

各种计算机语言简介和总结

版本：**v1.1**

Crifan Li

摘要

本文主要介绍了各种计算机语言的分类和关系，以及记录了笔者对其中部分语言，如Word VBA,Excel VBA,Python,C++,Javascript,C#等，的心得和体会。



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/language_summary/release/html/language_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/language_summary/

修订历史

修订 1.1	2015-05-07	crl
--------	------------	-----

1. 完成此文逻辑框架
2. 添加如何在Windows下的cmd环境下使用Python脚本
3. 添加了C++的内容
4. 添加了Javascript的内容
5. 添加了Word VBA的内容
6. 添加了Excel VBA的内容

7. 添加了Python的内容

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

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

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

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

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

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

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

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

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

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

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

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

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

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

-
8. 添加了C#的内容
 9. 添加了Python的crifanLib.py的全部内容
 10. 添加了C#的crifanLib.cs的全部内容
 11. 添加了Python的BeautifulSoup心得
 12. 已将Python移出至单独的book
 13. 已将C#移出至单独的book
 14. 添加为何高级语言不能直接操作硬件
-

各种计算机语言简介和总结:

Crifan Li

版本 : v1.1

出版日期 2015-05-07

版权 © 2015 Crifan, <http://crifan.com>

本文章遵从 : [署名-非商业性使用 2.5 中国大陆\(CC BY-NC 2.5\)](http://creativecommons.org/licenses/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

1. 各种计算机语言简介	1
1.1. 各种计算机语言简介	1
2. 脚本语言	2
2.1. 脚本语言综述	2
2.2. Python语言总结	2
2.3. Word VBA	2
2.3.1. 什么是word vba ? 为何要用VBA ?	2
2.3.2. Word VBA的一些基本概念	2
2.3.3. Word vba学习记录	3
2.3.4. 授人鱼不如授人以渔	9
2.4. Excel VBA	10
2.4.1. Excel VBA中基本的名词和概念	10
2.4.2. Excel VBA中的一些基本操作	13
2.4.3. Excel VBA更多的学习资料	14
2.4.4. 授人鱼不如授人以渔	15
2.5. Javascript	16
2.5.1. javascript学习记录	16
2.6. PHP	19
2.6.1. PHP学习记录	19
2.6.2. crifan的PHP库	19
2.6.2.1. 与字符串等有关的函数	20
2.6.2.1.1. 添加末尾的斜杠:addTailSlash	20
2.6.2.1.2. 去除开始的斜杠:removeFirstSlash	20
2.6.2.1.3. 检查一个字符串是否在另一个数组中:strExistInArr	20
2.6.2.1.4. 从文件过滤类型生成文件后缀列表:genValidSuffix	21
2.6.2.2. 与文件等有关的函数	22
2.6.2.2.1. 获得文件名的后缀:getFileSuffix	22
2.6.2.2.2. 如何获得某个文件下面的文件 (和子文件夹)	22
2.7. Linux Shell	22
2.7.1. Linux Shell学习心得	22
2.7.2. Linux Shell参考资料	25
2.8. Windows Batch File	25
2.8.1. Windows Batch File简介	25
2.8.2. Windows Batch File学习心得	25
2.8.3. Windows Batch File参考资料	25
3. 上层语言	26
3.1. 上层语言综述	26
3.2. C#学习心得	26
3.3. C++	26
3.3.1. C++学习笔记	26
3.4. 为何高级语言不能直接操作硬件?	27
4. 网络相关的语言	28
4.1. XML	28
4.1.1. XML相关的参考资料	28
4.2. HTML	28
4.2.1. HTML学习心得	28
4.2.2. HTML相关的参考资料	28
参考书目	30

表格清单

3.1. protected , public , private之间的区别	26
--	----

范例清单

2.1. addTailSlash使用范例	20
2.2. strExistInArr使用范例	21
2.3. genValidSuffix使用范例	22
2.4. getFileSuffix使用范例	22

第 1 章 各种计算机语言简介

1.1. 各种计算机语言简介

第 2 章 脚本语言

2.1. 脚本语言综述

脚本语言综述

2.2. Python语言总结

关于Python语言总结，已移至这里：[Python语言总结](#)¹

2.3. Word VBA

2.3.1. 什么是word vba ? 为何要用VBA ?

VBA是Visual Basic For Application的简称。

换句话说，就是将VB应用于Application应用程序，此处的Application指的是微软的Excel，word等应用程序。

所以，对于我用到的，在word里面写宏，去实现一些操作，用到的VB，就是Word VBA了。

而其他常用的，还有Excel里面也会用到用宏去实现一些根据自己需求实现的一些功能，比如自动提取一些内容到一个文档中，否则，自己手动慢慢地复制粘贴，往往效率极低，累死你不说，还不容易保证完全正确，而通过宏来处理，运行一下宏，点击一下某个菜单等，即可实现要做的事情，高效，准确。

就像别人说的，除了"Excel本身的内置函数其实已经很强大了"，但是只是很强大，有些更高级或复杂的操作，还是宏实现起来更方便和快捷。

再简而言之，如果你不是需要对某（些）文档特殊处理，只是一般的使用word或excel文档，多数人都是用不到的。

而如果有类似于我这样的需求，将某个word文档中的很多表格中的符合某些特定条件的某些列的内容，经过一定条件判断，将对应的值提取到一个xml文档中，并且输出的时候，要将该值根据某些约定，再分成几个部分等等，这类的操作，手动实现，要一个个去找对应的表格，然后找到对应的值，然后手动复制，粘贴，一点点做，效率很低，而如果可以用宏实现，只需要写好宏代码，运行一下，即可省去大量的精力。

不过，就像[2]说的："VBA功能很强大，但并不是万能的，也并不是所有工作都需要用VBA来解决，也并不是所有工作用VBA来解决都会很简单，这要根据实际情况而定，有些工作你用VBA来解决的话相反会变得很麻烦。"

2.3.2. Word VBA的一些基本概念

此处介绍一下，Word VBA中的一些基本单位（名词）：story, section, paragrah, line, character,table,cell,row, column

对于所选一段文字后，即引用Selection，其中会包含很多属性，Selection.XXX 其中XXX就有下面列出来的很多种，现在简单说说我对这些属性含义的理解：

1. Story

中文直译为 故事，经过测试，感觉对于word文档来说，好像就是整个文档的内容。

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

2. Text

不必多解释，就是所选的内容

3. Sections

节。经测试，好像整个word文档，就一个节，有点像story，具体两者的区别，暂未搞懂。

4. Paragraphs

段落。按照中文的段落的含义理解，怎么说也应该有个几十行之类的文字，才算一个段落的，这是之前学语文的时候的感觉。结果发现这里的段落和中文意思的段落，差别很大，一个文档里面，没多少行字，结果段落就几百个了，貌似是一个单词，一个换行，一个表格中的值，都算一个段落。

5. Sentences

句子，同常理，不多解释。

6. Words

单词，同常理，不多解释。

7. Characters

字符，同常理，不多解释。

8. Tables , Cells , Rows , Columns

表格，同常理。其中包含了很多个cell，即单元格。每个cell，都对应不同的行和列。

（打印）输出所选Range中（其包含很多个表格）中的第二个表格中的第三行第四列的值，表示为：

```
Print Selection.Tables(2).Cell(3, 4)
```



注意

其中第三行第四列，不是Cell(2,3)，因为其单元格的下表是从1开始，而不是原以为的类似于C语言或者Excel中的0为起始值的。

9. End , Start

即Selection.Start和Selection.End，都是直接赋值的

所以，如果你想要选中某段内容（文字），那么只要找到你所要的起点和终点，然后将其值赋值给Selection.Start和Selection.End，即可选中该段内容（文字）。

当然，也可以利用两者算出所选内容的长度：

```
selection_len = Selection.End - Selection.Start
```

10.

2.3.3. Word vba学习记录

1. 获得所选range的文字

尝试了一下，发现两种都可以：

```
Selection.Text  
Selection.Range.Text
```

2. 不等于符号

VBA里面的不等于号，用"<>"表示，而不是其他C等语言里面的"!="

<>，即大于号和小于号的组合，意思也很明确，既不大于也不小于，所以就是不等于了。

3. 如何选中一行并获得对应的文字 和 wdMove和wdExtend的区别

前提，光标（或叫插入点）已经移动到该行了，但是想要选中该行，并想要获得对应所选Range的文字

此时可以用如下代码：

```
Selection.MoveDown Unit:=wdLine①, Count:=1, Extend:=wdExtend②
```

① wdLine是单位（Unit）

类似的Unit还有：wdParagraph，wdWindow，wdScreen。

② 和WdExtend对应的还有wdMode

对此，VBA教程的解释是：“如果是 wdMove，则所选内容折叠到结束位置，并向下移动。如果是 wdExtend，则所选内容向下扩展。默认值为 wdMove。”

没完全理解，但是经过测试得知，如果是mdMove，那就是光标移动到对应位置了，但是没有选择这（光标移动的）起点到终点，此处为整个这一行，的内容，如果是wdExtend，那就是不仅移动光标了，并且同时选中了起点到终点的内容。

4. 如何跳转到某一标题 和 如何移动光标/插入点

```
Selection.GoTo① What:=wdGoToHeading②, Which:=wdGoToAbsolute③, Count:=5
```

① 这里主要是利用Goto语句

② wdGoToHeading说明掉转的单位是heading标题

③ wdGoToAbsolute表示是绝对位置的跳转

于此相对应的还有wdGoToFirst, wdGoToLast, wdGoToNext, wdGoToPrevious, wdGoToRelative。

更多详细内容，请参考[\[3\]](#)

另外，对于移动光标/插入点，可以借鉴如下做法：

本示例将所选内容移至当前文字部分的开头。如果所选内容位于文档正文部分，则本示例将所选内容移至文档的开头

```
Selection.HomeKey Unit:=wdStory①, Extend:=wdMove
```

① wdStory属于wdUnits

其他类似的还有wdColumn，wdLine，wdRow等

5. 如何获得选中的标题的序号

前提，已经选择某一标题了，想要获得该标题前面的序号

比如对这个标题：

2.1 PAPERBY / RubyVerifoneDefaults

想要获得标题的序号，即2.1

可通过下列代码获得当前标题的序号，即2.1

```
Selection.Bookmarks("\headinglevel❶").Range.ListFormat.ListString
```

❶ \headinglevel是VBA里面的预定义标签

关于此内容的详细解释，请参考：[\[5\]](#)

6. 如何返回函数的返回值

关于这点，网上找，没几个说的很清楚的，最后还是通过VBA的Help才搞懂如何返回函数返回值的。

此处，简单解释如下：

一个Function函数的语法为：

```
[Public | Private | Friend] [Static] Function name❶ [(arglist)] [As type]
[statements]
[name = expression❷]
[Exit Function]
[statements]
[name = expression❸]
End Function
```

❶ name即为函数名

❷ expression是表达式，当然也可以是某个变量，常量等，其就代表了返回值。

更具体的解释和例子，摘录如下，很容易看懂的：

要从函数返回一个值，只需将该值赋给函数名。在过程的任意位置都可以出现这种赋值。

如果没有对 name 赋值，则过程将返回一个缺省值：数值函数返回 0，字符串函数返回一个零长度字符串 ("")，Variant 函数则返回 Empty。

如果在返回对象引用的 Function 过程中没有将对象引用赋给 name（通过 Set），则函数返回 Nothing。

下面的示例说明如何给一个名为 BinarySearch 的函数赋返回值。

在这个示例中，将 False 赋给了该函数名，表示没有找到某个值。

```
Function BinarySearch(.....) As Boolean
.....
'值未找到，返回一个 False 值。
If lower > upper Then
    BinarySearch = False
Exit Function
```

```
End If
.....
End Function
```

看懂后，多说一句，有问题，还是先找VBA的Help系统，比百度，google等，有时候更准确，方便，易懂。

- ④ 函数内部任何地方都可以出现[name = expression]

7. 字符串处理函数

关于VBA中的字符串相关的函数，刚发现，office 2010 中的 VBA的help文件里面，有已经总结好的了：

作用	关键字
比较两个字符串	StrComp
变换字符串	StrConv
大小写变换	Format, LCase, UCase
建立重复字符的字符串	Space, String
计算字符串长度	Len
设置字符串格式	Format
重排字符串	LSet, RSet
处理字符串	InStr, Left, LTrim, Mid, Right, RTrim, Trim
设置字符串比较规则	Option Compare
运用 ASCII 与 ANSI 值	Asc, Chr

上述字符串处理相关的函数，多数都是见名知意，容易看懂和使用，但是也有些不太容易搞懂，或者和普通C语言等里面的字符串处理函数不同的，需要额外说明的，解释如下：

a. Space, String

其中Space很方便，比如想要输出10个空格，当然你可以通过自己手动的写成" "，但是有了space，用其更方便，直接写成

```
Space(10)
```

即代表了10个空格。同理

```
String(number,characterToRepeate)
```

用于输出多个某一字符，而对于前面的空格来说，也可以用string表示，即

```
Space(10)=String(10, 32)
```

其中32=0x20，是space的ASCII值。而String的优点在于，也可以输出其他字符，包括0x0~0x1F之类不可见字符。

b. InStr, InStrRev

这个InStr，其实就是C语言的strstr

语法为：

```
InStr([start, ]string1, string2[, compare])
```

意思为：“返回 Variant (Long)，指定一字符串在另一字符串中最先出现的位置。”

而InStrRev意思和InStr类似，只不过是“从字符串的末尾算起”。

举例说明：

```
posOfPoint = InStr("abc.def", ".") ' 查找"abc.def"中的".",返回值为4
```

c. Asc, Chr

Asc是将字符转换为ASCII数值，Chr是将数值转换为ASCII字符。

举例为：

```
MyNumber = Asc("A") ' 返回 65。  
MyChar = Chr(97) ' 返回 a。
```

d. 如何创建文件和输出信息到文件里面

在VBA中创建 (log) 文件的两种方法.

i. 方法1：

```
Function createFileTest()  
' the name of the file to be output, including full path  
Dim FileName As String  
FileName = "D:\create_file.txt"  
' open the file output text stream output, overwriting is necessary  
Dim fs As Object  
Set fs = CreateObject("Scripting.FileSystemObject")  
Dim file As Object  
Set file = fs.CreateTextFile(FileName, True)  
file.WriteLine "test string..."  
file.Close  
End Function
```

ii. 方法2：

```
Public gFileNum As Integer  
Function createOutputFile()  
Dim outputFileNameWithPath As String  
Dim gDestFile As String  
Dim outputFileNameWithPath As String  
Dim openFileOK As Boolean  
openFileOK = True  
' 1. create an XML file  
outputFileNameWithPath = "D:\test_output.txt"  
gDestFile = outputFileNameWithPath  
' Obtain next free file handle number.  
gFileNum = FreeFile()  
' Turn error checking off.  
On Error Resume Next  
' Attempt to open destination file for output.
```

```
Open gDestFile For Output As #gFileNum
' If an error occurs report it and end.
If Err <> 0 Then
    openFileOK = False
    MsgBox "Cannot create file: " & gDestFile
End If
' Turn error checking on.
On Error GoTo 0
createOutputFile = openFileOK
End Function
Sub createFileTest()
Print #gFileNum, "Test output string to this file..."
Print #gFileNum, "Test done."
Close #gFileNum
End Sub
```

e. 如何获得当前文件的文件名 和 当前文件所在路径

- i. 获得当前文件所在的路径：ActiveDocument.Path。
注意：所得结果，最后没有那个路径分隔符“\”
- ii. 获得当前文件的文件名：ActiveDocument.Name
- iii. 获得当前文件的路径和文件名，即全名：ActiveDocument.FullName

f. VB里面的一些常量

office 2010的help文件里面已有总结：

Visual Basic 常数

Visual Basic for Applications 定义了一些常数，使程序设计变得更为简单。下列常数可在程序代码中的任何地方代替实际值：

- Calendar 常数
- CallType 常数
- Color 常数
- Compiler 常数
- Date 常数
- Dir、GetAttr 和 SetAttr 常数
- IMEStatus 常数
- Instr、StrComp 常数
- Keycode 常数
- Miscellaneous 常数
- MsgBox 常数
- QueryClose 常数
- QueryClose 常数
- Shell 常数

- StrConv 常数
- 系统颜色常数
- VarType 常数
- Visual Basic 常数

其中，此处打算把一些我所常用到的一些常数列出来，以供参考
Miscellaneous 常数

下列常数由 Visual Basic for Applications 中的类型库定义，可用在代码中的任何地方代替实际值

常数	等于	描述
vbCrLf	Chr(13) + Chr(10)	回车符与换行符结合
vbCr	Chr(13)	回车符
vbLf	Chr(10)	换行符
vbNewLine	Chr(13) + Chr(10) or, on the Macintosh, Chr(13)	平台指定的新行字符；适用于当前平台
vbNullChar	Chr(0)	值为 0 的字符
vbNullString	值为 0 的字符串	用来调用外部过程；与长度为零的字符串 ("") 不同
vbObjectError	-2147221504	用户定义的错误号应当大于该值，例如：Err.Raise Number = vbObjectError + 1000
vbTab	Chr(9)	Tab 字。
vbBack	Chr(8)	退格字符
vbFormFeed	Chr(12)	在 Microsoft Windows or on the Macintosh 中没有作用
vbVerticalTab	Chr(11)	在 Microsoft or on the Macintosh Windows 中没有作用

2.3.4. 授人鱼不如授人以渔

下面介绍一些更多有用的参考资料，以及介绍一下遇到问题如何去解决问题。

1. 通过VBA内置的Help系统查询不懂的内容

在word里面打开VBA编辑器编辑vba代码后，对于常见的一些对象，属性，方法（函数）等的具体解释，多数都可以通过选择该对象/属性/方法，然后按F1键，打开帮助，其会自动跳转到对应的帮助内容，很是方便，而且很多的细节，解释的也比较详细，可以多多利用。

其中都包括了一些基本知识的详细解释，比如上面提到的，关于VBA中关于Function的返回值，其要去查看Help文件，就可以通过选择某函数，

比如，选中：

```
Function funcName()  
'XXXXXX
```

End Function

中的Function，然后按F1键，系统会自动跳转到Help文件中关于Function的解释，很方便。

2. 常见函数分类列表

office 2010中VBA的help文件做的很好，里面总结了一些常见的函数，前面列出来的，就是那个字符串处理相关的函数。

下面列出了，其他一些类别，先放到这里，以备后查：

分类	描述
数组	建立、定义与使用数组
编译命令	控制编译方式
控制流	循环及过程流程控制
变换	变换数值及数据类型
数据类型	数据类型与 variant 子型态
日期与时间	变换、使用日期与时间表达式
目录和文件	控制文件系统与处理文件
错误	侦测与返回错误值
金融	进行各种金融运算
输入与输出	接受输入与显示、打印输出
数学	完成各种数学运算
其他	激活其他应用程序与处理事件
操作符	比较表达式与完成其它操作
字符串处理	处理字符串及字符串数据类型
变量与常数	声明、定义变量与常数

3. 网上在线查找相关帮助内容

刚找到的，微软的官方的在线帮助：

<http://office.microsoft.com/zh-cn/access-help/CH010072891.aspx?CTT=97>

2.4. Excel VBA

看此部分内容之前，关于VBA的基础知识，以及word VBA的相关知识，可参考[第 2.3 节 “Word VBA”](#)

2.4.1. Excel VBA中基本的名词和概念

此处介绍一些Excel中的最基本的概念，搞清楚这些概念，是继续深入学习Excel的基础。

Excel中，我们已经知道的是，从概念上来说，从大到小是：

Excel文件 ⇒ Excel文件中页表sheet ⇒ sheet中的行和列 ⇒ 行列交叉点所对应的单元格

这些名词和概念，对应到Excel VBA中就是：

Workbook ⇒ Worksheet ⇒ Column, Row ⇒ Cell

并且，每个概念，都对应着一个该单词的复数形式，表示的是对应的一组的该单位，

如：Workbooks表示一组，很多个当前已经打开的Excel文件等。

下面稍微详细地介绍一下各自的含义和用法。

1. Workbook

workbook属于workbooks，workbooks是当前打开的所有的workbook的集合（总称）。

对于想要获得单个的workbook，有这些方式：

- `Workbooks.Item(N)`
- `Workbooks(N)`
- `Workbooks("excel_file_name.xlsx")`

Workbook所具有的一些常见函数有：

- `Workbook.Open`
- `Workbook.Add`
- `Workbook.Save`
- `Workbook.SaveAs`

对于其更多的函数和属性，细心的人，自然会想到利用前面提到的自学方法，去更加详细的学习了解的。以后对此类提示，就不再多提醒了。

2. Worksheet

worksheet即我们常说的Excel中sheet。

worksheet属于worksheets。

Excel软件，创建一个Excel文件的时候，默认会自动创建3个sheet，对应的就是该workbook的worksheets，包含了3个worksheet。

对应获得当前的worksheet的方式：

- `Workbooks.Item(1).Sheets.Item(N)`
- `Workbooks.Item(1).Sheets.Item("sheet_name")`

3. Column

对应的概念很好理解，不多解释。

对应每个列，都有一个自己的名字，比如第三列，名字一般是"C"。

获得第三列的方式有：

- `Columns(3)`
- `Columns("C")`

获得3到5列，可以：

- `Columns("C:E")`
- `Range("C:E")`

选中1, 5, 7, 8, 9列：

```
Range("A:A, E:E, G:G, H:H,I:I").Select
```

设置宽度：

```
Columns("C").ColumnWidth = 4.5
```

大小自动适应：

```
Selection.Columns.AutoFit
```

隐藏显示某列用对应的：

```
Columns("C").hidden=True
```

4. Row

行的操作，非常类似于列的操作：

选中2到6行：

```
Rows("2:6").Select
```

选中第一行：

```
Rows(1).Select
```

选中5, 7, 8行：

```
Range("5:5, 7:7,8:8").Select
```

设置行高：

```
Rows(6).RowHeight = 2.5
```

选中某6行，将其隐藏：

```
Rows("6:6").Select
Selection.EntireRow.Hidden = True
```



可以通过行Row或列Column的名称来操作行或列

另外，如果单独对于某列，比如第三列，默认名字，是"C"，可以同选中该列，鼠标移动到左上角那个输入框中，鼠标会自动显示"Name Box"，此时你可以在里面输入你想要的名字，比如"Col3"，然后接下来的所有的对第三列的操作和引用，之前是通过Columns("C")来获得的，现在就可以通过

```
Columns("Col3")
```

来操作了。

即Columns("XXX")中的XXX是对应的列的名字。

对于行的操作，也是同样适用的。

即可以先去对某行去命名，然后就可以通过对该名字来引用到该行了。

行和列交叉点，对应的就是单元格cell。

下面是各种操作：

第四列第六行：

```
Range("D6")
```

设置单元格的一些属性：

```
Range("B3").Font.Name = "Cambria" '也可以设置为宋体，微软雅黑  
Range("B3").Font.Bold = True
```

选中从D2到H6的整块的单元格：

```
Range("B2:H6")
```

用同样的方式去选中某个单个的单元格：

```
Range("D4:D4")
```

选中两块单元格：

```
Range("B2:C5,EC4:G8").Select
```

选中部分重叠的两块单元格：

```
Range("B2:C5, C4:G8").Select
```

取消显示网格线：

```
ActiveWindow.DisplayGridlines = False
```

2.4.2. Excel VBA中的一些基本操作

1. 如何添加注释

VBA中的注释，是以单引号"'"开头的，其后内容，都视为注释内容。

另外一种，也可以通过关键字REM或Rem加上个空格，之后的内容，也会被识别为注释。

比如：

```
REM this is a excel maro  
' this is a excel maro
```

2. 如何调试

简单的说，有两种方式：

a. MsgBox

一是直接调用Msgbox去弹出对话框打印信息。

比如：

```
MsgBox "Currently opening Excel file is: " & ActiveWorkbook.Name
```

其中，字符串链接/拼接，用&符号。

b. log文件

二是新建一个log文件，将调试或打印信息输出到log文件中去。

关于如何创建文件，可参考：[在VBA中创建（log）文件的两种方法](#)

3. 获得当前的Excel文件所在路径

类似于Word VBA中的ActiveDocument，Excel中是ActiveWorkbook

所以获得当前打开的Excel的文件路径是：

```
ActiveWorkbook.Path
```

与此相关的是：

```
ActiveWorkbook.FullName = ActiveWorkbook.Path + " + ActiveWorkbook.Name
```

更多相关的内容，请自己尝试：

- `MsgBox ActiveWorkbook.Name`
- `MsgBox ActiveWorkbook.FullName`
- `MsgBox ActiveWorkbook.FullNameURLEncoded`
- `MsgBox ActiveWorkbook.Path`

4. 一些通用的属性和方法（函数）

对于选中当前的内容，比如workbook，sheet等，可以通过.Select来选中，

比如：

```
Sheets("sheet_name").Select
```

对于当前的内容的名字，对应的是.Name属性，比如：

```
Sheets("sheet_name").Name
```

很多内容，都有move的功能，详细用法请自己探索。

新增一个新的内容，比如workbook，sheet等，都可以用Add方法。

2.4.3. Excel VBA更多的学习资料

1. VBA for Microsoft Office Excel 2007

关于Excel VBA的一些基础知识，可以去看这里：

[VBA for Microsoft Office Excel 2007](http://www.functionx.com/vbaexcel/)²

2. Excel 2007 VBA参考大全

关于Excel VBA，这本书介绍的很全：

² <http://www.functionx.com/vbaexcel/>

[《Excel 2007 VBA参考大全》扫描版 \[PDF\]](#) ³

电驴地址：

[ed2k://](#)

[\[file\]5BExcel.2007.VBA%E5%8F%82%E8%80%83%E5%A4%A7%E5%85%A8%5D.%EF%BC%88%E7%BE%8E%EF%BC%89%E6%A0%BC%E6%9E%97.%E6%89%AB%E6%8F%8F%E7%89%88.pdf|253406092|3cf57fc49d0fa875826ecc462cd9976c|h=eed6g2e25uuzc4c52obch2txlm7w65fx|/](#)

2.4.4. 授人鱼不如授人以渔

此部分内容，其实在之前[第 2.3 节 “Word VBA”](#) 中，已有提到，此处再次强调一下。

在学习VBA（和任何类似知识）的过程中，学会利用系统已有资源。

即，在VBA中，可以供我们利用的资源，至少有这些：

1. VBA中的对象浏览器（Object Browser）

可以通过在写VBA代码的过程中，对于任何你不太懂，想要了解更多细节的内容，选中该内容，比如我想要了解workbook，那就选中workbook，然后右键⇒Object Browser,即可在Object Browser中查找到你所要的内容，其中可以直观快速地看到对应目标所具有的变量成员和函数等，很是方便。

2. 内置的Help

选中想要查询的内容，然后按F1，VBA即可调用Help帮助系统，可以查询到更多的细节和实例用法。

例如，我知道了workbook是对应于当前Excel文件，想要了解workbook有哪些属性和方法，

那么可以去F1的Help中输入workbook，然后搜索到很多条目，其中第一条就是：

[Workbook Object Members](#) ⁴

“Article Represents a Microsoft Excel workbook. Methods Name Description
AcceptAllChanges Accepts all changes in the specified shared workbook.
Activate Activates the fi...”

然后点击进入后，就可以看到“Workbook Object Members”的页面，其中有三部分：Methods, Properties, Events

里面有你想要的所有的成员和函数等。

比如，表示当前Excel的文件名的.Name的属性，文件全名（包括路径和文件名）的.FullName的属性等。

然后你也可以通过Msgbox去自己尝试打印出来看看，都是什么值：

```
MsgBox ActiveWorkbook.Name
MsgBox ActiveWorkbook.FullName
```

这样的话，你就可以可以通过自学的方式，去搞懂很多内容了。

³<http://www.verycd.com/7topics/2872866/>

⁴<http://office.microsoft.com/client/help/preview.aspx?AssetId=HV805547869990&Icid=1033&NS=EXCEL%2EDEV&Version=12&respos=0&CTT=1&queryid=cc5dd30b%2D82da%2D4044%2Db28e%2D3ee83e8f8>

script的语法，说是script中type是必须的，即必须要写成：

```
<script type="XXX">
....
</script>
```

但是实际上看到一些网页，比如：

http://hi.baidu.com/recommend_music/blog/item/5fe2e923cee1f55e93580718.html

的html源代码中，script没有指定type：

```
<script>
  /******there is some function.
  function checkMail(s) {
    var pattern=/w+@w+.[a-z]+/;
    if(pattern.test(s)) {
      return true;
    } else {
      return false;
    }
  }
}
.....
</script>
```

一直很奇怪，为何可以用。

后来终于找到解释了：

<http://blog.csdn.net/slalx/article/details/5093756>

二．指定脚本语言的方式：

1.指定整个文档默认的脚本语言。

```
<META http-equiv="Content-Script-Type" content="type">
```

content：指定脚本语言的类型如："text/tcl"，"text/javascript"，"text/vbscript"。

由于文档默认的脚本类型是JavaScript，因此我们在使用JavaScript时最简单且兼容性最好的方式为<script></script>

不过，即使如此，还是没在上面那个网页的html源码中，看到有期望的：

```
<META http-equiv="Content-Script-Type" content="type">
```

估计是该网页的javascript写的不够严谨吧。

2. 关于字符串的format函数

参考这里：

<http://witmax.cn/js-function-string-format.html>

举例用法：

```
return "<a href=' {1}' target='_blank' class=' ucard' pid=' {0}' >{2}</a>".format(portraitId, cmturl, cmtname);
```

其中，{0},{1},{2}分别代表format后面的参数的第一个，第二个，第三个。

3. 打印或枚举出一个变量的所有属性property，或一个字典的所有键（key）和值（value）

a. 打印或枚举出一个变量（对象）的所有的属性property

参考这里：

<http://blog.csdn.net/sd2131512/article/details/6328300>

知道了，可以这样写：

```
for(var name in someObject){  
    document.write(name + "=" + someObject[name])  
}
```

此处someObject是某个对象变量，然后上面的for语句，加上"变量[属性]"的方式，可以获得所有的变量，并打印出来。

b. 打印或枚举出一个字典的所有键key和值value

参考这里：

http://www.w3schools.com/JS/js_loop_for_in.asp

后经过测试，下面代码在IE9里面可以运行：

```
var arrayData = {  
    a: "11",  
    b: "22",  
    c: "33",  
    d: "44"  
}  
  
for ( _key in arrayData)  
{  
    document.write( _key + "=" + arrayData[_key])  
}
```


其中上面的arrayData是一个字典，然后也是通过"字典[键]"的方式，获得对应键的值

4. 如果某一行javascript脚本出错，则执行结果出错，但是却没有任何提示

最新在折腾【记录】折腾给WordPress Google XML Sitemap中添加支持子目录自动扫描功能，包含HTML静态页面等可识别的格式⁵的过程中，发现一个现象：

html或php页面中，出现了非所预料的结果，知道是代码某处写错了，但是不值得是哪个文件的哪一行错了。

最后才无意间发现，原来是某个js脚本中，之前添加的多余注释TODO字符串，没有及时删除，导致页面结果出错。

由于感慨，js代码执行结果异常，某行出错了，也没有任何的提示和警告，导致最终结果虽然错了，也知道是代码问题，但是却无法定位错误的位置。

如此去调试html，php，javascript混合的页面，真的很痛苦。

期待高手可以指教指教，有没有更加有效的调试方法。

5. 如果改动了javascript中的内容，一定要刷新页面，才可以看到最新改动的效果！！

最新在折腾【记录】折腾给WordPress Google XML Sitemap中添加支持子目录自动扫描功能，包含HTML静态页面等可识别的格式⁶的过程中，在修改对应的javascript文件sitemap.js，但是，始终却不见修改后的效果。

但是很奇怪的是，之前对于HTML页面等内容，修改了后，点击对应选项，是可以看到最新的改动的效果的。

而此处不论如何改动js文件，结果都不见效果。

最后，才发现，原来对于js文件，需要重新刷新HTML页面，其会重新加载最新的js文件，然后才能看到最新的效果。

而之前不知道这一点，真的是浪费了非常非常多的时间。

2.6. PHP

2.6.1. PHP学习记录

1.

2.

2.6.2. crifan的PHP库

暂时还没有合并所有的php相关的函数到crifanLib.php中，此处只是整理出目前已有的一些有用的函数而已。

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

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

2.6.2.1. 与字符串等有关的函数

2.6.2.1.1. 添加末尾的斜杠:addTailSlash

```
//add the tail slash if not exist
//eg: from files/doc/docbook to files/doc/docbook/
function addTailSlash($str)
{
    if(substr($str, -1) != "/")
    {
        $str = $str."/";
    }
    return $str;
}
```

例 2.1. addTailSlash使用范例

```
//add tail slash
$homePath = $this->addTailSlash($homePath);
//echo "homePath=".$homePath."<br />";
```

2.6.2.1.2. 去除开始的斜杠:removeFirstSlash

```
//remove the first slash if exist
//eg: from /files/doc/docbook to files/doc/docbook
function removeFirstSlash($str)
{
    if(substr($str, 0, 1) == "/")
    {
        $str = substr($str, 1);
    }
    return $str;
}
```

2.6.2.1.3. 检查一个字符串是否在另一个数组中:strExistInArr

```
//eg: check txt is in array("txt", "html", "pdf")
function strExistInArr($str, $arr)
{
    if($arr[0] == "**")
    {
        $found = True;
    }
    else
    {
        $found = False;
        $arrLen = count($arr);
        for($i=0; $i<$arrLen; $i++)
        {
            //if(0 == strcmp($str, $arr[$i]))
        }
    }
}
```

```
    if($str == $arr[$i])
    {
        $found = True;
        break;
    }
}
return $found;
}
```

例 2.2. strExistInArr使用范例

```
$fileValid = $this->strExistInArr($fileSuf, $validSuf);
```

2.6.2.1.4. 从文件过滤类型生成文件后缀列表:genValidSuffix

```
//eg: from "*.html;*.txt;" to array("html", "txt")
function genValidSuffix($fileType)
{
    $validSuf = array();

    $splitedArr = explode(";", $fileType);
    $arrSize = count($splitedArr);
    //echo "splited array size=" . $arrSize . "<br />";
    for($i=0; $i<$arrSize; $i++)
    {
        $singleSufDef = $splitedArr[$i];
        //echo "singleSufDef=" . $singleSufDef . "<br />";
        if($singleSufDef && ($singleSufDef != ""))
        {
            //echo "singleSufDef not null <br />";
            $suffix = str_replace("*. ", "", $singleSufDef);
            //echo "suffix=" . $suffix . "<br />";
            if($suffix == "")
            {
                //echo "found & suffix <br />";
                $validSuf = array("");
                break;
            }
            else
            {
                //echo "add $suffix into array <br />";
                array_push($validSuf, $suffix);
            }
        }
    }

    return $validSuf;
}
```

例 2.3. genValidSuffix使用范例

```
//$validSuf = array("html", "htm", "txt", "pdf");  
$validSuf = $this->genValidSuffix($fileType);
```

2.6.2.2. 与文件等有关的函数

2.6.2.2.1. 获得文件名的后缀:getFileSuffix

```
//eg: from abc.txt got txt  
function getFileSuffix($fileName)  
{  
    $gotInfo = pathinfo($fileName);  
    // dirname, basename, extension  
    $suffix = strtolower($gotInfo["extension"]);  
    return $suffix;  
}
```

例 2.4. getFileSuffix使用范例

```
$fileSuf = $this->getFileSuffix($filename);
```

2.6.2.2.2. 如何获得某个文件下面的文件（和子文件夹）

相关的代码如下，供参考：

```
$dirHandler = opendir($fullPath);  
  
while(($filename = readdir($dirHandler)) !== false)  
{  
    if(($filename == ".") || ($filename == ".."))  
    {  
        //echo "omit filename=$filename";  
        continue;  
    }  
  
    echo "filename=". $filename";  
}  
closedir($dirHandler);
```

2.7. Linux Shell

2.7.1. Linux Shell学习心得

1. 递归函数中的变量值被下层调用所修改的问题

由于Linux shell中默认定义的变量都是全局的，所以导致在递归函数中定义的变量，在下一层调用中修改了值后，返回上一层，改变量已被修改，导致运行结果不对。

解决办法就是，给相应的变量加上local前缀，表示是局部变量，即可。

详情参看：[【已解决】Linux Shell中，递归函数调用过程中，变量的值被改变，即下层调用会影响上层中的变量的值](#)

2. ./xxx.sh: line xx: [: too many arguments

问题原因：参数太多

比如，使用

```
if [ -d $item_full_path ]; then
```

去判断item_full_path是否为文件夹，但是如果传入的参数是

```
arm_vs_mips.xml - Copy_bak
```

，则会出现too many arguments的错误。因为对于shell中的-d去判断一个参数是否为文件夹的时候，此参数，只能是一个字符串，而不能是多个字符串，而此处的

```
arm_vs_mips.xml - Copy_bak
```

虽然物理上表示对应的某个文件，但是作为此处的参数传递进来的话，的确就是三个字符串了：

```
arm_vs_mips.xml
-
Copy_bak
```

。

解决办法：自己去找错误的位置，是不是也是由于类似的，使用某个功能的时候，传递的参数，超过该功能所期望的参数的个数了。如果超过了，估计多数情况下，也都是你自己使用不当，或者没有进行合适的处理而导致的。

3. 文件名中包含空格的问题

在用shell过程中，经常会遇到处理文件名中包含空格的问题

对此，参考[Linux shell脚本 遍历带空格的文件名](#)⁸，尝试了用tr将ls输出结果替换为问号，的确是可行的：

```
CLi@PC-CLI-1 ~/develop/docbook/books/arm_vs_mips/src
$ ls
arm_vs_mips.xml  arm_vs_mips___xml - Copy_bak  glossary.xml  Makefile
remove_bak.sh
arm_vs_mips.xml.bak  ch01_reduced_instruction_set.xml  images  reference.xml
remove_bak.sh.bak
```

```
CLi@PC-CLI-1 ~/develop/docbook/books/arm_vs_mips/src
$ ls | tr " " "?"
arm_vs_mips.xml
arm_vs_mips.xml.bak
arm_vs_mips___xml? -?Copy_bak
ch01_reduced_instruction_set.xml
```

⁷ http://www.crifan.com/linux_shell_recursive_function_variable_be_changed/

⁸ <http://hi.baidu.com/cn/pojoflcc/blog/item/97123b1f6a89b5094034176f.html>

```
glossary.xml
images
Makefile
reference.xml
remove_bak.sh
remove_bak.sh.bak
```



cygwin中的tr在shell中工作不正常？ -> 实际上是cygwin中的显示有问题

不过，另外提及一下，目前遇到一个问题，就是上述的代码：

```
ls | tr " " "\?"
```

放到shell中执行：

```
local sub_folders=`ls $input_folder | tr "\32" "\?";
echo '444 sub_folders='$sub_folders;
```

当时结果是tr无效，输出的效果中文件名中的空格没有被替换为问号。

其中当前环境是用的Cygwin，tr版本为：

```
CLi@PC-CLI-1 ~/develop/docbook/books/arm_vs_mips/src
$ tr --version
tr (GNU coreutils) 8.15
Packaged by Cygwin (8.15-1)
Copyright (C) 2012 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

Written by Jim Meyering.
```

后来终于经过一番折腾，而搞清楚了：

把问号换成其他的，比如<,>等：

```
local sub_folders=`ls $input_folder | tr "\32" "\<";
local sub_folders=`ls $input_folder | tr "\32" "\>";
```

也都是可以看到替换的结果的。

并且也可以用：

```
each_item=`echo $each_item | tr "\<" " ";
each_item=`echo $each_item | tr "\>" " ";
```

再替换回来的。

而对于上述的问号替换空格，虽然没有显示出来，但是如果没有此操作的话，后面

```
for each_item in $sub_folders; do
```

所得到的每个item中，

```
arm_vs_mips.xml - Copy_bak
```

就会变成三个字符串：

```
arm_vs_mips.xml
```

```
-
```

```
Copy_bak
```

而有此操作，后面得到的就是一个独立的字符串：

```
arm_vs_mips.xml - Copy_bak
```

所以明白了根本原因是，Cygwin中，没有正确的显示出来，实际上，是已经替换成功的了。

不过很悲催的是，单独echo问号字符，却又是可以正常显示的。。。

说实话，这样的bug，真的很让人崩溃。。。太浪费时间了。。。

2.7.2. Linux Shell参考资料

1. [Linux Shell Scripting Tutorial - A Beginner's handbook](#) ⁹

用经典的那句中文介绍此书就是，对于Linux Shell的"从入门到精通",^_^

专门提及一下常用的供参考的章节：

- [test command or \[expr \]](#) ¹⁰

2.8. Windows Batch File

2.8.1. Windows Batch File简介

2.8.2. Windows Batch File学习心得

2.8.3. Windows Batch File参考资料

- 1.

⁹ <http://www.freeos.com/guides/lsst/index.html>

¹⁰ <http://www.freeos.com/guides/lsst/ch03sec02.html>

第 3 章 上层语言

3.1. 上层语言综述

3.2. C#学习心得

关于C#的学习心得，已移到这里：[C#学习心得](#)¹

3.3. C++

3.3.1. C++学习笔记

学习C++过程中，遇到的一些内容，觉得有必要记录下来，以供后用。

其中主要是之前用C语言，所以接触C++，有些内容不太习惯，因此记录C++中和C不同的，需要注意的部分内容。

1. protected, public, private

对应的含义，用下表，可以很容易区分：

表 3.1. protected, public, private之间的区别

	关键字		
	Private 私有	Protected 受保护	Public 公有
对应作用（允许谁访问）	友元函数	友元函数	友元函数
	该类自己的函数	该类自己的函数	该类自己的函数
		子类的函数	子类的函数
			该类的对象

2. 函数 带默认值的参数

可以在调用函数的时候，不给那些带默认值的参数赋值，即调用函数时候，参数少了几个。

不过要注意的是，带默认值的参数，肯定都是从最右边的一个或几个。

3. 枚举变量的引用

类名：：某一枚举变量值

而不是C语言中直接引用：某一枚举变量值

4. 用new生成的变量与直接定义变量的区别

知道new要和delete配对使用。

直接定义变量：函数中的局部变量的话，在栈stack中分配，静态分配，函数调用完，自动释放内存，即编译器自己帮你释放内存，不需要你关心。

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

全局的话，应该和C语言中的没区别，就是普通全局变量，始终占用内容，会增加程序耦合度，在可以的情况下，应尽量少用。

用new生成的变量：动态分配，在堆heap中分配，所以要自己关心内存的使用，在不用此变量的时候，用delete释放内存，否则就产生了内存泄露，即有部分不再使用的变量存在于内存中，属于站着茅坑不拉屎的类型。

一般常见用法一直就是在类的构造函数中用new分配内存给某个变量，然后对应地在析构函数中用delete释放对应变量的内存。

5. 重载Overload

同一个函数名，参数的个数和类型不同。

6. 关于继承inherit

没啥多说的，只是自己不太熟悉，导致看到某个类B调用某个函数func1，结果在B类的定义中找不到，最后才发现原来是B类继承A类，A类中有public的函数func1，因此才有B可以调用func1。

3.4. 为何高级语言不能直接操作硬件？

TODO：整理[为什么说汇编语言可以直接操作硬件-CSDN论坛-CSDN.NET-中国最大的IT技术社区](http://bbs.csdn.net/topics/380163054)²到这里。

² <http://bbs.csdn.net/topics/380163054>

第 4 章 网络相关的语言

此处记录一些和网络相关的一些语言，比如HTML，XML等

4.1. XML

此处记录一些关于XML的学习心得和一些有价值的参考资料。

4.1.1. XML相关的参考资料

1. XML标准概览

目前已有的和XML相关的一些标准的概览，可参考：

- a. [XML 标准概览: 第 1 部分](#)¹
- b. [XML 标准概览: 第 4 部分](#)²

4.2. HTML

此处记录一些关于HTML的学习心得，以及一些有价值的参考资料。

4.2.1. HTML学习心得

1. HTML页面功能不正常，应该首先去检查对应的HTML源代码是否正常

在折腾[【记录】折腾给WordPress Google XML Sitemap中添加支持子目录自动扫描功能，包含HTML静态页面等可识别的格式](#)³的过程中，发现一个问题：

当（混了和php和javascript的）HTML页面中，某功能不正常，结果调试了很长时间，最终发现，原来是某段用于生成HTML代码的javascript代码，不小心多写了个双引号，导致生成的HTML代码不正常，功能因此也就不正常了。

但是在此过程中，没有任何的提示，因此也就不知道是此处错了，所以，最大的感慨是，对于HTML页面来说，如果功能上不正常的话，那么第一时间应该去看看与此相关的HTML代码是否正确。

2.

4.2.2. HTML相关的参考资料

1. HTML实体定义，特殊字符等资料

特殊字符方面，已有有人整理出来，把字符都显示出来，并且分门别类列出来了，很不错。

下面列出几个值得推荐的：

¹ <http://www.ibm.com/developerworks/cn/xml/x-stand1/index.html>

² <http://www.ibm.com/developerworks/cn/xml/x-stand4/>

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

- [HTML特殊字符大全](#)⁴

特点：中文的，分类很全

- [HTML Special Characters](#)⁵

特点：英文的，也分类了

2. HTML的颜色和对应的名字

- [HTML Color Chart with 140 Color Names](#)⁶

特点：关于HTML中的颜色和对应的名字，这个页面很简洁直观

- [HTML Color Names](#)⁷

特点：英文。这个页面的内容，非常全，更加不错

3. Unicode字符集

- [Unicode Table](#)⁸

特点：关于Unicode字符集的相关字符的表格，几乎包括了所有字符。

- [Unicode Lookup](#)⁹

特点：可以找到每个Unicode的各种进制的值之外，还有对应的HTML的entity

4. BIG5

a. BIG5码表

- [Big5 \(Traditional Chinese\) character code table](#)¹⁰

特点：单一页面，包含了所有的字符，方便查找。

- [Big 5 Code Table](#)¹¹

特点：多个页面，不是很方便查找。但是带有颜色区分，便于分辨各种类型的字符。

b. 简体中文和繁体中文之间的转换

[简繁转换](#)¹²

⁴ <http://www.qianduan.net/html-special-characters-daquan.html>

⁵ <http://brucejohnson.ca/SpecialCharacters.html>

⁶ <http://www.html-color-names.com/color-chart.php>

⁷ http://www.w3schools.com/html/html_colornames.asp

⁸ <http://www.tamasoft.co.jp/en/general-info/unicode.html>

⁹ <http://unicodelookup.com/>

¹⁰ <http://ash.jp/code/cn/big5tbl.htm>

¹¹ <http://www.khngai.com/chinese/charmap/tblbig.php>

¹² <http://www.j4.com.tw/big-gb/>

参考书目

Word VBA

- [1] [【整理】word vba学习记录](#)¹
- [2] [菜鸟谈VBA最最基础入门《原创》](#)²
- [3] [Word VBA 教程：Microsoft Word Visual Basic 参考](#)³
- [4] Word VBA内置的Help
- [5] [【解决】word VBA中,获取当前所选内容的标题序号/标题数字/heading level/heading number](#)⁴

其他

- [6] [BlogsToWordPress 的用法的举例说明](#)⁵

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

² <http://club.excelhome.net/thread-470603-1-1.html>

³ <http://d.download.csdn.net/down/1924671/taozx11>

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

⁵ http://www.crifan.com/crifan_released_all/website/python/blogstowordpress/usage_example/