

C#学习心得

版本：v2.0

Crifan Li

摘要

本文主要介绍了C#中的一些学习心得，包括但不限于Regex,TreeView,字典类型变量,sprintf,UrlEncode等



本文提供多种格式供：

在线阅读	HTML ¹	HTMLs ²	PDF ³	CHM ⁴	TXT ⁵	RTF ⁶	WEBHELP ⁷
下载（7zip压缩包）	HTML ⁸	HTMLs ⁹	PDF ¹⁰	CHM ¹¹	TXT ¹²	RTF ¹³	WEBHELP ¹⁴

HTML版本的在线地址为：

http://www.crifan.com/files/doc/docbook/csharp_summary/release/html/csharp_summary.html

有任何意见，建议，提交bug等，都欢迎去讨论组发帖讨论：

http://www.crifan.com/bbs/categories/csharp_summary/

修订历史

修订 2.0	2013-08-20	crl
1. 将crifanLib.cs弄成独立的book 2. 整理更多的关于C#的心得		
修订 1.0	2012-11-06	crl
1. 将C#从language_summary移至此独立book 2. 更新crifanLib.cs的最新链接		

¹ http://www.crifan.com/files/doc/docbook/csharp_summary/release/html/csharp_summary.html

² http://www.crifan.com/files/doc/docbook/csharp_summary/release/htmls/index.html

³ http://www.crifan.com/files/doc/docbook/csharp_summary/release/pdf/csharp_summary.pdf

⁴ http://www.crifan.com/files/doc/docbook/csharp_summary/release/chm/csharp_summary.chm

⁵ http://www.crifan.com/files/doc/docbook/csharp_summary/release/txt/csharp_summary.txt

⁶ http://www.crifan.com/files/doc/docbook/csharp_summary/release/rtf/csharp_summary.rtf

⁷ http://www.crifan.com/files/doc/docbook/csharp_summary/release/webhelp/index.html

⁸ http://www.crifan.com/files/doc/docbook/csharp_summary/release/html/csharp_summary.html.7z

⁹ http://www.crifan.com/files/doc/docbook/csharp_summary/release/htmls/index.html.7z

¹⁰ http://www.crifan.com/files/doc/docbook/csharp_summary/release/pdf/csharp_summary.pdf.7z

¹¹ http://www.crifan.com/files/doc/docbook/csharp_summary/release/chm/csharp_summary.chm.7z

¹² http://www.crifan.com/files/doc/docbook/csharp_summary/release/txt/csharp_summary.txt.7z

¹³ http://www.crifan.com/files/doc/docbook/csharp_summary/release/rtf/csharp_summary.rtf.7z

¹⁴ http://www.crifan.com/files/doc/docbook/csharp_summary/release/webhelp/csharp_summary.webhelp.7z

C#学习心得:

Crifan Li

版本 : v2.0

出版日期 2013-08-20

版权 © 2013 Crifan, <http://crifan.com>

本文章遵从 : [署名-非商业性使用 2.5 中国大陆\(CC BY-NC 2.5\)](#)¹⁵

¹⁵ http://www.crifan.com/files/doc/docbook/soft_dev_basic/release/html/soft_dev_basic.html#cc_by_nc

目录

前言	iv
1. 本文目的	iv
2. 待完成	iv
1. C#学习心得	1
1.1. C#中的字典类型变量	1
1.2. C#中关于字符串的处理	1
1.2.1. C#中类似于sprintf的函数	1
1.2.2. 字符串的中间对齐左右填充的效果	2
1.3. C#中ComboBox的使用	2
1.3.1. 给ComboBox设置数据源	2
2. C#的网络相关处理	4
2.1. C#中HTTP处理	4
2.1.1. C#中自动处理返回的压缩的HTML	4
2.1.2. C#中给HTTP添加代理Proxy支持	4
2.1.3. ReadLine或ReadToEnd会无限期挂掉	4
2.2. C#中处理cookie	5
2.2.1. C#中自动处理cookie	5
2.2.2. SetCookie解析有bug	5
2.3. C#中处理URL	5
2.3.1. 关于UrlEncode把空格编码为加号" + "而不是%20的问题	5
2.4. C#中的WebBrowser	5
2.4.1. DocumentCompleted被调用两次	6
3. C#的处理Excel和CSV	7
3.1. C#中处理Excel	7
3.1.1. Excel自动适应列宽	7
3.1.2. 选中Excel的某列后再去自动适应列宽	7
3.1.3. Could not load file or assembly 'Microsoft.Office.Interop.Excel'	7
3.1.4. Microsoft.Vbe.Interop.dll和office.dll	8
3.2. C#中处理CSV	8
4. C#的TreeView控件使用心得	10
4.1. 添加Node的方法	10
4.2. 给TreeView添加图标	10
4.3. 获得右击所在节点	10
4.4. 添加右键菜单 (Context Menu)	11
4.5. 给TreeNode添加名字	11
参考书目	12

前言

1. 本文目的

本文目的在于，把之前自己折腾C#的心得和注意实现都整理出来，供其他人参考。

2. 待完成

目标只整理了到了page 2：

http://www.crifan.com/category/work_and_job/programming_language/csharp/page/2/

之后的，有空再整理到此book中。

第 1 章 C#学习心得

1.1. C#中的字典类型变量

C#中也有类似于Python中的字典类型的变量

参考代码如下：

```
Dictionary<String, int> dicPicIdx;  
Dictionary<string, int>.KeyCollection dicKeys;  
  
dicPicIdx = new Dictionary<string, int>();  
dicPicIdx.Add("EmptyDocumentFolder", 1);  
dicPicIdx.Add("NonEmptyDocumentFolder", 1);  
  
dicPicIdx.Add("NonEmptyAlbum", 3);  
dicPicIdx.Add("EmptyAlbum", 3);  
  
dicPicIdx.Add("NonEmptyFavoriteFolder", 4);  
dicPicIdx.Add("EmptyFavoriteFolder", 4);  
  
dicPicIdx.Add("Photo", 6);  
dicPicIdx.Add("Audio", 7);  
dicPicIdx.Add("Video", 8);  
  
dicKeys = dicPicIdx.Keys;
```

1.2. C#中关于字符串的处理

1.2.1. C#中类似于sprintf的函数

C#中，类似于C中常用的sprintf函数，是String.Format函数。

最简单的用法举例如下：

```
string sprintfTestStr = String.Format("Test sprintf in C#, number={0:D}, string=\"{1:s}\",  
float={2:0.000}", 100, "crifan", Math.PI);  
//sprintfTestStr = Test sprintf in C#, number=100, string="crifan", float=3.142
```

关于Format函数的更多的示例，可以参考微软官方文档：[String.Format Method \(String, Object\)](http://msdn.microsoft.com/en-US/library/fht0f5be(v=vs.80).aspx)¹

关于其他更多不同类型的参数，比如日期，数值，枚举等，如何指定对应的格式，可以参考：

- For more information about the composite formatting feature supported by methods such as [Format](#)², [AppendFormat](#)³, and some overloads of [WriteLine](#)⁴, see [Composite Formatting](#)⁵.

¹ [http://msdn.microsoft.com/en-US/library/fht0f5be\(v=vs.80\).aspx](http://msdn.microsoft.com/en-US/library/fht0f5be(v=vs.80).aspx)

² [http://msdn.microsoft.com/en-US/library/system.string.format\(v=vs.80\).aspx](http://msdn.microsoft.com/en-US/library/system.string.format(v=vs.80).aspx)

³ [http://msdn.microsoft.com/en-US/library/system.text.stringbuilder.appendformat\(v=vs.80\).aspx](http://msdn.microsoft.com/en-US/library/system.text.stringbuilder.appendformat(v=vs.80).aspx)

- For more information about numeric format specifiers, see [Standard Numeric Format Strings](#)⁶ and [Custom Numeric Format Strings](#)⁷.
- For more information about date and time format specifiers, see [Standard DateTime Format Strings](#)⁸ and [Custom DateTime Format Strings](#)⁹.
- For more information about enumeration format specifiers, see [Enumeration Format Strings](#)¹⁰.
- For more information about formatting, see [Formatting Types](#)¹¹ and [Formatting Overview](#)¹².

1.2.2. 字符串的中间对齐左右填充的效果

代码：

```
//input: [4] Valid: B0009IQZFM
//output: ===== [4] Valid: B0009IQZFM
=====
public string formatString(string strToFormat, char cPaddingChar = '*', int iTotalWidth = 80)
{
    //auto added space
    strToFormat = " " + strToFormat + " "; //" [4] Valid: B0009IQZFM "

    //1. padding left
    int iPaddingLen = (iTotalWidth - strToFormat.Length)/2;
    int iLefTotalLen = iPaddingLen + strToFormat.Length;
    string strLefPadded = strToFormat.PadLeft(iLefTotalLen,
cPaddingChar); //"===== [4] Valid: B0009IQZFM "
    //2. padding right
    string strFormatted = strLefPadded.PadRight(iTotalWidth,
cPaddingChar); //"===== [4] Valid: B0009IQZFM
===== "

    return strFormatted;
}
```

详见：[【已解决】C#中实现字符串的中间对齐左右填充的效果](#)¹³

1.3. C#中ComboBox的使用

1.3.1. 给ComboBox设置数据源

主要包含三步：

⁴ [http://msdn.microsoft.com/en-US/library/system.console.writeline\(v=vs.80\).aspx](http://msdn.microsoft.com/en-US/library/system.console.writeline(v=vs.80).aspx)
⁵ [http://msdn.microsoft.com/en-US/library/txafckwd\(v=vs.80\).aspx](http://msdn.microsoft.com/en-US/library/txafckwd(v=vs.80).aspx)
⁶ [http://msdn.microsoft.com/en-US/library/dwhawy9k\(v=vs.80\).aspx](http://msdn.microsoft.com/en-US/library/dwhawy9k(v=vs.80).aspx)
⁷ [http://msdn.microsoft.com/en-US/library/0c899ak8\(v=vs.80\).aspx](http://msdn.microsoft.com/en-US/library/0c899ak8(v=vs.80).aspx)
⁸ [http://msdn.microsoft.com/en-US/library/az4se3k1\(v=vs.80\).aspx](http://msdn.microsoft.com/en-US/library/az4se3k1(v=vs.80).aspx)
⁹ [http://msdn.microsoft.com/en-US/library/8kb3ddd4\(v=vs.80\).aspx](http://msdn.microsoft.com/en-US/library/8kb3ddd4(v=vs.80).aspx)
¹⁰ [http://msdn.microsoft.com/en-US/library/c3s1ez6e\(v=vs.80\).aspx](http://msdn.microsoft.com/en-US/library/c3s1ez6e(v=vs.80).aspx)
¹¹ [http://msdn.microsoft.com/en-US/library/fbxft59x\(v=vs.80\).aspx](http://msdn.microsoft.com/en-US/library/fbxft59x(v=vs.80).aspx)
¹² [http://msdn.microsoft.com/en-US/library/26etazsy\(v=vs.80\).aspx](http://msdn.microsoft.com/en-US/library/26etazsy(v=vs.80).aspx)
¹³ http://www.crifan.com/csharp_string_align_center_padding_left_and_right/

1. 设置结构体，其中的field的名字首字母大写，支持get和set

```
private struct keyValueList
{
    public string Key{get;set;} // key
    public List<string> ValueStrList{get;set;} // the string value list for the key
}
```

2. 创建对应的结构体数组

```
List<keyValueList> gFootprintTypeSelList; // footprint type

gFootprintTypeSelList = new List<keyValueList>();

//1. option1: Guest Blogging
keyValueList keyValueListGuestBlogging = new keyValueList();
keyValueListGuestBlogging.Key = "Guest Blogging";
keyValueListGuestBlogging.ValueStrList = new List<string>();
//keyValueListGuestBlogging.ValueStrList.Add("Specify Footprint");
keyValueListGuestBlogging.ValueStrList.Add("Guest Blogging");
keyValueListGuestBlogging.ValueStrList.Add("Contribute");
keyValueListGuestBlogging.ValueStrList.Add("Write for us");
keyValueListGuestBlogging.ValueStrList.Add("Guest Category");
keyValueListGuestBlogging.ValueStrList.Add("Submit Content");
gFootprintTypeSelList.Add(keyValueListGuestBlogging);
```

3. 把结构体数组赋值给ComboBox的DataSource，把DisplayMember设置为对应的结构体的field的名字（全小写）

```
cmbFootprintType.DataSource = gFootprintTypeSelList;
cmbFootprintType.DisplayMember = "key";
```

详见：[【已解决】C#中给ComboBox设置数据源](http://www.crifan.com/csharp_combobox_set_data_source/)¹⁴

¹⁴ http://www.crifan.com/csharp_combobox_set_data_source/

第 2 章 C#的网络相关处理

2.1. C#中HTTP处理

2.1.1. C#中自动处理返回的压缩的HTML

当你处理http时，返回的html的header中包含：

```
Content-Encoding: deflate
```

或

```
Content-Encoding: gzip
```

时，说明是返回的是压缩的HTML。

想要支持压缩的html的话，C#中，在发送HttpRequest之前，设置AutomaticDecompression为对应的参数即可：

```
HttpRequest req = (HttpRequest)WebRequest.Create(url);  
  
.....  
  
req.Headers["Accept-Encoding"] = "gzip, deflate";  
//req.AutomaticDecompression = DecompressionMethods.GZip;  
req.AutomaticDecompression = DecompressionMethods.GZip |  
    DecompressionMethods.Deflate;
```

更多解释详见：

[【已解决】C#中HttpRequest使用Proxy后异常](#)¹

2.1.2. C#中给HTTP添加代理Proxy支持

示例代码：

```
using System.Net;  
WebProxy gProxy = new WebProxy("127.0.0.1", 8087);  
HttpRequest req = (HttpRequest)WebRequest.Create(url);  
req.Proxy = gProxy;
```

详见：[【已解决】C#中HttpRequest支持代理](#)²

2.1.3. ReadLine或ReadToEnd会无限期挂掉

在用C#的GetResponseStream，常会遇到，ReadLine或ReadToEnd会无限期挂掉，所以，需要去加上对应的超时（timeout）设置：

¹ http://www.crifan.com/csharp_httpwebrequest_proxy_abnormal_not_working/

² http://www.crifan.com/csharp_add_proxy_for_httpwebrequest/

详见：[【已解决】C#中在GetResponseStream得到的Stream后,通过StreamReader去ReadLine或ReadToEnd会无限期挂掉 + 给StreamReader添加Timeout支持](#)³

2.2. C#中处理cookie

2.2.1. C#中自动处理cookie

[【整理】C#中如何自动处理cookie](#)⁴

2.2.2. SetCookie解析有bug

C#中，关于解析SetCookie，前后遇到多个bug，详见：

[【经验记录】C#中，库函数有bug，会将http所返回的response响应中的headers头信息中的Set-Cookie值，解析错误，丢失部分cookie](#)⁵

[【已解决】又发现一个C#中解析Set-Cookie的一个bug：无故地添加cookie的path域](#)⁶

2.3. C#中处理URL

2.3.1. 关于UrlEncode把空格编码为加号" + "而不是%20的问题

默认的HttpUtility.UrlEncode会把空格编码为加号，而不是很多人所期望的%20，导致很多人以为函数出了问题呢。其实，微软官方文档[HttpUtility.UrlEncode方法](#)⁷中，已经解释了此问题：

您可以使用 UrlEncode 方法或 UriPathEncode 方法对 URL 编码。

但是，方法返回不同的结果。UrlEncode 方法将每个空格字符转换为加号 (+) 字符。

UriPathEncode 方法将每个空格字符转换为字符串 "%20"，它表示一个用十六进制表示法表示的空格。

在对 URL 的路径部分编码时使用 UriPathEncode 方法，以保证一致的已解码 URL，与执行解码的平台或浏览器无关。

所以，想要把空格编码为%20，用 UriPathEncode 即可。

2.4. C#中的WebBrowser

C#中使用WebBrowser，相对还是很简单的，比如，打开网页，直接用uri即可：

```
wbsChaseFootprint.Url = new Uri(strEncodedFullFootprintUrl);
```

³ http://www.crifan.com/csharp_httpwebresponse_streamreader_readline_dead_again/

⁴ http://www.crifan.com/csharp_auto_handle_cookie/

⁵ http://www.crifan.com/csharp_parse_response_cookie_uncorrect/

⁶ http://www.crifan.com/csharp_another_set_cookie_parse_bug_add_wrong_path/

⁷ <http://msdn.microsoft.com/zh-cn/library/4fkewx0t.aspx>

详见：[【记录】C#中使用WebBrowser浏览google页面](#)⁸

2.4.1. DocumentCompleted被调用两次

目前的解决办法，在DocumentCompleted中加上：

```
if (!e.Url.Equals(wbsChaseFootprint.Url))
{
    //not actually complete, do nothing
    return;
}
```

详见：[【已解决】C#中的WebBrowser的DocumentCompleted被调用两次](#)⁹

⁸ http://www.crifan.com/record_csharp_use_webbrowser_to_browse_google/

⁹ http://www.crifan.com/csharp_webbrowser_documentcompleted_multiple_time_called/

第 3 章 C#的处理Excel和CSV

3.1. C#中处理Excel

3.1.1. Excel自动适应列宽

全选所有列再调用AutoFit：

```
//auto adjust column width (according to content)
Range allColumn = xlWorkSheet.Columns;
allColumn.AutoFit();
```

详见：[【已解决】C#中操作刚导出的Excel，设置其为自动调整列宽](#)¹

3.1.2. 选中Excel的某列后再去自动适应列宽

核心逻辑：

获得对应的列的Range，再去AutoFit

```
Range firstColumn = xlWorkSheet.get_Range("A1");
firstColumn.EntireColumn.AutoFit();
```

详见：[【已解决】C#中操作Excel文件实现单行的自动适应列宽+C#中如何选中Excel的某列](#)²

3.1.3. Could not load file or assembly 'Microsoft.Office.Interop.Excel'

使用C#操作Excel常会出现类似错误：

Could not load file or assembly 'Microsoft.Office.Interop.Excel, Version=14.0.0.0

其可能原因，现总结如下：

C#的exe用到了excel的话，希望拿到别人的地方也正常运行不报错的话，

其中，别的地方：

- 可能没装excel
- 也可能装了excel，但是版本低

那么有几种选择：

1. 最好自己在exe中集成了对应的excel的dll库。
关于，集成dll到exe中，简述为：

¹ http://www.crifan.com/csharp_excel_auto_adjust_column_width_autofit/

² http://www.crifan.com/csharp_excel_auto_adjust_column_width_and_howto_select_single_column/

1. 把dll加到自己的资源resource中
2. 把dll加到自己的项目中，且属性设置为 嵌入的资源 (Embedded Resource)
3. 自己的类的初始化函数中，加上对应的load dll的相关代码

详见：[【已解决】C#中集成DLL库到自己的exe程序中³](#)

2. 即使不集成excel的dll，在引用excel的dll

1. 也要尽量引用低版本
比如别人已装的excel，即Office是Office12

那么你也就不要去引用Office14==Office2010：

```
C:\Program Files (x86)\Microsoft Visual Studio 10.0\Visual Studio Tools for Office\PIA\Office14\Microsoft.Office.Interop.Excel.dll
```

了，而去引用Office12==Office2007：

```
C:\Program Files (x86)\Microsoft Visual Studio 10.0\Visual Studio Tools for Office\PIA\Office12\Microsoft.Office.Interop.Excel.dll
```

2. 或者是和对方电脑中安装的excel的版本一致
当然，最好确认一下，对方的电脑中装了哪个版本的Office（的excel）

比如是Office 2010==Office14，那么你也就去引用对应的Office14：

```
C:\Program Files (x86)\Microsoft Visual Studio 10.0\Visual Studio Tools for Office\PIA\Office14\Microsoft.Office.Interop.Excel.dll
```

就好了。

详见：[【已解决】虽然已经安装了Office的Excel但是C#的exe还是运行出错：Could not load file or assembly 'Microsoft.Office.Interop.Excel, Version=14.0.0.0⁴](#)

3.1.4. Microsoft.Vbe.Interop.dll和office.dll

Microsoft.Vbe.Interop.dll和office.dll，与Microsoft.Office.Interop.Excel.dll，一起，都是属于“Microsoft Office system 的可用程序集”

所以，最好也是要在集成，Microsoft.Office.Interop.Excel.dll，时，连带的一起把：Microsoft.Vbe.Interop.dll和office.dll都集成进来。

这样，才可以避免，别人在使用exe时，内部用到excel的dll时，完整的所需要的函数，都可以自带的找到了。

不会再对于Microsoft.Vbe.Interop.dll或office.dll报错说找不到。

详见：[【已解决】Microsoft.Vbe.Interop.dll和office.dll是啥⁵](#)

3.2. C#中处理CSV

³ http://www.crifan.com/csharp_integrate_dll_library_into_exe_application

⁴ http://www.crifan.com/csharp_could_not_load_file_or_assembly_microsoft_office_interop_excel_version_14/

⁵ http://www.crifan.com/what_is_microsoft_vbe_interop_dll_and_office_dll/

第 4 章 C#的TreeView控件使用心得

4.1. 添加Node的方法

当前已有一个TreeView控件trvFolder，添加节点的代码如下：

```
TreeNode curNode = new TreeNode(name);
trvFolder.Nodes.Add(curNode);
```

4.2. 给TreeView添加图标

先拖拽出来一个ImageList控件iglIcons，然后手动编辑ImageList的Images属性Collection，添加几个小图片。

对应的index分别是0,1,2,...，然后用：

```
trvFolder.ImageList = iglIcons;
```

将其关联到当前TreeView控件trvFolder。

最后，每次添加TreeNode的时候，多加一句：

```
curNode.ImageIndex = 0;
```

即为：

```
TreeNode curNode = new TreeNode(name);
curNode.ImageIndex = 0;
trvFolder.Nodes.Add(curNode);
```

其中0为图片的index，可根据自己的实际情况改为对应的index。

这样就可以实现给不同的TreeNode添加对应的图标了。

4.3. 获得右击所在节点

选择的是别的节点，对着另外一个节点右击，要获得被右击的节点，主要代码是：

```
private void trvCategoryTree_MouseUp(object sender, MouseEventArgs e)
{
    if (e.Button == MouseButton.Right)
    {
        // Select the clicked node
        trvCategoryTree.SelectedNode = trvCategoryTree.GetNodeAt(e.X, e.Y);
    }
}
```

之后通过cmsSelection_ItemClicked也就可以通过SelectedNode得到当前右击的那个TreeNode了。

详见：[【已解决】如何获得C#中右击弹出菜单时对应的当前所右击那个TreeNode](#)¹

4.4. 添加右键菜单 (Context Menu)

给TreeNode添加右键的总体的思路是：

1. 添加ContextMenuStrip
2. 在其下添加子菜单
3. 实现ItemClicked事件，在其中判断点击了哪个子菜单，然后添加代码做你要做的事情即可。

详见：[【已解决】C#中给TreeNode添加右键 \(Context Menu \)](#)²

4.5. 给TreeNode添加名字

TreeNode加入到TreeView但是不显示名字，是因为加入的TreeNode没有设置Text

（虽然设置了TreeNode的Name，但是实际上用于显示的是Text而不是Name）

所以，在把TreeNode加入TreeView之前，设置好对应的Text即可正常显示。

详见：[【已解决】给TreeView添加TreeNode节点后但是名字没显示出来](#)³

¹ http://www.crifan.com/csharp_get_current_right_click_treenode/

² http://www.crifan.com/csharp_treenode_add_right_menu_context_menu/

³ http://www.crifan.com/csharp_added_treenode_for_treeview_but_not_display_treenode_name/

参考书目

- [1] [正则表达式包含引号怎么办?](#)¹
- [2] [c# Regex常用](#)²
- [3] [c#学习笔记《1》——regex类（个人理解）](#)³
- [4] [为 Windows 窗体 TreeView 控件设置图标](#)⁴
- [5] [winform中利用Treeview模仿资源管理器实现图片文件列表](#)⁵
- [6] [C#键值对容器](#)⁶

¹ <http://tech.techweb.com.cn/thread-236996-1-1.html>

² <http://www.cnblogs.com/jorgen/archive/2010/10/14/1851555.html>

³ <http://hi.baidu.com/wenyouming1989/blog/item/c0d1f95a1e296cd39c8204b4.html>

⁴ <http://msdn.microsoft.com/zh-cn/library/aa983725>

⁵ <http://blog.csdn.net/mngzilin/article/details/5535136>

⁶ <http://lujian811.blog.163.com/blog/static/10190113201071933020824/>