necessary to manipulate the contents of a String. There are a number of built in routines that will help out here.

Splitting a String into smaller Strings

This can be used where a String contains data with some structure such as a gap between words, a line space between records in a file or a quotation between a set of data. This separating character (even the whitespace of a gap) is called a delimiter.

Consider the 3 lines below each being a different record

```
"'L008','3D Modelling Lightwave 6.0','Commercial','CD','DVC',1874.13,3.00,10  
'E004','A Level statistics','Educational','CD','The Times',11.48,1.00,3  
'U003','Adobe Acrobat 5.0','Utility','CD','Adobe',189.99,2.20,10"
```

For each the file structure is as follows, with a ',' separating the fields:

```
"'Product Number','Title','Class','Media','Publisher','Price','Carriage','Delivery (days)'"
```

The entire file could be a single String (shown here in ""). As a speech mark encloses a String this cannot be used with a list of records, the ' symbol is used instead.

Here Split is used to spilt apart the String word based on the delimiter ','

```
String word = "'L008','3D Modelling Lightwave  
6.0','Commercial','CD','DVC',1874.13,3.00,10";  
String [] items = word.Split(',');  
foreach (String x in items)  
{  
    Console.WriteLine(x);  
}
```

The following is the output to the Console:

```
'L008'  
'3D Modelling Lightwave 6.0'  
'Commercial'  
'CD'  
'DVC'  
1874.13
```

3.00

10

The quotation marks might need to be stripped out. The leading character is relatively easy but note that C# is using 3 symbols for quotations; ' , ' and ' .

Unfortunately the handy `foreach` loop does not allow the contents of the array to be easily modified so we need to fall back on a regular `for` loop.

```
for(int x =0;x<items.Length;x++)
{
    items[x]=items[x].Trim();
    if(items[x].StartsWith("'") ||items[x].StartsWith("'") )
    {
        items[x] = items[x].Substring(1, items[x].Length - 1);
    }
}
```

`Trim` is not strictly necessary but will remove any whitespace that has cropped in. `StartsWith` looks for a String that begins with another String. `Substring` will find a portion of a String and assign it to another or the same String.

The output will now be:

```
L008'
3D Modelling Lightwave 6.0
Commercial'
CD'
DVC'
1874.13
3.00
10
```

Previously `Substring` was used to remove the String that began at position 1 (the 2nd letter) and go along to the end. As the Strings are 0 based the number of characters to go along is `Length - 1`.

`Substring` can be used again to remove the last String character. Here it starts at the beginning (position 0) and ignores the last character (at length -1).

```
items[x] = items[x].Substring(0, items[x].Length - 1);
```

This will tidy up the output.

```
L008
3D Modelling Lightwave 6.0
Commercial
CD
DVC
```

1874.13
3.00
10

When the String is split into an Array the usual array methods such as sort and reverse can be applied to it.

It might also be useful to put the manipulated data back into a single String for storage or other processing. Luckily there is a routine called Join that will sort this out. This code will reverse the Array called items and put it back in a new String (not an array) with a ',' between each of the old array members.

```
Array.Reverse(items);  
String backwards = String.Join(",", items);  
Console.WriteLine(backwards);
```

The output is as follows:

10,3.00,1874.13,DVC,CD,Commercial,3D Modelling Lightwave 6.0,L008

Finding data with regular expressions

The contents of a String can be treated in different ways depending on the characters present. A file extension may identify an image that needs to be loaded or the @ symbol suggests the presence of an email.

Here a String contains some data that might be images.

```
"'L008','3D Modelling Lightwave  
6.0','Commercial.jpg','CD','logo.jpg',1874.13,3.00,10"
```

To use regular expressions the following using is required.

```
using System.Text.RegularExpressions;
```

This code will look for a match with ".jpg".

```
Regex re = new Regex(".jpg");  
foreach(String x in items) //items is an array of Strings  
{  
    if (re.Match(x).Success)  
        Console.WriteLine(x + " is a picture");  
}
```

There are several picture formats. Regular expressions can handle this with a | symbol.

```
Regex re = new Regex(".jpg|.png|.bmp");
```

They can also account for the text being in upper or lower case.

```
Regex re = new Regex(".jpg|.png|.bmp", RegexOptions.IgnoreCase);
```

A regular expression can search for patterns within words. The following rules are used below:

- `^` String begins with
- `$` String ends with
- `+` a sequence of 1 or more characters

The following will search for a whole number only;

```
Regex re = new Regex("[0-9]+$");
```

This will also include numbers with a decimal;

```
Regex re = new Regex("[0-9.]+$");
```

Here are some more examples

- Begins with L followed by any characters `"^L"`
- Begins with 4 digits `"^\\d{4}"`
- Ends in .com `".com$"`
- Begins with www. and ends with .com `"^www.*.com$"`
- Begins with http or ftp then continues www. ending in .com `"^(http|ftp)://www.*.com$"`

Regular expressions can also be used to replace a sequence found within the regular expression with another specified String. The following code will replace a .com domain name with a .co.uk name.

```
String url = "www.address.com";  
re = new Regex(".com$");  
url = re.Replace(url, ".co.uk");  
Console.WriteLine(url);
```

Practice

1. A file contains information on customer data. Luckily it is well organised. Display the first and last names (only) of the customers. The following data can be used to test the program. Split will sort out the String into an Array. The whole String could be processed 1 line at a time giving a different Array for each customer. Another approach would be to create 1 long Array for the whole file and examine the positions of the elements that represent names, there will be a pattern to where these crop up.

'Mr','Frank','Streeter','78 Fratton Way','Perth','BD9
6ZZ','13/08/1990','M','J',1021,,1,'B'
'Ms','Heena','Shah','127 Lonner Street','Molden','BD3
6AZ','17/07/1979','F','L',1022,'JohnsonJ',0,'I'
'Mr','William','Ponsonby','26 Railway Street','Highcliffe','BD6
2HF','16/12/1978','M','M',1023,'DouglasS',1,'A'
'Mr','Mark','Pratney','45 Stoddard Way','Wendhoven','DR3
4GH','02/03/1971','M','M',1024,,1,'I'
'Mrs','Sally','Kent','34 Way Street','Dratton','DR4 8JK',24/09/1978,'F','L',1025,,0,'A'
'Mr','Thomas','Pranney','56 Franconia Way','Molden','BD3
5TH','05/06/1967','M','M',1026,'JohnsonJ',0,'I'
'Mr','David','Pratton','721 Merther Road','Highcliffe','BD6
8LS','09/07/1978','M','M',1027,'DouglasS',1,'I'
'Mr','David','Pratter','34 Marks Way','Molden','BD3
6GC','04/06/1958','M','M',1028,,0,'A'
'Miss','Sarah','Fertgin','113 Leighton Avenue','Molden','BD3
2DS','18/03/1963','F','L',1029,,1,'A'

2. You are writing a text adventure game in homage to Zork. You want the player to enter the direction they are to walk. Detect if the user wants to go West and make a suitable confirmation. The user might type in West, west, W, turn West, go west or any other reasonable input. Select the key elements of the String input and act on it.
3. Your web bot has harvested some text. You would like to pull out the emails in it with a computer program. These can later be used for 'marketing' purposes. The String can be pulled apart with Split based on the gaps between words. Some of the resulting Array members will still end in a ',', ';' or '.' that will need sorting out. Each member of the Array can then be tested with a regular expression.

I was born in the year 1632, in the city of York, www.york.gov.uk, of a good family, though not of that country, my father, dadcruoe@yahoo.com being a foreigner of Bremen, www.bremen.de, who settled first at Hull. He got a good estate by merchandise, and leaving off his trade, lived afterwards at York, from whence he had married my mother, mumcruoe@plusnet.co.uk, whose relations were named Robinson, robinsonfamily@msn.com, a very good family in that country, and from whom I was called Robinson Kreutznaer; Kreutznaer@talktalk.co.uk but, by the usual corruption of words in England, we are now called—nay we call ourselves and write our name—Crusoe, crusoefamily@hotmail.com; and so my companions always called me.

4. In the following text the author has made some attempt to hide email addresses. They should not be too hard to pick out. For bonus points the Regex replace method could be used to convert the results back to well formatted addresses.

If I may believe the woman whom I was taught to call mother, masingletonatyahoodotcom I was a little boy, of about two years old, innocentchildathotmaildotcom, very well dressed, had a nursery-maid maidsingletonatmsndotcom, to attend me, who took me out on a fine summer's

evening into the fields towards Islington, as she pretended, to give the child some air; a little girl being with her, of twelve or fourteen years old, that lived in the neighbourhood, little girl at talk dot codot uk. The maid, whether by appointment or otherwise, meets with a fellow, her sweetheart, as I suppose; he carries her into a public-house, to give her a pot and a cake; and while they were toying in the house the girl plays about, with me in her hand, in the garden and at the door, sometimes in sight, sometimes out of sight, thinking no harm.

At this juncture comes by one of those sort of people, very bad man at child salvery dot codot jm who, it seems, made it their business to spirit away little children. This was a hellish trade in those days, and chiefly practised where they found little children very well dressed, or for bigger children, to sell them to the plantations.