

# string类型的各种方法

## 一.方法介绍

### 1.stringBuilder

**注：**StringBuilder类位于System.Text库中

**注：**Stopwatch类位于System.Diagnostics库中

```
StringBuilder sb = new StringBuilder();
sb.Append();    //用于拼接StringBuilder

//创建一个计时器，用来记录程序运行的时间(位于System.Diagnostics库之中)
Stopwatch sw = new Stopwatch();
sw.Start();    //开始计时
for(int i = 0; i < 1000000; i++){
    sb.Append(i);
}
sw.Stop();    //结束计时
System.Console.WriteLine("共用时: " + sw.Elapsed);
```

### 2.字符串的比较

```
/** 普通方法:
 * ToUpper()将字符串转换成大写
 * ToLower()将字符串转换成大写
 */
/*常用简易方法*/
str1.Equals()    //用来比较字符串是否相同
//StringComparison : 枚举类型参数, 用于指定将如何去比较字符串
//OrdinalIgnoreCase : 忽略字符串的大小写进行比较
str1.Equals(str2, StringComparison.OrdinalIgnoreCase)
//例如:
if(lessonOne.Equals(lessonTwo, StringComparison.OrdinalIgnoreCase){
    System.Console.WriteLine("lessonOne与lessonTwo相同");
}
```

### 3.字符串的分割

```
/**使用Split分割字符串
 * 注: Split会将字符串分割成多个单字符。(返回的是字符串数组string[])
 * 参数chs: 字符数组, Split会对照本数组中的元素, 将原字符串中的对应字符替换成null
 * 参数StringSplitOptions: 枚举类型, 字符串分割选项
 * StringSplitOptions.None : 返回的值中包含null项
 * StringSplitOptions.RemoveEmptyEntries : 移除null项
```

```

*/
string s = "aa b    cde _ + - =,,, _fgh";
char[] chs = {' ', '_', '+', '-', '=', ',', ' '};
string[] str1 = s.Split(chs, StringSplitOptions.None);    //包含null项
string[] str2 = s.Split(chs, StringSplitOptions.RemoveEmptyEntries);    //去除
null项

```

#### 4.字符串的包含和替换

```

//Contains(string value) : 判断字符串中是否含有子串value (返回bool型)
//Replace(string oldValue, string newValue) : 将字符串中所有oldValue替换成
newValue
string str = "国家关键人物小何";
if(str.Contains("小何")){
    str = str.Replace("小何", "****");
}
System.Console.WriteLine(str);

```

#### 5.字符串的开始、结束判断

```

//StartsWith(string value) : 判断字符串是否以子串Value开始 (返回bool型)
//EndsWith(string value) : 判断字符串是否以子串value结束 (返回bool型)
string str2 = "2022年8月16日11:55,今天是我的生日";
if(str2.StartsWith("2022")){
    System.Console.WriteLine("是的");
}else{
    System.Console.WriteLine("不是的");
}
if(str2.EndsWith("日")){
    System.Console.WriteLine("是的");
}else{
    System.Console.WriteLine("不是的");
}

```

#### 6.字符串的截取

```

//Substring(string value) : 截取字符串 (返回bool型)
string weather = "今天天气好晴朗, 处处好风光";
string tempStr = " ";
tempStr = weather.Substring(2);    //从字符串第2位开始往后截取
System.Console.WriteLine(tempStr);
tempStr = weather.Substring(2, 7);    //从第2位开始截取, 一共截取7个字符
System.Console.WriteLine(tempStr);

```

## 7.查找字符串中某子串出现的位置

```
//IndexOf(string value) : 取子串value第一次出现的位置 (返回int型)
//LastIndexOf(string value) : 取子串value最后一次出现的位置 (返回int型)
string Birthday = "2022年8月16日11:55,今天是我的生日";
System.Console.WriteLine("\"生日\"这个词出现在句子中的第" + Birthday.IndexOf("生日") + "号位置上");
System.Console.WriteLine("2最后一次出现在第" + Birthday.LastIndexOf("2") + "");
```

## 8.截取与查找位置的联合应用

```
//IndexOf 经常和 Substring联合使用,如:
string Path = @"c:\a\b\隐藏文件夹\家传古画\hch\唐三藏赤壁拳打王熙凤.jpg";
//@的作用 : 取消反斜杠'\ '的转义作用,因此本句中不需要连用两个\\

int index = Path.LastIndexOf(@"\"); //找到最后一个 \ 的
Path = Path.Substring(index + 1); //截取 \ 之后的文件名
System.Console.WriteLine(Path);
```

## 9.字符串去空格

```
//字符串去空格 只能去除前后空格,不能去除字符串中间的空格
string spaceStr = " hahaa haha ";
spaceStr = spaceStr.Trim();
spaceStr = spaceStr.TrimStart(); //只去除字符串前面的空格
spaceStr = spaceStr.TrimEnd(); //只去除字符串后面的空格
System.Console.WriteLine(spaceStr);
```

## 10.字符串null与空判断

```
//判断字符串是否为null或者空
string determineStr1 = null;
if(string.IsNullOrEmpty(determineStr1)){
    System.Console.WriteLine("该字符串为null或者空");
}else{
    System.Console.WriteLine("该字符串既不为null也不为空");
}
```

## 11.串联字符串数组中的元素,并插入指定字符

```
/**
 * Join(string separator,params object[] value)
 * params : 可变参数,说明既可以放数组也可以放元素
```

```

* object[] : 说明不局限于string数组
*/
string[] names = {"张三","李四","王五","赵六","牛七",};
string nweStr = string.Join("|",names);    //在每个元素中插入 | 并用一个新字符串
接收
System.Console.WriteLine(nweStr);

```

## 12.字符串与字符数组相互转换

```

//将字符串的每个字符逐一转换成字符数组中的每个元素
string str = "张三李四王麻子";
char[] chs = str.ToCharArray();

//将字符数组的每个元素合并，转换成一个字符串
string str2 = new string(chs);

```

## 二.练习

### 1.读取修改文件中的书名和作者名

**注：File类位于System.IO库中**

```

//练习：文本文件中储存了多个文章标题、作者
//标题和作者之间用若干空格隔开，每行一个
//标题输出到控制台时最长10个字符
//如果标题超过10字符，则截取前8个字符并在后面加上".....",并加一个竖线后输出作者名字
string filePath = @"E:\Code\C# for VS_Code\string类型练习\文件\书名和作者.txt";
//取到文件路径
string[] arrayContents = File.ReadAllLines(filePath);    //根据路径找到文
档，并逐行读取数据存入string数组中
char[] chs = {' '};    //准备根据空格将书名和作者名分隔开
for(int i = 0;i < arrayContents.Length;i++){
    string[] newStr =
arrayContents[i].Split(chs,StringSplitOptions.RemoveEmptyEntries);    //分割书名和
作者名，去除null项
    if(newStr[0].Length >= 10){
        string tempStr = newStr[0].Substring(0,8);    //若书名超长则截取
前八个字符
        System.Console.WriteLine(tempStr + "....." + "|" + newStr[1]);
    }else{
        System.Console.WriteLine(newStr[0] + "|" + newStr[1]);
    }
    System.Console.WriteLine((newStr[0].Length >= 10 ?
newStr[0].Substring(0,8) + "....." : newStr[0]) + "|" + newStr[1]) ;
}

```

### 2.让用户输入一句话，找出所有e的位置

```
string str = "abcdefefffebglhesefasecb";
int index = 0;
int count = 0;
while(index != -1){
    count++;
    index = str.IndexOf("ef",index + 1);
    if(index == -1){
        break;
    }
    System.Console.WriteLine("第" + count + "次出现ef的位置是:" + index);
}
```

### 3.提取数组中的元素并插入|,然后再删掉|变回数组

```
string[] names = {"诸葛亮","鸟叔","卡卡西","卡哇伊"};
string str = string.Join("|",names);
string[] newArray = str.Split('|',StringSplitOptions.RemoveEmptyEntries);
for(int i = 0;i <= newArray.Length;i++){
    System.Console.WriteLine(newArray[i]);
}
```