谈 VBA 最最基础入门《原创》

如果你是学习 Excel 的朋友，对 VBA 这个概念不陌生吧？

VBA 的功能实在太强大，可初学的我们，面对那一串串尤如外星文的代码，你是不是看得头昏脑胀？想学，是不是却茫然得找不到入口？

在这里，我把我自己的学习心得和笔记放出来，和大家一起分享交流，一起学习，共同进步。对了，还要告诉大家，我也是初学者，不是高手，如果有什么说得不对的，希望大家及时指出。我的目的不是教学，而是期望创造一个有你，有我，有他的学习氛围！ 开始吧，记住一个共同的目标——掀开那层神秘的面纱，看看 VBA 的真实面目。

-----贴子内容---

1、什么是 VBA?为什么要学习 VBA

2、VBA 与宏

3、VBE 简介

4、对象、属性、方法、事件

5、VBA 过程

6、数据类型、变量、数组、常量

7、(回复)该定义变量为何种数据类型

8、VBA 的函数

9、程序流程控制:if 语句

10、程序流程控制: Select Case 语句

11.程序流程控制: For—next 语句

12、几个 for—next 循环的例子(作者:老朽

13、程序流程控制: For—Each 语句

14、程序流程控制: Do While 语句

15、程序流程控制: Do Until 语句

16、程序流程控制: Go to 语句

17、用户窗体

18、再说 Excel 的对象模型

19、[小试牛刀]制作一个个性化的欢迎界面

20、处理单元格数据(复制、剪切、删除

21、Range 对象的 End 属性(带作业)

22、花名册分类(实例)

23、花名册汇总(实例

24、新建工作表

25、认识数组(一)

26、Worksheet 的 Change 事件(带作业)

27、关于 Application.EnableEvents 属性(有实例动画)

Application.EnableEvents 补充说明

28、关于 Worksheet 的 SelectionChange 事件(有实例动画)

29、Worksheet 的 SelectionChange 事件实例(trustwxq 朋友提供)

什么是 VBA？为什么要学习 VBA？

=========

VBA 是 Visual Basic For Application 的简称，具体说来，可能让叶枫三天三夜也说不完，叶枫也只知道 VBA 是建立在 Office 中的一种应用程序开发工具，其实知道这也就够了。很懒很笨的叶枫只想吃香喷喷的大米饭，可从没想要知道大米饭是怎么来的。

Excel 本身的内置函数其实已经很强大了，甚至有人说，只要学会二三十个函数，就已经可以满足普通用户日常的工作需求。

那我们为什么要学 VBA？

是的，如果只需要满足一些普通的工作需求可能永远也不会用到 VBA，但在实际应用的过程中，人们的操作却越来越大，需要也越来越高，这时候就需要用 VBA 来对 Excel 进行二次开发了，VBA 可以有效地自定义和扩展 Excel 的功能。

但有一点叶枫想要告诉大家，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。

VBA 与宏

= = = = = =

刚开始的内容听起来很枯燥，请大家不要分心，耐着性子看下去，兴趣总是慢慢积累的。

1、亲自动手，录制一段属于自己的宏

提到VBA，相信很多人想到的就是宏，但是VBA与宏是不是一回事？我觉得它们不是一回事，但这不重要，我们要做的是先来认识一下宏。

宏是什么？

我理解的宏是一系列固定动作的集合，这个集合当遇到让他执行的条件后就逐个执行。

有人把我们录制的宏比喻成一个武术运动员在比赛中练就的一段武术套路动作，这个套路遇到一定的条件（比赛）就执行。

好了，别闲着，动一动你可爱的右手，点点鼠标，打开一个Excel文件，录制一段宏来来研究研究，步骤如下：

（1）选中任意一个单元格；

（2）打开菜单—>工具—>宏—>录制新宏，调出宏录制对话框，输入宏名（mysub），确定；

（3）设置单元格字体为仿宋体，红色；

（4）点击停止录宏。

这样，一个简单的设置单元格格式的宏就录好了。

2、鼠标点一点，宏给单元格换新装

下面我们来执行这段宏。

（1）选中需要设置的任意单元格或单元格区域（为了看出效果，单元格最好有字符);

（2）打开菜单—>工具—>宏—>宏，显示“宏”对话框（快捷键：Alt+F8）；

（3）选择我们刚才录制的宏，单击“执行”。

这样我们看到已经为刚才选中的单元格的字体已被修改成仿宋体，红色。自己动手试着录制几段宏，再执行一下，加深印象。

神奇了吧？以后当你再需要一遍又一遍地重复若干相同的操作的时候，会不会想到让宏来替你做这一切？这是一个不错的奴隶，不用花钱，且不用开工资，赚了吧？

好了，收起你贪婪的笑容，继续往下看。

宏是什么？现在相信你能想出一千种说法，由你了，又笨又懒的叶枫可没你聪明，也不再多说。

上面说的执行宏的过程是不是让你感觉很麻烦？希望用快捷键来控制宏？这个我们可以在录制宏之前在录制宏对话框里进行设置。也可以在录制宏后进行设置，按 Alt+F8 显示宏对话框，单击“选项”，在“宏选项”对话框里进行设置

3、寻根问底，找到宏的老窝

如果你忘记了，请你再打开宏录制对话框，在宏名的下面，有一个“保存在”的选项，点一点下拉的箭头，我们可以看到，宏可以保存在三种不同的位置，分别是:

（1）当前工作簿（系统默认）: 宏保存在当前工作薄的模块里，只有当该工作薄打开时，宏才起作用。

（2）新工作簿: 新建一个工作薄保存。

（3）个人宏工作簿: 这是为宏而设计的一种特殊的具有自动隐藏特性的工作簿，如果你要让某个宏在多个工作簿都能使用，那么就应当创建个人宏工作簿，并将宏保存于其中。

上面我们录制的宏是保存在当前工作薄里，今天我们在这里也只介绍保存在当前工作表里的宏。

4、拨开层层迷雾，查看庐山真面目

宏靠什么来控制 Excel 的运行？是不是感觉深在浓浓的迷雾里，感觉一头雾水？让我们鼓一鼓劲，拨开它的衣服，开开那家伙里面到底是什么？

（1）按 Alt+F8 快捷键打开“宏”对话框;

（2）选择我们刚才录制的宏，点“编辑”按钮。

眼前一亮吧？这是一个新的天地——VBA 的编辑器窗口（VBE），这个我们以后再说，先将注意力集中到显示的代码上。代码如下:

复制内容到剪贴板

代码:

```
Sub mysub()
' mysub Macro
      ' 宏由 ggsmart 录制，时间: 2009-2-27'
    With Selection.Font
        .Name = "仿宋_GB2312"
        .Size = 12
        .Strikethrough = False
        .Superscript = False
        .Subscript = False
        .OutlineFont = False
        .Shadow = False
        .Underline = xlUnderlineStyleNone
        .ColorIndex = xlAutomatic
    End With
Selection.Font.ColorIndex = 3
End Sub
```

现在你可能感觉到上面这些乱七八糟的代码会很陌生，甚至在问自己是不是来到外星人的世界，反正叶枫当初是这样感觉的，但不要为此感到害怕，将来我们会慢慢熟悉他，就像闭上眼睛也能准确地将那个又大又红的苹果送到自己的嘴里。

5、再进一步，给宏安上门铃

一遍一遍地在宏对话框里选择宏名称，按“执行”按钮来运行宏的麻烦是不是让你郁闷得有些想吐血？指定快捷键虽然方便，但别人用自己的工作表时能不能快速上手？就算是自己，如果宏多了，或是放的时间长了，你还记得哪个快捷键控制哪个宏？

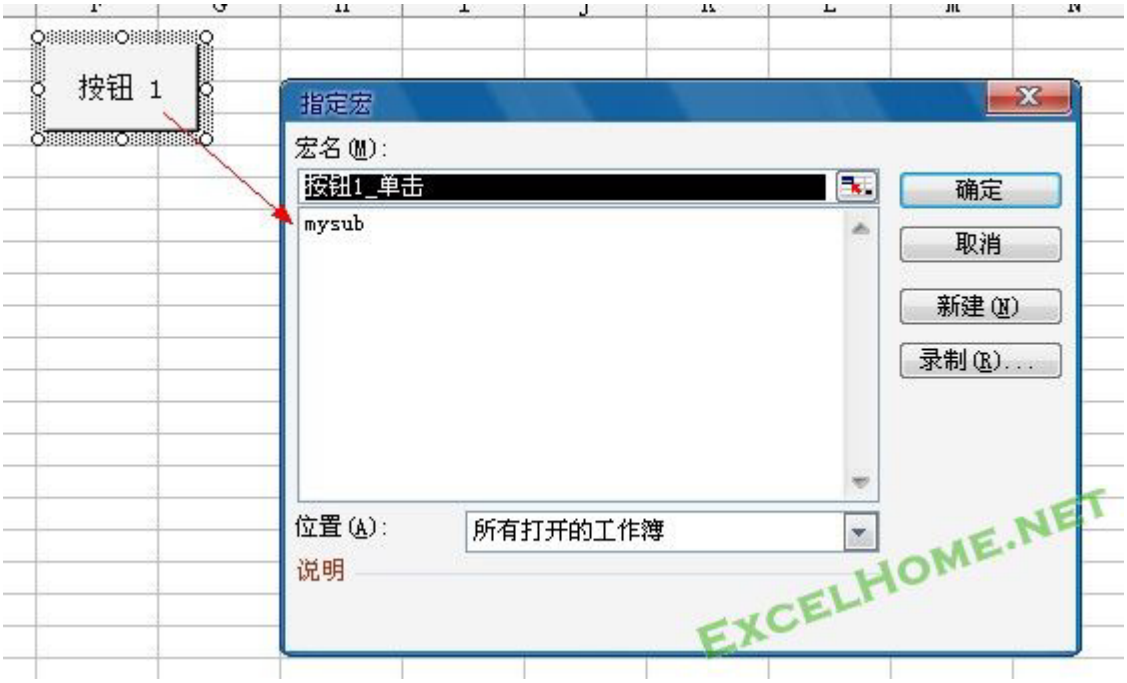
作为 excel 开发者，一个主要的目标是为自动化提供一个易于操作的界面，“按钮”是最常见的界面组成元素之一，我们可以把宏指定给特定的按钮，通过按钮来执行宏，还是用刚才我们录制的那个宏来举例。通过使用“窗体”工具栏，可以为工作簿中的工作表添加按钮。在创建完一个按钮后，可以为它指定宏，然后你的用户就可以通过单击按钮来执行宏。在本练习中，将创建一个按钮，并为它指定一个宏，然后用该按钮来执行宏。具体步骤如下:

(1)打开菜单—>视图—>工具栏—>窗体，调出窗体工具栏。



1.jpg (7 KB)

（2）单击”窗体“工具栏中的“按钮”控件，在工作表中希望放置按钮的位置按下鼠标左键，拖动鼠标画出一个按钮，松开鼠标后，Excel 会自动显示“指定宏”对话框。



2.jpg (31.38 KB)

（3）从“指定宏”对话框中选择“mysub”，单击“确定”。这样，就把该宏指定给命令按钮。

引用:

课间休息，插播一段广告：按钮,工作表等这些对象就像是 Excel 的众多儿女,儿女太多,怎么称呼这群儿女也是一个问题. 为了不出现叫" 儿子" 就七八个一起跑出来，同许多农村父母给儿女起" 老大，老二，老三……" 的称呼一样， Excel 也用这种顺序的起名方法给这些儿女贴上了标签. 比如第一个按钮

是" 按钮一" ，再画一个就是" 按钮二" ，然后是" 按钮三" ，" 铵钮四" 等等. 这样，以后你要叫第三个儿子的时候直接叫" 按钮三" 即可. 如果你嫌这个名字乡下味太浓，如果你嫌这个名字不能很好地反映点击它后会执行怎样的操作不清楚，你可以通过双击按钮表面激活它，可重新为按钮贴上合适的显示标签. 需要提醒你的是，这里的标签只是它显示在你面前的样子，实际上它是什么名字？我们可以点击它，看一下名称框里是什么.



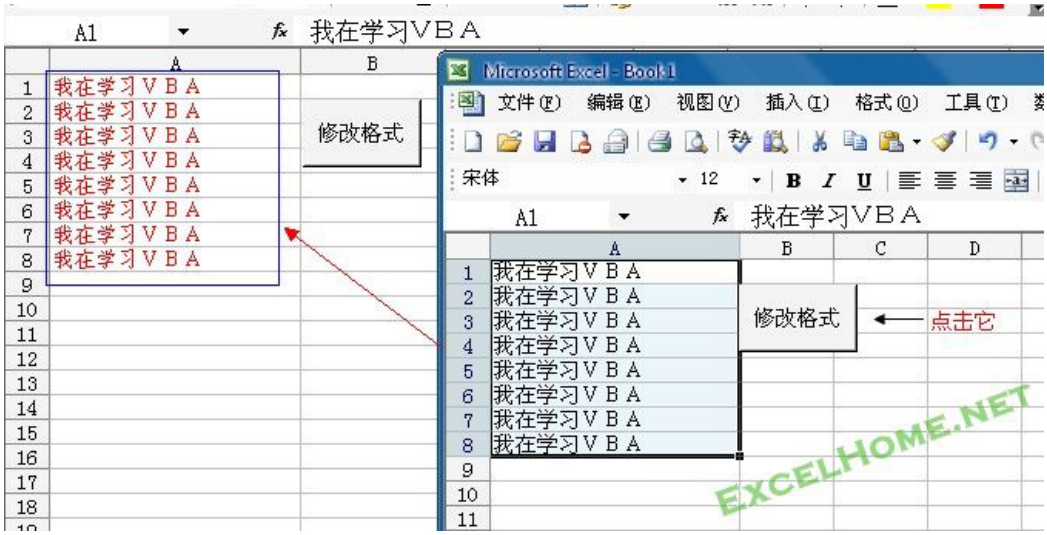
3. jpg (16.03 KB)

为什么显示出来的是一样，而实际的名字却是另一样呢？也许你现在会感到迷茫，但不用管它，以后当我们学习了属性后，一切就自然明白了.

下面我们来试一试用按钮操作:

（1）选择单元格或单元格区域;

（2）单击按钮。



4.jpg (61.93 KB)

按钮就像装在楼下的门铃，美丽的嫦娥仙子家住在 502，猪八戒来到楼下，找准号码 502，轻轻一按，“叮咚，叮咚……”，嫦娥在楼上就给八戒开门了，八戒不用在楼下大声地喊“嫦娥，开门。”也不用花两毛钱给嫦娥打个电话，当然更不用顺着下水管从墙上爬进嫦娥的窗户，方便而实在吧？

这种遥控式的命令的确能让繁琐的操作变得简单而方便，让我们获得了一些使用 Excel 标准命令所不能实现的功能。如果你是八戒，当你熟悉了如何使用这种遥控一样的门铃以后，你可能都会奇怪自己当初在没有门铃的情况下，那段漫长的爬下水管道的日子是怎么熬过来的。

让嫦娥仙子开门的方法有很多种，同样执行宏也有多种方法可以选择，我们还可以把宏指定给图片，自选图形，这些比较简单，和指定给窗体按钮是一样的，同时我们还可以把宏指定给某个“事件”，比如单击工作表，双击工作表，激活工作表，打开工作簿等等，你要安什么要的门铃，选什么样的音乐，随你了。

需要强调一点的是，我们刚说的“事件”是一个重要的概念，八戒按门铃，就是一个“事件”，这个“事件”引发了嫦娥的开门，只有门铃响了，嫦娥才会去开门。除此而外“方法”“对象”“属性”都是接下来我们会经常接触到的，慢慢走，留心点，一路上将会精彩不断。

6、小结

到这里，我们对宏应该有一个简单的了解了吧？宏实际上就是一个简单的 VBA 的 Sub 过程，它保存在模块里，以 Sub 开头，以 End Sub 结尾，执行时就从第一句逐句执行，直到 End Sub 结束。就像前面我们说的武术运动员练的那段套路动作，总是抱拳，扎马，出拳然后踢腿，永远不可能没有扎马就出拳，没有出拳就踢腿。我们今天录的这段宏也总是先设字体，再改字体颜色，这种顺序永远不会乱。

说完这些，你是不是感觉这些操作古板得没有生气？是不是感觉这宏的操作不能满足自己的需求？

叶枫想告诉你的是，宏代码绝不等于 VBA，它只是 VBA 里最简单的运用，尽管许多 Excel 过程都可以用录制宏来完成，但是通过宏代码还是无法完成许多的工作，如:

（1）不可以建立公式，函数;

（2）没有判断或循环的功能;

（3）不能进行人机交互;

（4）无法显示用户窗体;

（5）无法与其他软件或文件进行互动。

…….所以，你看了上面的内容后千万不要以为宏就是 VBA 了，更不要以为自己已经会 VBA 了，事实上是你才刚沾上 VBA 的边，或者说连边都没沾上，甚至连宏今天我们都只是作了简单的介绍。

路漫漫其修远兮，还等着我们上下去求索，路在脚下，武林高手总是从练习简单的套路动作开始，不要厌倦扎马的单调，静下心多练练，这会是你以后练习武功招式的一个基础，孔子老人家说了，要温故而知新，记得复习巩固，不要学了后面忘了前面，到下次叶枫再见到你的时候你什么都忘记了，好了，今天就到这里，让我们一起加油——

VBE 简介==========

什么是 VBE?

相信你还没忘记吧？VBE 就是 VBA 的编辑窗口，所有的 VBA 操作都在 VBE 里完成。

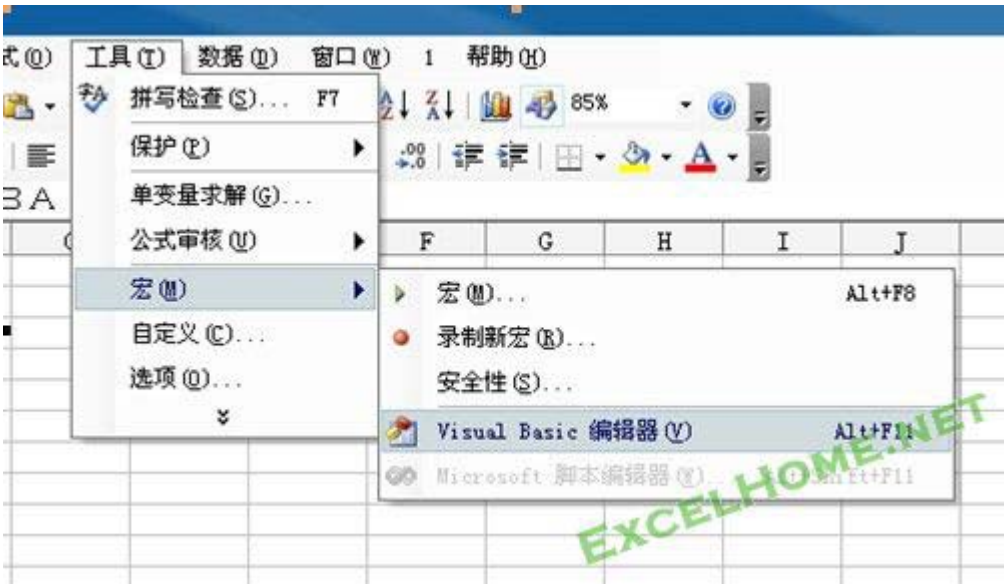
VBE 是一个分离出来的应用程序,它可以与 Excel 无缝结合，但是需要说明的是要运行 VBE 必须先运行 Excel，VBA 模块与 Excel 工作薄文件一起存储，除非激活 VBE，否则 VBA 模块是看不见的。

1、运行 Excel 后，怎样切换换到 VBE 窗口？

打开大门的钥匙有很多把，千万别猴急地去爬下水管道。

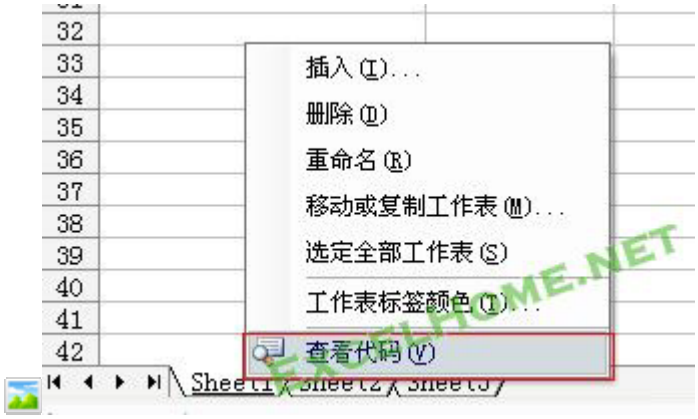
（1）按 ALT+F11 快捷键;

（2）选择“工具”—>“宏” —>“Visual Basic 编辑器”命令。



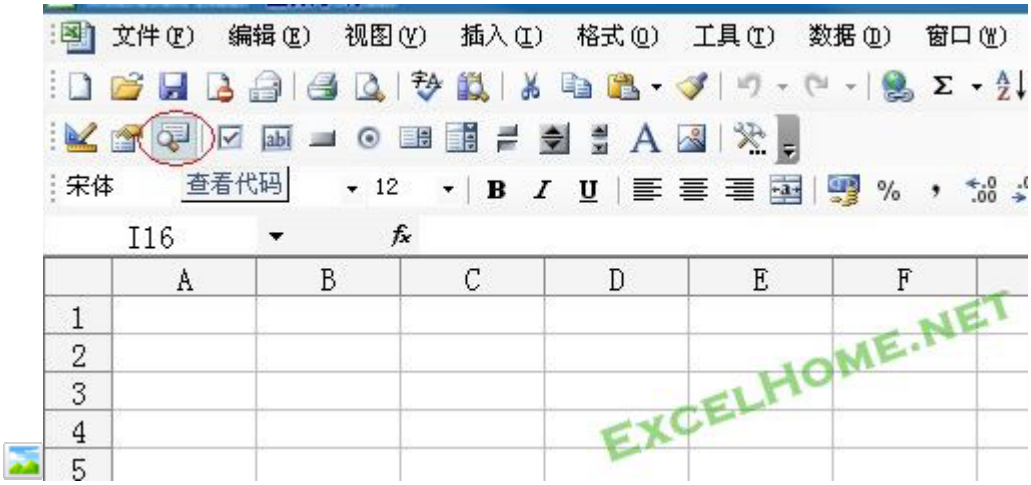
1.jpg (29.14 KB)

（3）右击工作表名称标签，点击查看代码。



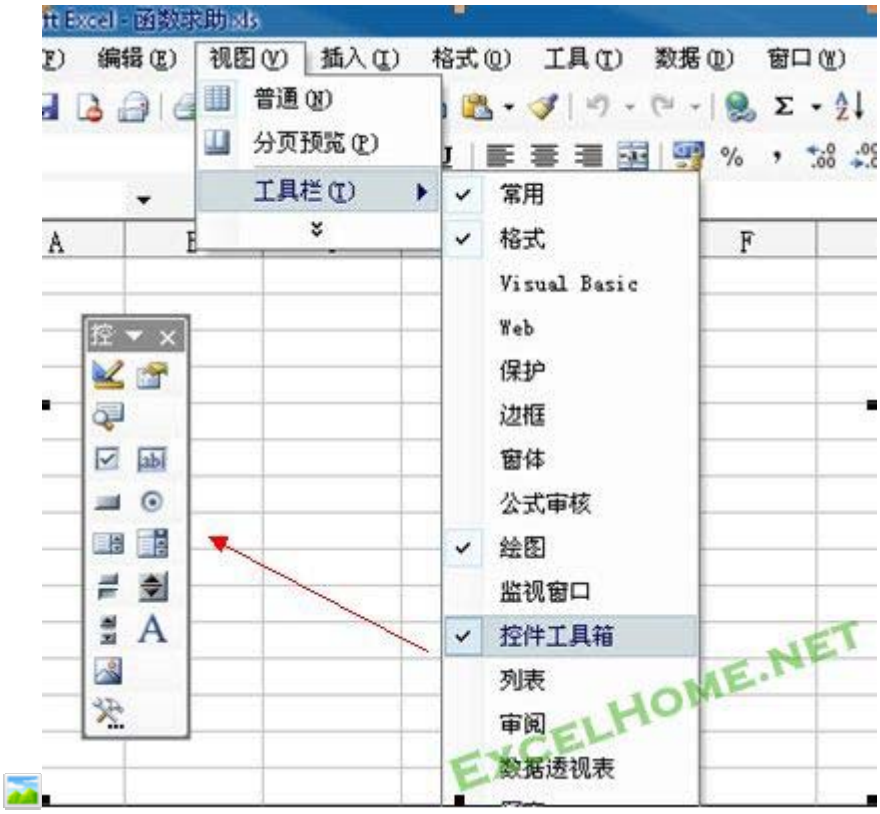
2.jpg (17.69 KB)

（4）单击控件工具箱里的“查看代码”。



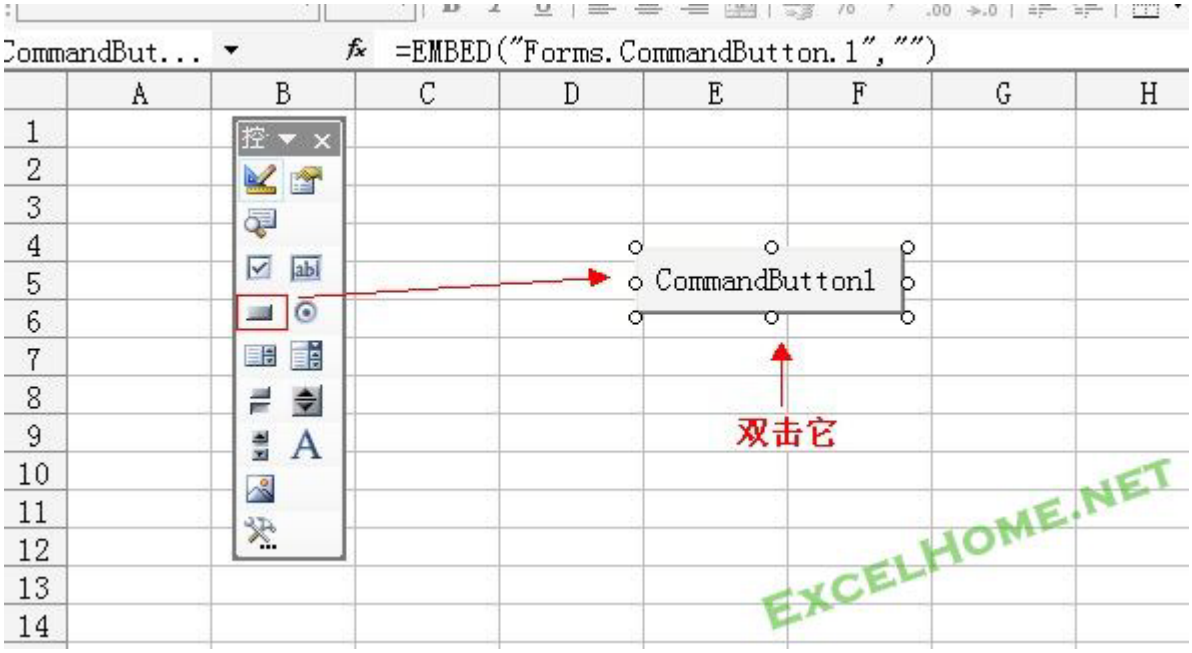
3.jpg (28.24 KB)

如果你的窗口里找不到控件工具箱，请通过“视图”—>“工具栏” —>“控件工具箱”打开它。



4.jpg (35.59 KB)

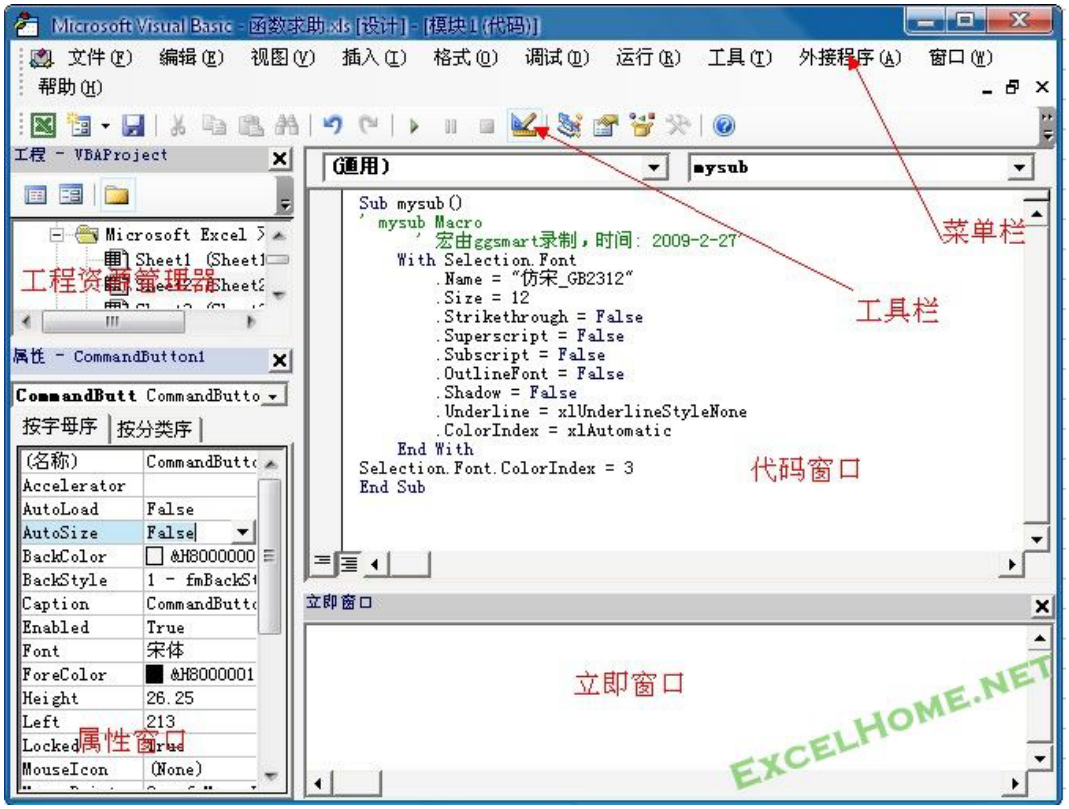
（5）通过控件工具箱建立一个新的控件，双击控件。



5.jpg (36.24 KB)

2、初识 VBE 窗口

这个界面相信你不陌生吧？查看宏的时候我们已经见识过它了。



6.jpg (85.09 KB)

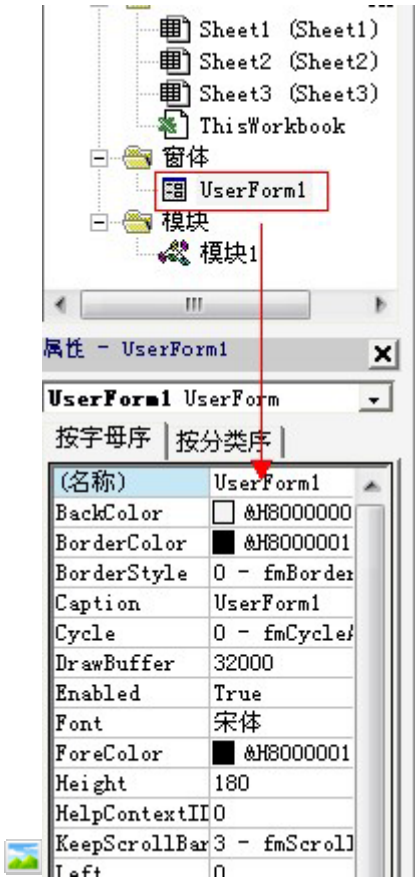
“工程资源管理器”显示一个树型图示,包含了当前在 Excel 中打开的所有 Excel 对象，包含工作表，模块，窗体，加载宏及隐藏的工作薄，每个工作薄被认为是一个工程。

在工程资源管理器里右击，可以在右键菜单里选择相应的命令插入模块或窗体。如果你想删除它或者保存它，同样也可以在这里进行相应的操作。



7.jpg (30.89 KB)

“属性窗口”显示当前你选择的 Excel 对象的属性。选中某个对象后，可以在属性窗口中修改选中的对象的的各样属性. 如颜色，名称等。

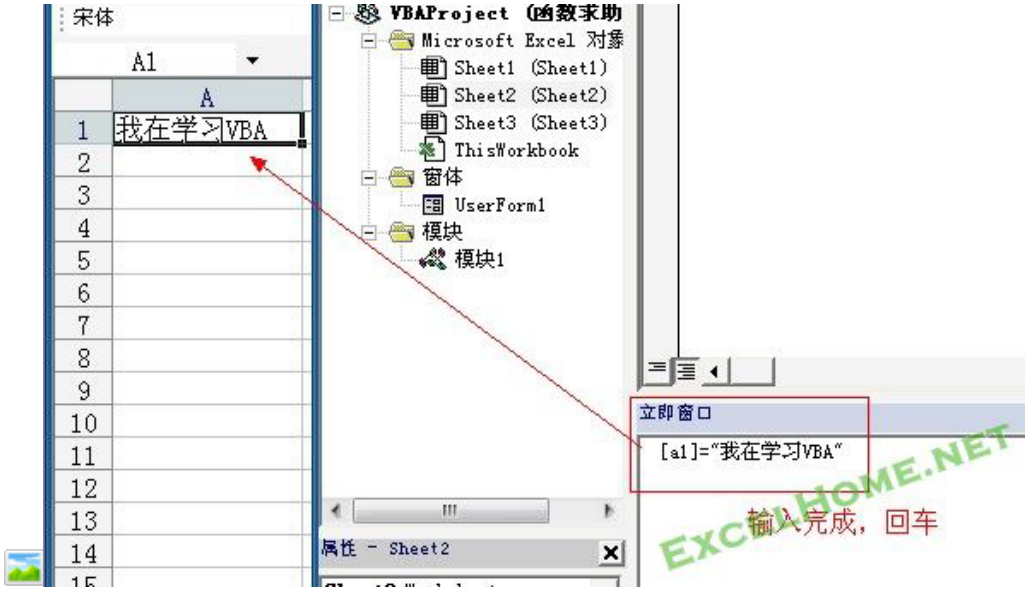


8.jpg (37.77 KB)

“立即窗口”，这是一个非常有用的窗口，在其中可以直接执行 VBA 语句，测试语句和调试代码，就跟我们在 DOS 下输入 DOS 命令一样。如果你的立即窗口不可见，按 CRTL+G 调出来，可以在里面尝试输入:

[a1]=”我在学习 VBA”

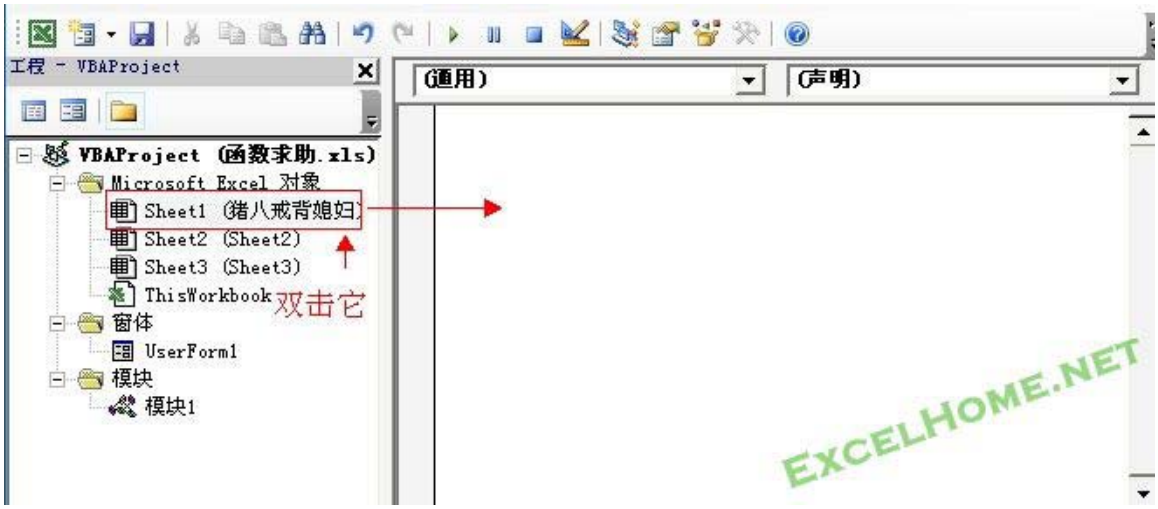
回车，看一看 A1 单元格里发生了什么变化?



9.jpg (36.51 KB)

“代码窗口”是干嘛用的？相信你已经很清楚了，当然是用来编辑 VBA 代码的地方，工程中的每一项都有一个与之相关联的代码窗口，

如果要查看某对象的代码窗口，或者说如果要查看某对象上面究竟编写了什么 VBA 代码，在“工程资源管理器”窗口中双击对象即可。



10.jpg (30.07 KB)

当然，你打开后的窗口不一定和上面一样，VBE 的窗口也不只上面几个，我们可以打开“视图”菜单，在里面进行选择需要显示的窗口。



对象、属性、方法，事件

＝＝＝＝＝＝＝＝＝＝＝＝＝＝

这是很重要的几个概念，是写 VBA 程序的基础，大家仔细读，一定要弄清楚。

1、对象及对象层次结构

对象就是存在的东西，是 VBA 处理的内容，包括工作薄、工作表、工作表上的单元格区域、图表等等。对象可以相互包含，

就像一个文件夹里可以包含多个文件夹一样，而这个文件夹又可以被其他的文件夹包含，一个工作薄对象可以包含多个多个工作表对象，一个工作表对象又可以包含多个单元格(或图表,图形等)，这种对象的排列模式称为 Excel 的对象模型。

集合这个概念经常遇到，我理解的集合是对相同类型的对象的统称。

某中学有 5000 个学生，名字各不相同，我们对他都统称为某中学的学生，而不和一一去叫他们的名字，"某中学的学生"在这里是集合。

对象的引用,在 Excel 里,Workbooks 集合包含在 Application 对象里,当我们要引用某工作薄的时候,要遵循从大到小的规则。跟表示硬盘里的某个文件的位置一样，比如我们想引用 D 盘“我的文档”文件夹下的名为“我的 VBA 课程.doc”文件时要输入的是:

D: \我的文档\我的 VBA 课程.doc

同样，如果我们要引用名称为“mybook.xls”的工作薄时就是:

Application.Workbooks(“mybook.xls”)

和引用文件不同的是，VBA 里使用的分隔符是点。

同理，当我们引用“mybook.xls”里面的工作表“mysheet”时应是:

Application.Workbooks(“mybook.xls”).Worksheets(“mysheet”)

可以继续延伸到下一层，引用“mybook.xls”里工作表“mysheet”里面的单元格区域“A1:D10”:

Application.Workbooks(“mybook.xls”).Worksheets(“mysheet”).Range(“A1:D10”)

但是并不是每一次引用都必须这么呆板，就像猪八戒不用每一次都去爬下水管道。如果我们引用的是活动对象，也就是被激活的对象，引用就可以进行简化。

如果是 mybook 工作薄是激活的，引用可以简化为

Worksheets(“mysheet”).Range(“A1:D10”)

如果 mysheet 当前也是激活的，引用甚至还可以简化为 Range(“A1:D10”)，也可以直接输入[A1:D10]，如果引用的单元 Range 是单个的单元格，还可以用 Cells（行号,列号）的引用方式。

2、属性

每一个对象都有属性，一个属性就是对一个对象的一个设置。

猪八戒背着媳妇高秀兰回自己的紫云洞，猪八戒的媳妇就是对象，“高秀兰”就是猪八戒的媳妇的一个属性（name 属性），引用对象的属性同样也要用点来分隔。

猪八戒的媳妇.name = 高秀兰

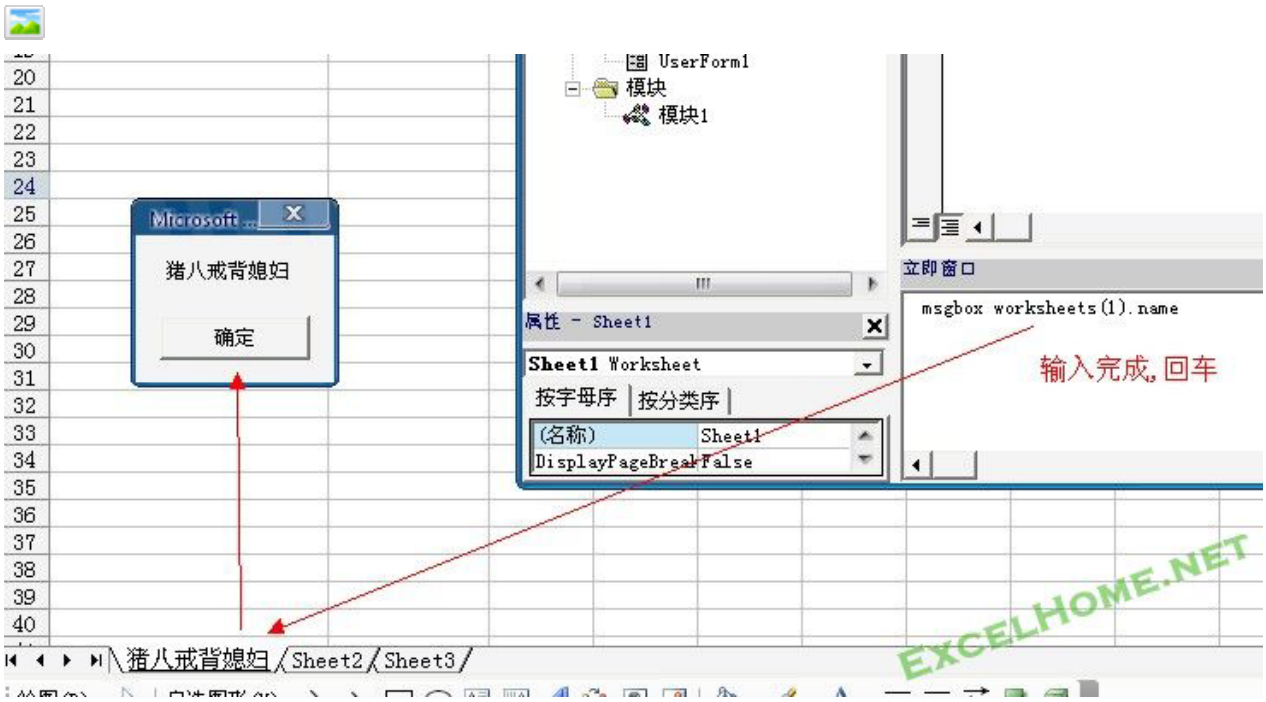
别闲着，还是动动你的右手，打开一个工作表，Alt+F11（千万别说你不知道这个快捷键是干什么，要不我保证被你气个半死），如果立即窗口没有打开，按 Ctrl+G 打开，在里面输入：

复制内容到剪贴板

代码:

```
Msgbox Worksheets(1).name
```

回车。



1.jpg (53.4 KB)

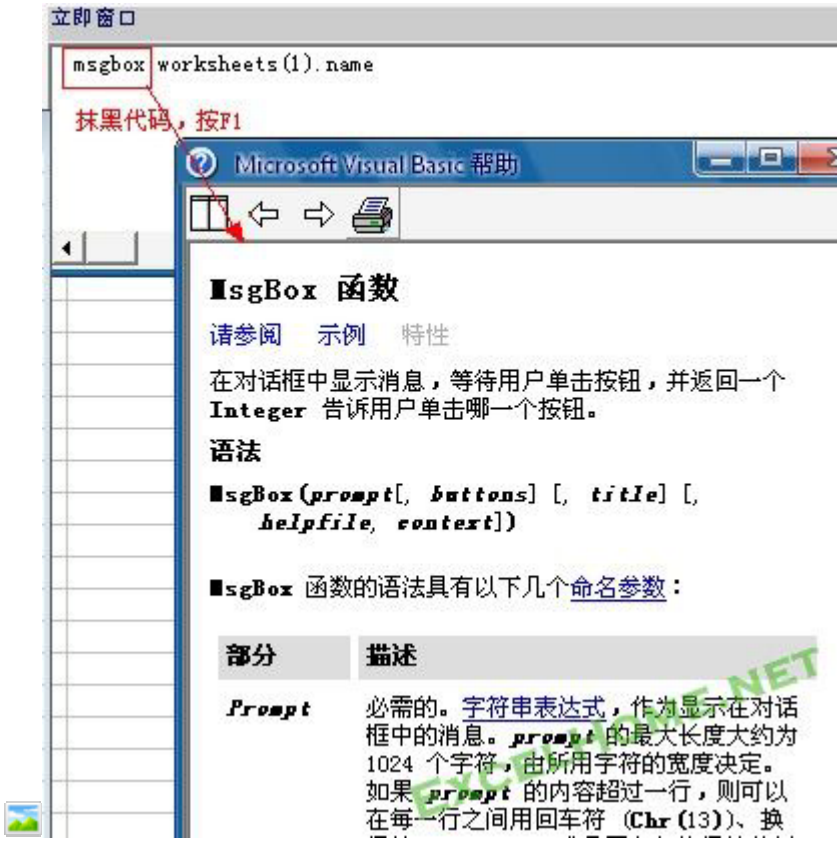
Worksheets(1)和 Worksheets(“sheet1”)有什么区别？

Worksheets(1)表示 Worksheets 集合里的第一个工作表。

Worksheets(“sheet1”)表示 Worksheets 集合里名为"sheet1"的工作表。

至于 Msgbox 是什么，那就自己问 Help 了，在立即窗口里用鼠标左键把 Msgbox 抹黑，按 F1，就弹出 Help 里对它的说明了。

抹黑代码，再按 F1 即可看到相应的帮助。叶枫说：“这个办法很适用，一般人我不告诉他。”



2.jpg (42.52 KB)

一个对象有哪些属性我们可以在属性窗口里查看，要修改一个对象的某种属性，如名称、显示状态、颜色等等，也可以在属性窗口里进行修改，当然我们还可以利用代码进行修改。改当前工作薄里的第三个工作表的名称为“这个名字是我用 VBA 改的”，想想怎么写代码？

把你的代码输在立即窗口里，回车，看看效果，你做到了吗？

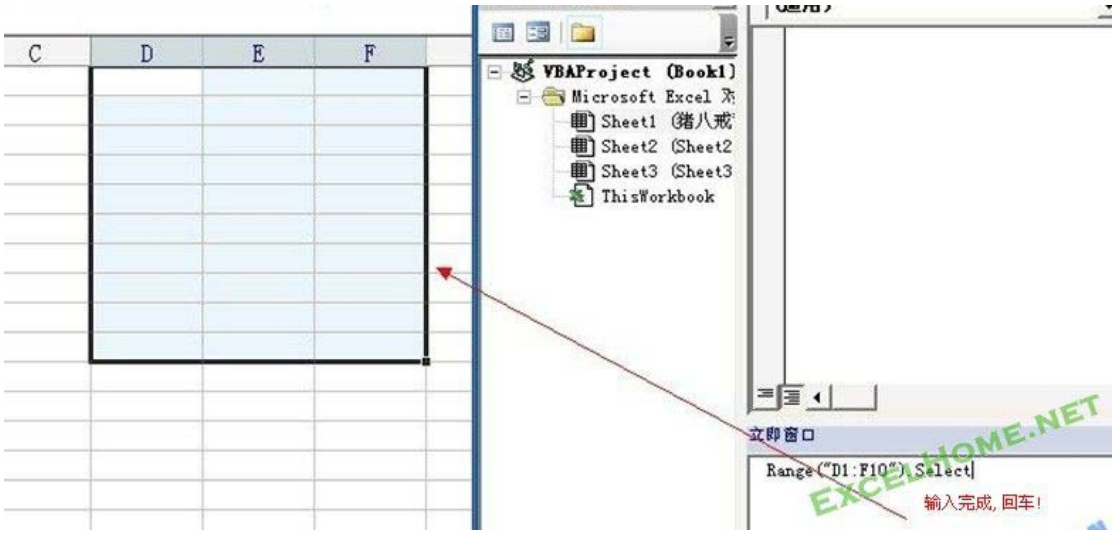
3、方法

每一个对象都有方法，方法就是在对象上执行的某个动作。

和属性相比，属性表示的是对象某种状态或样子，是静态的，就像是语文里的名词、形容词和副词，而方法则是做某件事的一个动作，就像动词，对象和方法同样用点来分隔。

例如 Range 对象有有一个方法是 Select，他的作用是选中指定的 Range（单元格区域）对象，在立即窗口里输入代码:

回车，可以看到 D1:F10 已经被选中了。



3.jpg (36.23 KB)

4、事件

在上一面猪八戒按门铃那里，我们已经接触过什么是事件了。

简单点说，事件就是由用户或者系统触发的，可以在代码中响应的一段代码。比如，当我们移动鼠标，打开工作薄，激活工

作表，选中单元格，改变单元格的数值，点击按钮或窗体，敲击键盘等等这些都会产生一系列的事件，通过编写代码响应这些事件，当发生此类事件时，程序代码就会进行相应的操作。

这样解释你会不会有些晕？感觉又一次进入了外星人的世界？还是举例说明，我们需要当激活某工作表的时候，自动弹出一个对话框，告诉我们激活的工作表的名称。

（1）打开一个工作表，Alt+F11 打开 VBE 窗口，在“工程对象管理器”窗口里双击你要进行设置的工作表，使其代码窗口显示。

（2）左面选择对象 Worksheet(工作表对象)，右面选择 Activate 事件，我们可以看到在代码窗口里系统已经为我们自动输入了一段代码。

复制内容到剪贴板

代码:

```
Private Sub Worksheet_Activate()
End Sub
```

引用:

提醒：初学的我们，不必完全记住对象及事件的名称，也不必手工输入，系统早为你准备好了，你可以在代码窗口里进行选择，左边是对象，右面是事件，如果你想知道某个对象（例如工作薄、工作表、窗体等）有哪些事件，只需要双击这个对象，然后在代友窗口里查看即可。我们需要做的只是:当自己需要它的时候，

知道打开哪个箱子把它拿出来即可.



4.jpg (47.51 KB)

废话说完，再回到问题里:

要达到问题的目的，我们只需要在已给我们列出的两段代码中间加入需要进行操作的代码就可以了，这里我们需要的是一个对话框来提醒，对了，还记得 Msgbox 吧?

输入代码:

复制内容到剪贴板

代码:

```
MsgBox "你现在激活的工作表名称是: " & ActiveSheet.Name
```

上面的代码相信你应该能看懂吧?

&和我们工作表里的函数是一样的，连接文本的作用，Activesheet.name 是当前活动工作表的名字（用了一个 name 属性）。

回到工作表，激活你刚才设置代码的工作表，如果你刚才设

置的工作表是激活状态，请选择其他工作表，然后再重新激活它，看看你看到了什么？



5.jpg (34.5 KB)

看一下上面的代码，对象和事件之间用什么来分隔？还是不是点？

千万别懒，一定要动手，换其他的事件或其他的对象试一试，试着用代码改一下其他对象的属性，在单元格里添加点什么东西，这些随你了，你可千万别说你不知道哪些事件是干嘛用的。

别忘记，抹黑代码，按 F1，再重复一遍，这是一件好武器，一般人我不告诉他.

VBA 过程

＝ ＝ ＝ ＝ ＝ ＝ ＝ ＝ ＝ ＝ ＝ ＝ ＝ ＝ ＝ ＝

一个过程就是执行某些动作的代码组合。

VBA 过程分 Sub 过程和 Function 过程。

1、Sub 过程总是以“sub 过程名()”开头，以“End Sub”结尾，一个过程就是执行某项动作的一套指令，Sub 过程不返回运行的结果。

2、Function 总是以“Function 程序名()”开头，以“End Function”结尾，和 Sub 过程的区别是 Function 过程返回程序运行的值，值可以是一个值或一个数组，就像我们的工作表函数，Function 过程也就是我们说的自定义函数。

在这里，叶枫主要要给大家讲的是 Sub 过程，Function 过程如果需要的话，在后面我们再讲。

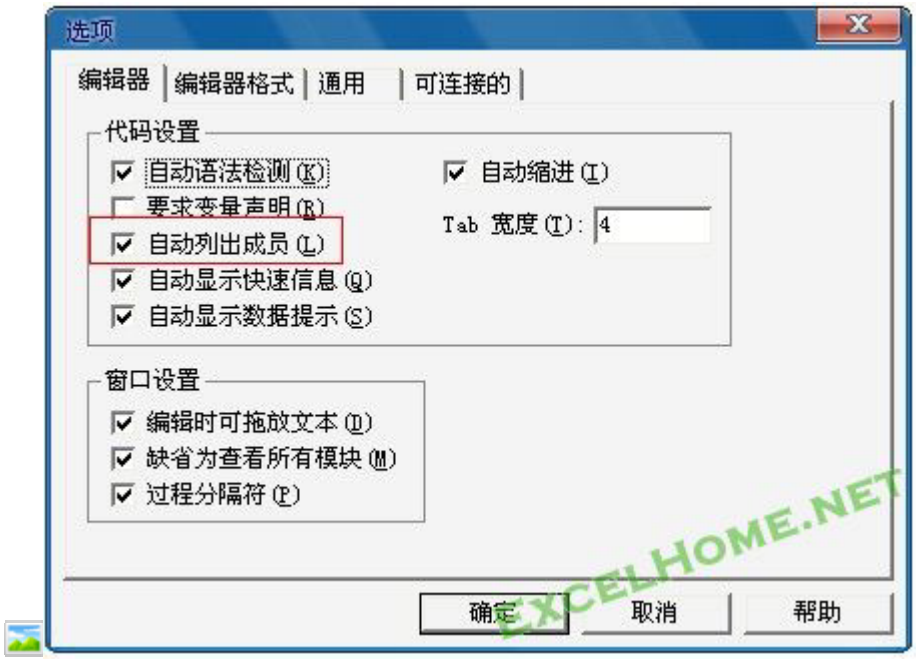
好了，今天已经讲得够多了，你对 VBA 应该已有一个简单的认识了吧？

要设计一个过程，其实很简单，把需要的对象收集起来，看看我们要对他们的属性进行怎样的修改？我们需要做什么？需要用什么样的方法？要用哪些事件？收集起来就是一个完整的过程。

只要你肯动手，一定会觉得很简单，你不要被那一大堆的对象、属性、方法和事件给吓晕了，完全不必全部记住它们，需要的时候知道在哪里找就行，而代码也不必一个一个的去输入，还记得前面一贴我们说的录制宏吗？当你需要进行某项操作的时候，不妨先录一段宏，但是宏是呆板的，有许多我们不需要的

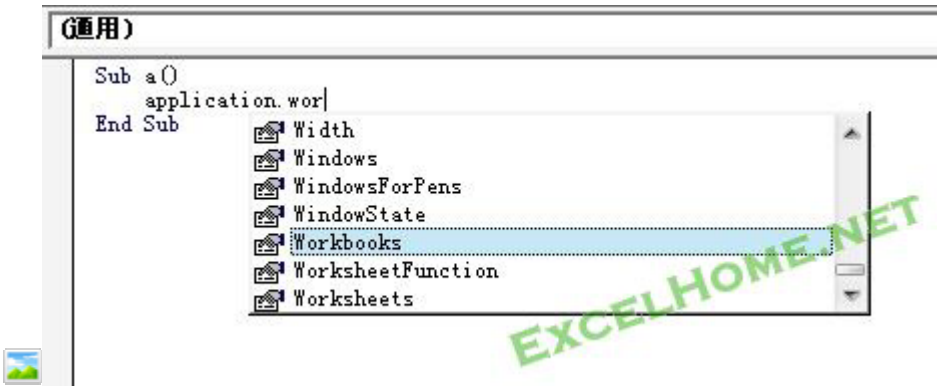
东西，给它修修枝，剪剪叶，整理整理，一个程序就 OK 了。

同时，建议你在 VBE 窗口中，单击“工具——选项”，在弹出的“选项”对话框里勾选“自动列出成员”。



1.jpg (32.45 KB)

设置好后，试着在代码窗口里输入点代码，看看，有什么？呵呵……，不用我说了，对于初学的我们，很需要它吧？



2.jpg (15.69 KB)

了解了对象，属性，方法及事件后，猪八戒按门铃开门的过程我们都可以把它写成程序：

```
Sub 门铃_单击（）
    如果 嫦娥.位置 = 在家 那么 门.开
End sub
```

上面这个并不是真的程序，但都有了程序的思想了，程序里，事件、对象、属性及过程都有了，你能找出来吗？

多尝试，温故知新，你可以想着用程序去解决一些你在使用 Excel 过程中遇到的问题,每一个问题的解决,你都会发现自己得到了很大的进步.

好了，还是那句话，路在脚下，继续走，精彩会慢慢为你上演，让我们一起努力，共同进步。

在上面,我们简单介绍了 VBA 的开发环境、几个常用的窗口，并对对象、对象的属性、对象的方法、对象的事件等概念作了简单的介绍,贴子大家认真看了吗？有没有一点点的收获？学习成绩如何？

什么是对象？什么是集合？什么是属性？什么是方法？什么是事件？如何正确地表示它们？

这些，你都能回答了吗？

先沉思三秒钟，如果你不能回答，那请赶快回过头去，再看一遍，我们说 VBA 是采用面向对象的程序设计方式，这些都是很重要的概念，如果不弄清楚，实际应用时，你可能会感到手忙脚乱。

如果已经掌握了，那让我们一起继续本贴的内容。

这次，我们将继续单调的基本功练习，讲一些 VBA 里面的关键字，为程序设计打下基础，还是那句话，希望你不要嫌内容的单调，一口你永远也吞不下一个大胖子，学习是一个循序渐进的过程。

数据类型、变量、常量、数组

＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝

VBA 的主要目的是什么？不用说了你也知道，当然是处理数据。某些数据存在于对象中，如工作表的单元格区域里，某些数据存在于我们自定义的变量中。

1、数据类型

数据类型告诉计算机如何将数据存储在内存中，如以整数、字符串、日期等。

数据类型是变量的特性，数据类型包括:

数据类型	存储空间大小	范围
Byte	1 个字节	0 到 255
Boolean	2 个字节	True 或 False
Integer	2 个字节	-32,768 到 32,767
Long(长整型)	4 个字节	-2,147,483,648 到 2,147,483,647
Single (单精度浮点型)	4 个字节	负数时从 -3.402823E38 到 -1.401298E-45; 正数时从 1.401298E-45 到 3.402823E38
Double (双精度浮点型)	8 个字节	负数时从 -1.79769313486232E308 到 -4.94065645841247E-324 ; 正数时从 4.94065645841247E-324 到 1.79769313486232E308
Currency (变比整型)	8 个字节	从 -922,337,203,685,477.5808 到 922,337,203,685,477.5807
Decimal	14 个字节	没有小数点时为 +/-79,228,162,514,264,337,593,543,950,335, 而小数点右边有 28 位数时为 +/-7.9228162514264337593543950335; 最小的非零值为 +/-0.0000000000000000000000000001
Date	8 个字节	100 年 1 月 1 日 到 9999 年 12 月 31 日
Object	4 个字节	任何 Object 引用
String (变长)	10 字节加字符串长度	0 到大约 20 亿
String(定长)	字符串长度	1 到大约 65,400
Variant(数字)	16 个字节	任何数字值，最大可达 Double 的范围
Variant(字符)	22 个字节加字符串长度	与变长 String 有相同的范围
用户自定义 （利用 Type）	所有元素所需数目	每个元素的范围与它本身的数据类型的范围相同。

2、变量

变量是用于保存在程序运行过程中需要临时保存的值或对象。就相当于我们在操作工作表的时候插入的辅助单元格、辅助列或辅助表一样。

同工作表的单元格一样，变量可以接纳很多种的数据类型，如其名，程序运行后，变量的值是可以改变的。

如何定义变量?

定义变量可以使用 Dim 语句:

Dim 变量名 As 数据类型

变量名有一定的命名规则，这里我就不详细说明了，需要提醒的是在程序的设计过程中，你可能会定义很多的变量，为了阅读及修改程序的方便，你定义的变量名尽量能让人一看就明白这个变量具体代表的是什么。在 Dim 语句中，我们可以不必声明变量的数据类型，直接输入“Dim 变量名”此时定义的变量将被指定为 Variant 类型。

但我们在程序设计的过程中，一般应该明确数据的类型，这是一个好的编程习惯，因为指定数据类型后会提高程序的运行速度。

我们可以在模块中输入“Option Explict”作为第一句

语句来强制声明所有变量。也可以点“工具——选项”,在选项对话框里勾选“要求声明变量”。这样，VBA 在遇到没有声明的变量名称，该语句将导致程序停止。

定义变量除了可以使用 Dim 语句外，比较常的还有: static 语句，Private 语句，Public 语句。使用不同的语句定义的变量不同的是它们的作用作用域不同，具体为:

（1）如果在一个过程中包含了一个 Dim 或 Static 语句，此时声明的变量作用域为此过程，即本地变量。

（2）如果在一个模块的第一个过程之前包含了 Dim 或 Prvate 语句，此时声明的变量作用域为此模块里所有的过程，也就是在此模块里所有的过程都可以使用它，即模块作用域下的变量;

（3）如果在一个模块的第一个过程之前包含了 Public 语名，此时声明的变量作用域为所有模块，即公有变量。

变量的作用域是指变量保留其值的这段时间，也称为变量的生成周期，它决定变量可以用于哪个模块或过程中。

给变量赋值

用等号（ = ）是 VBA 里的赋值运算符。

比如我们把"我在学习 VBA 变量！"这个字符串赋给变量

A,则直接输入:

```
A="我在学习 VBA 变量！"
```

对了，你可以在立即窗口里试一下逐条输入:

```
A="我在学习 VBA 变量！"
Mgbox A
```

看一下效果。

注意，在 VBA 里文本是要用英语引号引起来的，还有日期要用#号，比如将日期 2009-3-10 赋给变量 A，则为:

```
A=#2009-3-10#
```

3、常量

执行过程时，变量的值会发生改变，所以我们称它为“变量”，但有时候我们需要引用不会发生变化的数据，这就是常量。常量声明后，不以对它进行再赋值。

声明常量设定常量的值，需要用 Const 语句:

```
Const 常量名 As 数据类型 = 常量的值
```

4、数组

除了变量和常量以外，数组也是我们常用到的。

什么是数组?

数组是具有相同数据类型并且共享一个名字的一组变量的集合。数组也是变量。

数组中的元素通过索引数字加以区分。

如何声明数组?

可以用 Dim 语句或者 Public 语句声明数组。

Dim/Public 数组名 (a to b) as 数据类型

a 和 b 是数组中元素的个数，数组元素素最小值为 a，最大值为 b，元素个数为(b-a+1)个，当然，你也可以直接输入个数，此时默认最小索引号为 0。

如: Dim myarr(5) As Integer

这个数组的最小元素索引号为 0，元素个数为 6。

上面定义的只是一维数组，你还可以定义二维、三维、四维…….

如: Dim myarr(1 to 5,1 to 10) As Integer

这就是一个二维数组，元素个数为 5*10＝50 个。

在程序设计的过程中，定义数组的时候我们可能不能确定数组的元素个数，对数组进行声明后，可以在运行时用 ReDim 语句重新指定数组的大小。

今天的这一部分内容实在很枯燥，连叶枫都有点耐不住了，拉拉杂杂说了这么多，也不知道你看出点门道没？同前一贴讲的内容一样，这些基础都是非常重要的，多读几遍，扎好马才能修练更高深的武功，千万别嫌这个过程的漫长。

一个让你心仪的女孩子，想牵上她的小手，徜徉在公园里的林荫小道上，先前的送花、打电话等等那一串串巴结的行动你会不会嫌太漫长？

VBA 不像函数那样直面我们，它躲在 Excel 的背后，像一个“养在深闺未得见”的美丽姑娘，她飘在墙角的那缕模糊却美丽的身影对我们充满了诱惑，现在我们正在想法设法寻找她的联系电话，要想成功俘虏她，这一步能少吗？

看的没有动的快，动手操作才是学习的正确方法。发现问题，解决问题，这岂不是一个快乐的过程。

有人问:如何知道该定义变量为何种类型？

==========

这就要根据你的实际需要，再结合各种数据的的范围来看。

在编写程序的过程中，选择需要存储空间尽量小的数据类型来保存所需要的数据，这是很有必要的，虽然我们可以把变量设为 Variant 型，但上面我说了，定义变量类型会提升程序的运行速度，这也是 VBA 提供各种数据类型存储空间的原因。

该定义变量为哪一种类型?看一看它的长度。

用生活中的一个实例来说这个问题:

有一天你上街去买一斤菜油，我们需要选择一个容器来装我们所要买的菜油（相当于声明变量的类型），如果我们没有选择选择容器，上帝可不知道你要买多少斤菜油，他就扔给你一个超级大的水缸，这个水缸无论你要买多少菜油都是可以装得下的。可是携带这样一个水缸上街你会不会觉得太不方便了？用一个水缸来装一斤菜油会不会太浪费了?所以我们得自己选择一个瓶子去装我们的油，一斤油，用一个瓶子刚好合适，当然，如果你只用一个茶杯去装，那也是不行的。

用数据来打比方，我们需要用变量来保存一个学生某学科考试成绩的分数，那么用 Byte 数据类型就可以了（它的范围是 0——255），如果在这里使用 Integer、Long、Single 这些数据类型的话，就相当于我们用水桶，水缸去装那一斤菜油，只是对计算机存储空间的浪费，当然，如果你的值的范围超过了 Byte 的范围，那定义成 Byte 也是不行的。

定义变量的类型时，除了考虑长度外，还要考虑变量的值的类型。

买油的时候不管是用瓶子还是用水缸，反正都能装回来，但是如果你用麻袋能不能装那一斤油回来？

还是那句话，如果你没有定义变量类型或定义变量类型为 Variant，聪明的计算机会先看你打的油是固体还是液体，是液体的话把麻袋扔在一边，给你弄口大缸子。但是如果你自己选择麻袋上街去买油，你还能不能把你的油带回来？

用数据来打比方，如果你定义了一个变量为 Integer 型，那你再用把学生姓名的字符串赋给它，行吗?

复制内容到剪贴板

代码:

```
Sub mysub()
    Dim a As Integer
    a = "我在学习 VBA 变量"
    MsgBox a
End Sub
```

将这个过程提制到模块里,按 F5 运行它,看看是什么结果?尝试修改一下定义的数据类型,看看修改后的运行结果!

VBA 函数

=========

用 Excel 的时间相信你也不短了，什么是函数就不再细说了。

与其他编程语言一样,VBA 含有各种内置的函数。很多的 VBA 函数都与 Excel 的工作表函数类似，或者一样。

使用 VBA 函数的的方式与使用工作表公式中函数的方式相同，在 VBA 代码中，我们可以使用很多 Excel 的工作表函数，例如 Sum,Abs 等。

例如:

复制内容到剪贴板

代码:

```
Sub myabs()
    a = InputBox("请输入数值: ", "提示")
    labs = Abs(a)
    MsgBox "你输入的值的绝对值为:" & labs
End Sub
```

这是一个求绝对值的过程，通过 InputBox 提示用户输入一个数值给变量 a，再用 Abs 函数求出变量的绝对值，最后通过MsgBox返回结果。

InputBox 同 MsgBox 函数一样，是很常用的函数，关于它的功能及用法，请用绝招：抹黑，按 F1,

但并不是所有的工作表函数在 VBA 里都可以直接调用，但由于 WorksheetFunction 对象包含在 Application 对象中，如果我们要在 VBA 语句里使用某个不能直接调用的工作表函数，只需要在函数名称前加上 Application.WorksheetFunction 语语即可。

如：在 VBA 里使用 counta 函数则代码为：

```
application.worksheetfunction.counta(range("a1:a10"))
```

VBA 里面的函数很多，可以简化我们的计算和操作，功能强大啊。但当你面对那一堆英文字母，是否为该怎么记住它们感到晕菜？有人对我说，记住常用的几个，其他的不用记那么清楚，有个模糊的印向，用时再翻 Help，知道在哪里找就万事 OK 了。我觉得这个方法很好，现转送给你!

控制程序的流程，有判断、分支和循环三种语句。

接下来我们将一样一样地进行讲解。

＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝

程序流程控制—>if 语句

＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝

1、if—then 语句

这个句式相当于中文里面的“如果.....那么”。

格式:

if 逻辑表达式 then

语句块

end if

解释: 如果逻辑表达式的结果为 true，则执行语句块里的所有语句，否则执行 end if 后面的语句。语句块可以是一句指令或多条指令，也可以调用一个或多个过程，当然，也可以为空，就是没有语句。

如: 要判断 A1 单元格是否已输入内容，设计程序:

复制内容到剪贴板

代码:

```
Sub test1()
 If [a1] = "" Then
    MsgBox "A1 单元格没有输入任何内容！"
 End If
```

End Sub

2、 if—then—else 语句

这个语句相当于中文里面的“如果.....那么.....否则.....”句式。

格式:

```
if 逻辑表达式 then
        语句块 1
else
        语句块 2
 end if
```

解释: 如果逻辑表达式的结果为 true, 则执行语句块 1 里的所有语句, 如果逻辑表达式的结果为 Flase 则执行语句块 2 里的所有语句。

如: 要判断 A1 单元格是否已输入内容, 设计程序:

复制内容到剪贴板

代码:

```
Sub test2()
 If [a1] = "" Then
     MsgBox "A1 单元格没有输入任何内容! "
 Else
     MsgBox "A1 单元格已经输入了内容! "
 End If
```

End Sub

3、if—then—elseif 语句

这就相当于中文里面的若“如果…那么…否则如果…那么…否则”语句，当然，可以有多句“否则如果.....”

格式:

```
If 逻辑表达式 1 Then
        语句块 1
ElseIf 逻辑表达式 2 Then
        语句块 2
ElseIf 逻辑表达式 3Then
        语句块 3
  …
Else
        语句块 n
Eed If
```

解释: 若逻辑表达式 1 的结果为 True，则执行语句块 1，接着继续执行 EndIf 后面的语句;若逻辑表达式 1 的结果为 False，则检查逻辑表达式 2 的结果，若为 True 则执行语句块 2，接着继续执行 EndIf 后面的语句.....一直到所有结果的条件都不满足时，才执行 Else 后面的语句块 n。

如: 要判断 A1 单元格的数是否能被 2、3、5 其中之一整除，设计程序:

复制内容到剪贴板

代码:

```
Sub test3()
  If [a1] = "" Then
     MsgBox "A1 单元格没有输入任何内容! "
  ElseIf [a1] Mod 2 = 0 Then
     MsgBox "A1 单元格的数能被 2 整除! "
  ElseIf [a1] Mod 3 = 0 Then
     MsgBox "A1 单元格的数能被 3 整除! "
  ElseIf [a1] Mod 5 = 0 Then
     MsgBox "A1 单元格的数能被 5 整除! "
  Else
     MsgBox "A1 单元格的数不能被 2、3、5 其中之一整除! "
  End If
End Sub
```

IF 语句.rar (6.7 KB)

程序流程控制——Select Case 语句

==========

Select Case 语句同 IF 语句一样，也是条件判断的语句。它的功能也可以用 IF 语句来完成。但是，当程序的条件太多，用 IF 语句来判断的话就存在一些不足了，比如程序是否美观？是否便于阅读？程序的执行效率等等。Select Case 语句在执行

的效率上，同 IF 语句相比，肯定是要快得多。

[Select Case 语句的语法]

```
    Select Case 测试表达式
        Case 条件表达式 1
            语句块 1
        Case 条件表达式 2
            语句块 2
       Case 条件表达式 3
            语句块 3
        ......
       Case 条件表达式 n
            语句块 n
        Case Else
            语句块 Else
   End Select.
```

当某个条件表达式与测试表达式相匹配时，则执行其后的语句块，否则执行 Case Else 后的语句块，然后结束 Select Case 块的执行。同 IF 语句一样，可以不要 Case Else 语句。

“如对 A1 单元格的学生成绩进行等级评定，小于等于 30 分的为差，30 分到 59 分的为不及格，60 分到 79 分的为及格，80 分到 89 分的为良好，90 到 100 分的为优秀。”用 Select Case 语句来解决就是:

复制内容到剪贴板

代码:

```
Sub test()
 If [a1].Value = "" Then
    MsgBox "A1 单元格没有输入数字。"
    Exit Sub ’   退出程序
 End If
 Select Case [a1].Value
    Case 0 To 29
       MsgBox "差"
    Case 30 To 59
       MsgBox "不及格"
    Case 60 To 79
       MsgBox "及格"
    Case 80 To 89
       MsgBox "良好"
    Case Else
       MsgBox "优秀"
 End Select
End Sub
```

Select Case 语句.rar (6.44 KB)

Select Case 语句.rar (6.44 KB)

需要补充一点的是，无论是IF语句还是Select Case语句，都是可以进行嵌套的!

引用:

ggsmart的补充说明:楼上写的代码,我只考虑分数是整数的时候，这里我只是举个例子，目的是让大家认识Select Case语句，因为没有考虑小数和负数的情况.一旦出现29.5之类的,就都全是优秀.所以，程序本身并不全面.为此,38度OFFICE技术论坛的老朽对代码进行了修定,提供了另一段全面的代码,非常感谢!

代码:

```
'修订:38度:老朽
'网址:http://www.38duoffice.cn/bbs
'日期:2009-7-1 上午 11:22:21
Sub Test()
 If [a1].Value = "" Then
    MsgBox "A1单元格没有输入数字。"
    Exit Sub '   退出程序
 End If
 Select Case [a1].Value
    Case Is < 30
       MsgBox "差"
    Case Is < 60
```

```
        MsgBox "不及格"
    Case Is < 80
        MsgBox "及格"
    Case Is < 90
        MsgBox "良好"
    Case Else
        MsgBox "优秀"
 End Select
End Sub
```

程序流程控制——For—Next 循环语句

==========

什么是循环?

这里我们指的循环是指重复地执行某项动作(语句块)。同学们在操场上跑步,两万米长跑,每圈 400 米。“呼”,枪一响,开始跑,一圈,两圈,三圈.....直到跑满 50 圈才停止。这里,同学们就是在循环地在操场的跑道上执行跑步的动作。

我们来看一下 For—next 的句式:

```
 For 循环变量=初值 to 终值 step 步长
     循环体 1
    [exit for]
```

循环体 2

next 循环变量

解释:从开始到结束，反复执行 For 和 Next 之间的指令块，除非遇到 Exit For 语句，将提前跳出循环。其中，步长和 Exit For 语句以及 Next 后的循环变量均可省略，步长省略的时候默认为

1。Exit for 语句是强制终止循环的语句，执行它后将退出循环，执行 next 后面的语句。

循环变量是一个变量，可以在循环体中对其进行修改，但一般就避免在循环体中对其修改。

对句式进行了初步的认识后，我们用这个句式来叙述上面学生跑步的问题。把它编成程序，20000 米，要跑 50 圈，也就是要 50 次，同学们要在上面一圈，两圈。。。。直到满 50 圈才停止循环跑步，当然，如果你要提前退赛，就不用跑满 50 圈了。有了这个思想，程序可以写成:

```
Sub 循环跑步()
   dim 圈数 as byte
   for 圈数=1 to 50 step 1
      If 学生.要求=退赛
         exit for
      End If
   Next 圈数
```

End Sub

当然，这只是一个实例，帮助你对 For—next 语句的理解，下面我们来用一个实例来讲解 For-next 语句！

问题：求 1 到 1000 的自然数的和。程序编为：

复制内容到剪贴板

代码：

```
Sub mysum()
Dim Lsum As Long, i As Long
For i = 1 To 1000
   Lsum = Lsum + i
Next
MsgBox "1 到 1000 的自然数和为： " & Lsum
End Sub
```

for_next-1 到 1000 自然数的和.rar (6.14 KB)

以下代码由老朽提供

引用：

ggsmart 说明：以下代码为 38 度 OFFICE 技术论坛的老朽版主提供，主要是帮助大家对 VBA 语句的认识.

复制内容到剪贴板

代码：

```
撰写：38 度：老朽
    '日期：2009-7-1 上午 11:40:59
```

```
Sub 求1到10000之间偶数和()
    Dim I&, J&
    For I = 0 To 10000 Step 2
      J = J + I
    Next
    MsgBox "1到10000之间偶数和为" & J
End Sub
```

复制内容到剪贴板

代码:

```
    '撰写:38度:老朽
    '日期:2009-7-1 上午 11:40:59
Sub 求1到10000之间偶数和2()
    Dim I&, J&
    For I = 10000 To 1 Step -2
      J = J + I
    Next
    MsgBox "1到10000之间偶数和为" & J
End Sub
```

复制内容到剪贴板

代码:

```
    '撰写:38度:老朽
    '日期:2009-7-1 上午 11:40:59
```

```
Sub 求1到10000之间奇数和()
    Dim I&, J&
    For I = 1 To 10000 Step 2
      J = J + I
    Next
    MsgBox "1到10000之间奇数和为" & J
End Sub
```

复制内容到剪贴板

代码:

```
    '撰写:38度:老朽
    '日期:2009-7-1 上午 11:40:59
Sub 求1到10000之间奇数和2()
    Dim I&, J&
    For I = 9999To 1 Step -2'此句于2009年8月18日纠正
      J = J + I
    Next
    MsgBox "1到10000之间奇数和为" & J
End Sub
```

引用:

感谢109楼artemis_xu朋友对以上代码的纠正!

复制内容到剪贴板

代码:

```
    '撰写:38 度:老朽
    '日期:2009-7-1 上午 11:40:59
Sub 求 1 到 10000 之间能被 5 整除的数之和()
   Dim I&, J&
   For I = 0 To 10000 Step 5
     J = J + I
   Next
   MsgBox "1 到 10000 之间能被 5 整除的数之和为" & J
End Sub
```

复制内容到剪贴板

代码:

```
    '撰写:38 度:老朽
    '日期:2009-7-1 上午 11:40:59
Sub 求 1 到 10000 之间能被 5 整除的数之和 2()
   Dim I&, J&
   For I = 10000 To 1 Step -5
     J = J + I
   Next
   MsgBox "1 到 10000 之间能被 5 整除的数之和为" & J
End Sub
```

程序流程控制——For—Each 循环语句

==========

有了前面对循环一词的理解，这里我们不用再多说，如果你知道了 For—next 语句的工作原理，这里再来学习 For—each 循环以及下面我们接着要讲的句式，那就相当简单了，同样的工作原理。

学会了吃苹果，那么离会吃梨的道路还会远吗？

呵呵，只是开个小小的玩笑，下面我们继续，首先来看看 For—Each 语句的句式：

For Eacn 元素变量 In 对象集合或数组名称

语句块 1

[Exit For]

语句块 2

next 元素变量

和上面 For—next 句式是一样的工作原理。

这里的“元素变量”是用来遍历集合或数中中元素的变量，它从集合或数组的第一个元素开始，直到最后一个元素，然后退出循环。

这里我们举个例子，把当前工作薄中工作表的名称写在 A 列，程序为：

复制内容到剪贴板

代码：

```
Sub shtname()
Dim i As Integer, sht As Worksheet
i = 1
For Each sht In Worksheets
   Cells(i, 1) = sht.Name
   i = i + 1 '让写入名称的单元格下移一行
Next
End Sub
```

for-each 返回工作表名称.rar (7.52 KB)

for-each 返回工作表名称.rar (7.52 KB)

程序流程控制——Do While 语句

==========

Do While 也是循环语句，它分为两种情况，一种是把循环条件放在开头，一种是把循环条件放在结尾。

1、开头判断循环条件

语句格式:

```
Do Wihle 循环条件
    语句块 1
   [Exit Do]
    语句块 2
Loop
```

首先判断循环条件，条件为真则执行 Do 到 Loop 之间的语句。

2、结尾判断循环条件

语句格式:

```
Do
  语句块 1
[Exit Do]
  语句块 2
Loop Wihle 循环条件
```

先执行一次 Do 到 Loop 之间的语句，再判断循环条件，满足条件则进行循环。

两种格式的区别: 因为第二种格式是把循环体放在尾部，得先执行一遍语句再进行循环条件判断，所以，同样的条件，第二种格式的循环会比第一种多执行一次循环部份的语句。

我们用 Do While 语句来解决求 1——1000 的自然数的和的问题:

复制内容到剪贴板

代码:

```
Sub mysum()
Dim Lsum As Long, i As Long
i = 1
Do While i <= 1000
  Lsum = Lsum + i
```

```
  i = i + 1
Loop
MsgBox "1 到 1000 的自然数和为: " & Lsum
End Sub
```

do while1 到 1000 自然数的和.rar (6.19 KB)

do while1 到 1000 自然数的和.rar (6.19 KB)

程序流程控制——Do Until 语句

这个和 Do While 语句类似，它也有两种形式，学了 Do While 语句，再看这个，就可以无师自通了。下面就简单介绍下:

1、开头判断循环条件

语句格式:

```
Do Until 循环条件
    语句块 1
   [Exit Do]
    语句块 2
Loop
```

2、结尾判断循环条件

语句格式:

```
Do
    语句块 1
```

[Exit Do]

语句块 2

Loop Until 循环条件

执行 Do 和 Loop 之间的指令，直到循环条件为真时退出循环。

直到循环条件为真时退出循环，这是 Do Until 与 Do While 的区别。而它两种语句格式的区别，也是第二种比第一种多执行一次循环部分的语句。

还是求 1——1000 自然数的问题,这次我们换用 Do Untile 语句:

代码:

```
Sub mysum()
    Dim Lsum As Long, i As Long
    i = 1
    Do
      Lsum = Lsum + i
      i = i + 1
    Loop Until i > 1000
    MsgBox "1 到 1000 的自然数和为: " & Lsum
End Sub
```

do until1 到 1000 自然数的和.rar (6.19 KB)

do until1 到 1000 自然数的和.rar (6.19 KB)

是的，你看了上面的代码，觉得和楼上我们写的代码变化并不大。这里我用的是第二种格式(结尾判断)的语句，你可以试着用第一种句式写一写,或者改一改其他的问题.这里,我只是举个例子，路还得让你自己走!

程序流程控制——Go to 语句

==========

Go to 在英语里是什么？相信大家的英语都比我好得太多，不用多说。其实 Go to 语句的作用用定两个英语单词就说明问题了，“Go to 地点”就是去到哪儿的意思。

Go to 语句是将程序转到指定的标签的语句位置，然后继续往下执行。Go to 语句通常用来作错误处理。

同样的，还是用 Go to 语句来做 1——1000 自然数的和:

代码:

```
Sub mysum()
Dim Lsum As Long, i As Long
i = 1
x: ′为 go to 语句设置的标签，必须以英文状态下的冒号结尾
  Lsum = Lsum + i
  i = i + 1
If i <= 1000 Then GoTo x ′如果 i<=1000,则程序跳到标签 X 处
MsgBox "1 到 1000 的自然数和为: " & Lsum
```

End Sub

go to 1到1000自然数的和.rar (6.34 KB)

go to 1到1000自然数的和.rar (6.34 KB)

注意：为Go to 语句设置的标签必须以英文状态的冒号结尾。有人说，为了方便程序的阅读，规范程序，在程序里也避免少用Go to 语句，我很赞同，也希望你们能养成这个习惯。

用户窗体

用户窗体(UserForm 对象)是 VBA 的另一类对象。它表现出来是一个窗口或对话框，如前面我们说的MsgBox 与 InputBox 语句弹出来的都属于窗体。使用窗体可以提供一个形象的用户界面。同我们可以在工作表（WorkSheet）上在添加不同的控件一样，在这个界面上，我们可以添加命令按钮、标签、图像、列表框等控件，用户可以通过修改控件的属性、编写不同的事件过程在程序运行过程中与程序进行交互。

1、添加窗体

如果你还没有忘记怎么插入模块？那请用同样的方法插入一个用户窗体。

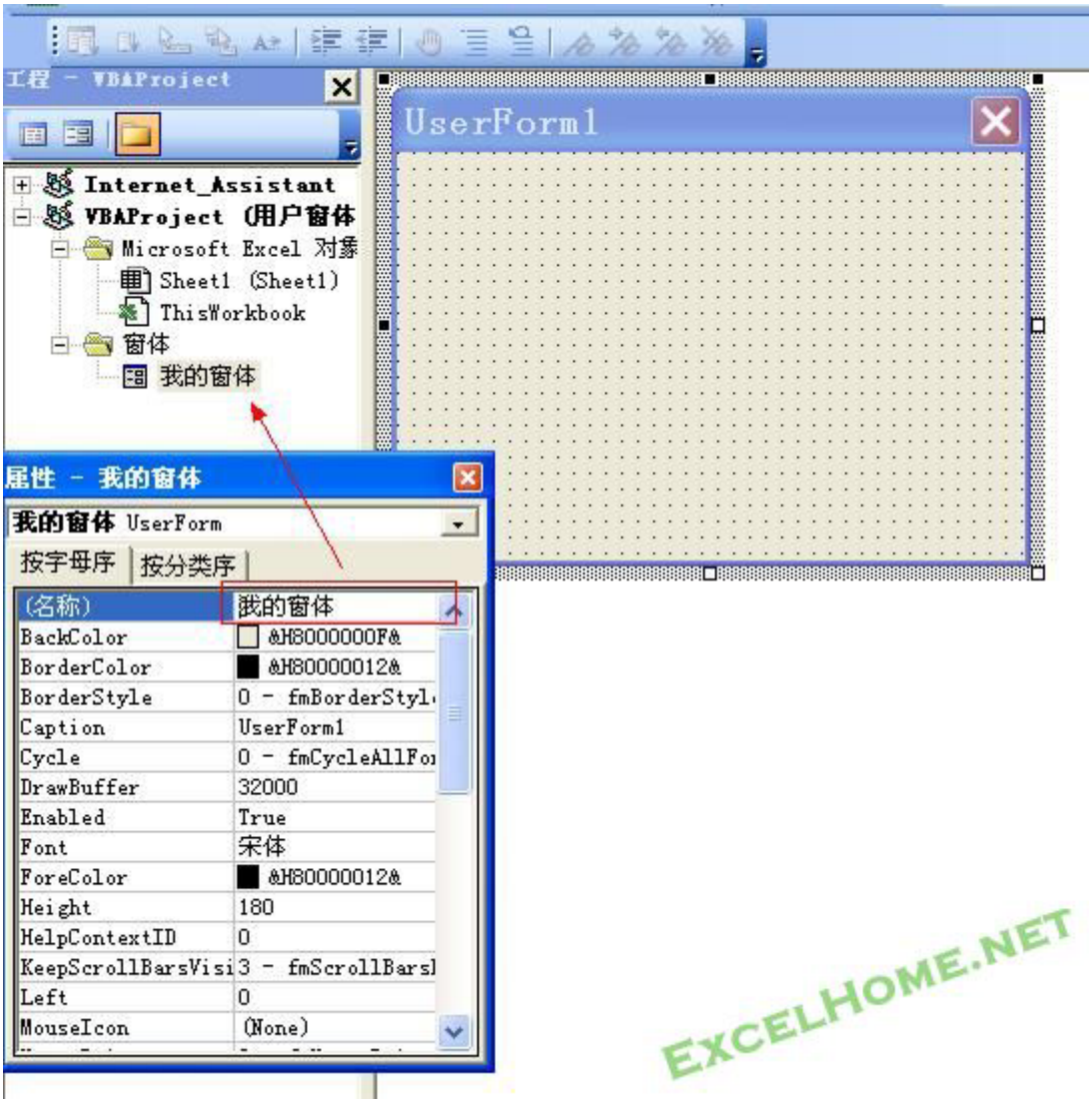
Alt+F11 快捷键打开 VBE 编辑窗口，在工程资源管理器窗口中单击右键，选择“插入——用户窗体”命令。



1-1.jpg (33.67 KB)

Excel 为用户窗体取名，同为工作表、按钮等取名一样，采用“老大，老二，老三……”的取名方式。所以新插入的用户窗体默认名称为“UserForm1”，如果再继续插入窗体，就是“UserForm2”，接着是“UserForm3”……

当然，你可以为它重新取一个你喜欢的名字，新名字叫大花二花，或者是大狗二狗，随你了，为了方便学习，这里我们统一为其更名为“我的窗体”，方法为：在工程资源管理器中双击窗体对象，在属性窗口中重新修改“(名称)”(对象的 name 属性)为“我的窗体”即可。修改完成后，可以在工程资源管理器窗口中看到相应的名称已经改变了。



1-2.jpg (81.78 KB)

这时，窗体的“名称”已经修改好了，再看看窗体，左上角还显示着“UserForm1”，这就相当于贴在一个按钮表面的标签，想要修改它，只需要修改对象的“Caption”属性即可。用改“名称”的方法，我们把这个窗体的“Caption”属性修改为“我的第一个 VBA 窗体”。

就像你家里贴的地板一样，新建的窗体的大小样子都一样。但是我们根据自己的需要对其进行美化或修改。如大小，背景颜色，背景图片等等，我们可以手动在属性窗口里对其进行修改，

也可以程序中用代码进行修改。如修改该窗体的宽度，代码设置为:

代码:

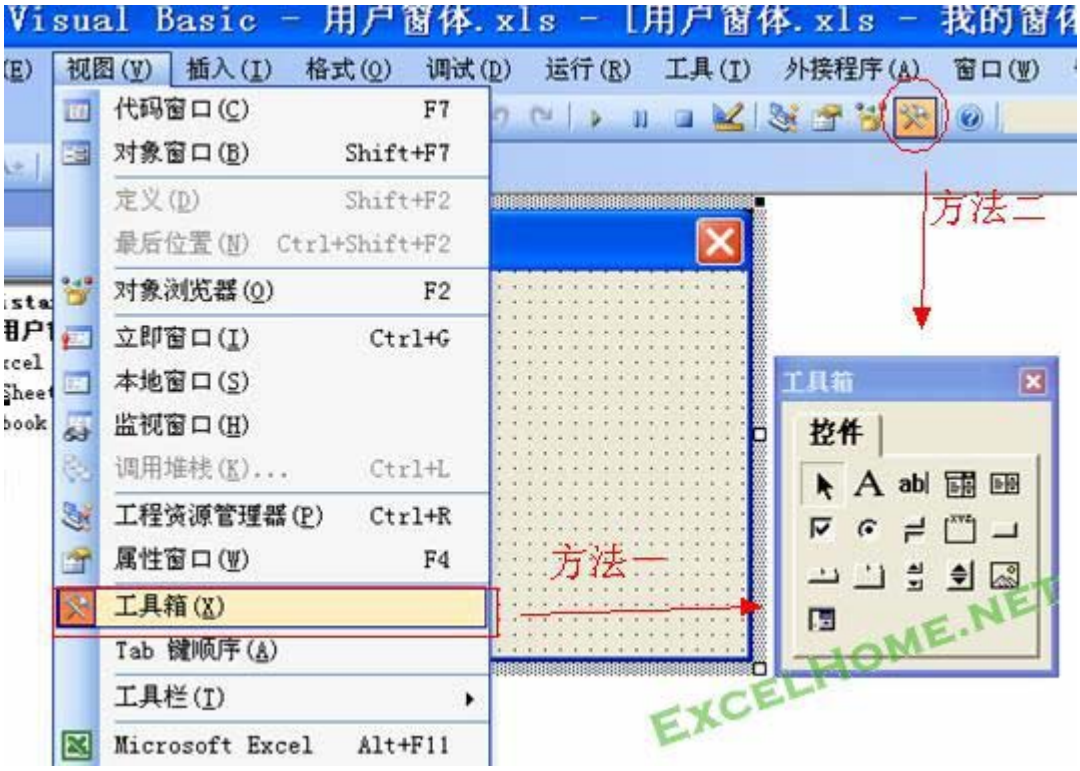
我的窗体.Width = 500　　′500是你需要修改后的宽度，以磅（Point）为单位

动手试一试，在属性窗口里设置“我的窗体”宽300 Point（Width属性），高200 Point(Height属性)。

2、为窗体添加界面元素

窗体就像一个舞台，舞台上得有演员才能和观众进行互动。同样，我们得为窗体这个舞台添加演员，演员可以是命令按钮、标签、文字框等等，你可以根据自己的剧本的需要安排不同的角色。

向窗体添加控件，同在工作表中添加窗体控件一样。如果你现在的VBE编辑窗口中没有“工具箱”，请通过“视图——>工具箱”菜单命令，或单击“工具栏”上的“工具箱”按钮打开它。



1-3.jpg (55.44 KB)

想知道“工具箱”里的控件是哪一种是“命令按钮”，哪一种是“标签”，哪一种是“文字框”，只需要把鼠标指针移到相应的控件上面，即会显示相应的说明。

在工具箱里用鼠标点选相应的控件，在窗体上按下鼠标左键，拖动即可将控件添加到用户窗体。控件的大小、在窗体上的位置我们可以通过鼠标进行修改，当然，同修改其他属性一样，我们也可以在属性窗口或程序的过程中用代码进行修改。

别闲着，动手试一试：

（1）向窗体添加一个命令按钮

改按钮名称为“确定”。设置显示在按钮上的文字（Caption 属性）为“完成”，高 25（Height 属性），宽 60（Width 属性），距“我的窗体”左端 220（Left 属性），距“我的窗体”顶端 80

（Top 属性）。

引用:

当然，上面只是教你怎样在属性窗口里对控件的属性进行修改。要求不是太完美的时候，我们可以在窗体上用鼠标拖动修改一下就可以了。

（2）向窗体添加一个标签

改标签的名称为“说明”。设置字体（Font 属性）为“宋体，五号，斜体”，显示的内容（Caption 属性）为“请输入内容”（慢速双击激活控件即可输入）。

（3）向窗体添加一个文字框

改文字框的名称为“内容”，设置字体（Font 属性）为“楷体五号字”，字体颜色（ForeColor 属性）为“红色”。

回到窗体设计模式，按 F5 键（或者执行“运行——>运行子过程/窗体”菜单命令，或单击工具栏上的“运行子过程/窗体”命令按钮），完整的一个用户窗体就显示在桌面上了。



1-4.jpg (23.78 KB)

当然，因为没有添加任何的 VBA 指令，此时，窗体里的控件是不会工作的。

（4）为命令按钮添加事件过程

上面的标签，文字框，命令按等控件（ActiveX 控件，同我们刚开始说的窗体控件是有区别的）都属于对象，我们可以通过编写相应的代码设置其属性，调用其方法，响应其事件来达到我们需要实现的操作。

比如在“我的窗体”中，如果需要在点击“完成”按钮（按钮名称为“确定”）后，用一个 Msgbox 告诉我们，文字框（名称为“内容”）里输入的是什么内容。

有了这个思想，我们可以模拟把程序写为：

```
Sub 确定_单击()
```

Msgbox “文字框里输入的内容是:” & 内容.输入的内容

End Sub

有了这个模拟的程序，再来写就简单了。

“单击”是“确定”的一个事件，即 Click 事件，“内容.输入的内容”是读取“内容”的 Value 属性。

这样，我们的程序实际上是:

代码:

```
Sub 确定_Click()
    MsgBox "文字框里输入的内容是: " & 内容.Value
End Sub
```

回到窗体，双击按钮，切换到代码模式，在左面“对象列表”中选择对象“确定”，右面事件列表中选择事件“Click”，将 Sub 与 End Sub 之间输入代码 MsgBox "文字框里输入的内容是: " & 内容.Value。回到设计模式下，按 F5 键，再次运行窗体，在文字框里输入内容，单击“完成”。

也许你需要在文字框里什么也没有输时，单击按钮后，提示我们“文字框里没有输入任何内容”，并退出程序。可以的，加一个 If—Then 语句即可。如果需要将文字框里的内容换行显示，那么在提示内容与显示内容之间再连接一个“Chr(13)”，就个的作用相当工作表中的“Char(10)”，这里不再多说。

我最后写的代码是:

代码:

```
Private Sub 确定_Click()
    If 内容.Value = "" Then
      MsgBox "文本框里没有输入任何内容!"
      Exit Sub    '退出 sub 过程，不再执行下面的语句
    End If
    MsgBox "文字框里输入的内容是: " & Chr(13) & 内容.Value
End Sub
```

3、显示窗体

显示窗体就是把窗体显示给用户。

显示窗体可以调用窗体的 Show 方法。

窗体的显示模式有两种:

（1）模式窗体: 窗体显示后将停显示之后的代码直到退出或隐藏此窗体，并且必须退出或隐藏此窗体后，才可以操作非此窗体的其他界面元素。

也就是说，当你使用模式窗体显示某窗体的时候，当程序执行完“显示窗体”这条命令后，将暂停执行程序里剩余的语句，真正这个窗体被隐藏或退出。关且当窗体显示出来以后，你不能激活该窗体以外的任何区域。

（2）无模式窗体: 窗体显示后会继续执行程序里余下的语

句，而其他的窗体或界面也可以进行操作。

显示窗体的语句:

（1）模式窗体的显示代码:

窗体名称.Show vbModal

也可以写成: 窗体名称.Show 1

（2）无模式窗体的显示代码:

窗体名称.Show vbModeless

也可以写成: 窗体名称.Show 0

引用:

悄悄告诉你: 因为对于 Show 方法，如果不指定模式，则默认为模式窗体，又因为我很懒，所以我的习惯是: 当我需要用模式窗体的时候，就直接“窗体.Show”，需要无模式的时候就“窗体.Show 0”，多个 0 少个 0 而已，懒得去记那有模式无模式的字母，我英文差，反正看了也不一定认识。

4、关闭窗体

当不再需要使用某个窗体后，我们需要把它关闭。当然，每一个窗体上都有“关闭”按钮。但这肯定不能满足我们在程序的设计过程中的需求。

我们可以通过 Hide 来隐藏窗体。代码为:

窗体名称.Hide

也可以通过 Unload 来卸载窗体。代码为:

Unload 窗体名称

二者都可以关闭窗体，但是却是有区别的。Show 一个 Hide 的窗体比 Show 一个 Unload 的窗体用的时间要短。所以，当你需要反复使用某个窗体的时候，建议你用 Hide 而不用 Unload。

上面讲的这些内容，打开 VBE 窗口，插入窗体，添加控件，修改属性，编写事件过程等等，这些，你都跟着做了吗？如果你已经做了，那最后我再留给你一点事情算是作业吧:

（1）插入一个模块，编写两个 Sub 过程，分别以“模式窗口”和“无模式窗口”显示“我的窗体”（可在显示窗体命令的下一行再编写另外一条指令查看两者在程序执行过程中的区别）;

（2）在工作表上插入两个窗体控件的按钮（记住是窗体控件），分别改按钮上显示的字为“模式窗体”和“无模式窗体”，并把刚才编写的两个宏分别指定给它们;

（3）再次进行入“我的窗体”的设计模式，添加两个按钮，分别改其“Caption”属性为“隐藏窗体”和“卸载窗体”。分别为两个按钮编写响应其相应事件的代码，使得当单击“隐藏窗体”按钮时，隐然“我的窗体”，单击“卸载窗体”时，卸载“我的窗体”。

弄好后，你可以点击工作表上的铵钮，调出窗体，看看自己的第一件作品了。

为避免大家都不习惯，我还是不客气好了。。。。。11 楼的代码 为什么前面要 if 呢？为什么不全部 case?

```
    Private Sub Worksheet_SelectionChange(ByVal Target As Range)
Select Case [a1].Value
   Case ""
   MsgBox "未输入任何字符"
   Case Is < 60
   MsgBox "不及格"
   Case Is < 85
   MsgBox "较好"
   Case Is < 100.0001
   MsgBox "优秀"
   Case Else
   MsgBox "不能识别"
End Select
End Sub
```

引用:

> 为避免大家都不习惯，我还是不客气好了。。。。。11 楼的代码 为什么前面要 if 呢？为什么不全部 case?

```
Private Sub Worksheet_SelectionChange(ByVal Target As
```

Range)

```
Select Case [a1].Value
   Case ""
   Ms ...
```

=====

你的这个更规范了，呵呵。。。。赞一个!

我也不知道当时我为什么不全用 Case。。。。汗。。。。

再说 Excel 的对象模型

在前面的贴子里，我们简单介绍了 VBA 的开发环境、语法基础及常用的语句。同时对集合、对象、属性、方法和事件等几个重要的概念作了简单的讲解。通过学习，你是否有所收获？我说了，我只是给你指路。就像你说你要到北京天安门看升国旗仪式，我只能告诉你打出租车到机场，花一千块买张机票你就可以去了，我可不能把你背去北京我再要着饭走回来。所以现在，对前面学过的内容，你可以在心里面作一个回顾，问问自己，通过学习，收获了什么？还存在什么不足？接下来该怎么做？

适时的小结是很有必要的，可以看到存在的不足和努力的方向。

所以这一贴，没有新内容，再来说一说 Excel 的对象模型，当作是一种对前面的回顾也罢，当作是和大家的一种闲聊也行。不管了!

VBA是通过对象操作来操作和控制Excel。不管是操作Excel程序（Application 对象）、工作薄（Workbook 对象）、工作表（Worksheet 对象）、单元格（Range 对象）还是操作图表（Chart 对象）、窗体（Userfrom 对象）等等，都是在操作对象。每个对象都是其他对象的一部分，也可以由其他对象组成。这个前面的贴子已经讲过，这里不再多说。

我要说的是，VBA 的语法和 Excel 的对象模型组成了应用 Excel VBA 程序设计的基础，从某种意义上来说，理解和熟悉 Excel 对象模型的过程，也就是使用 Excel VBA 进行程序设计的过程。所以，熟悉 Excel 的对象模型，对我们学习 VBA 的重要性可想而知。

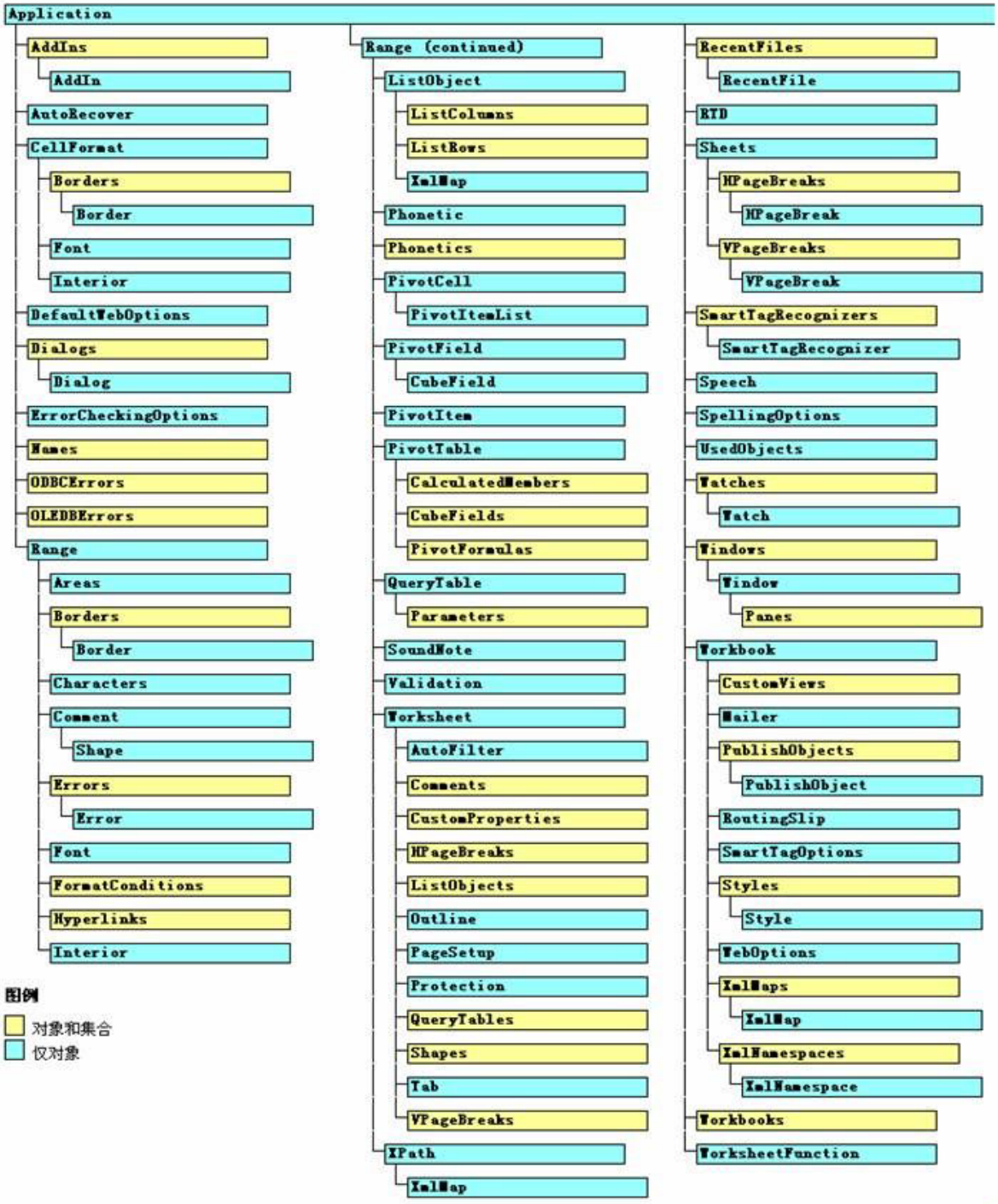
可是，什么是 Excel 的对象模型？

不知道这样讲你是否能理解：家里面有厨房吧？厨房里面有很多的东西，比如碗柜，冰箱等，而碗柜里又有饭碗，盘子等，冰箱里有鸡蛋，黄瓜等。Excel 就像一个装了很多东西的大厨房，Workbook 对象就像冰箱，Worksheet 就像冰箱里的蔬菜啊什么的。当我们给这个大厨房及厨房里的这些对象画一个简单的结构图时，这个结构图我们就称之为对象模型。

下面就是 Excel 的对象模型结构图：



Microsoft Excel 对象模型



请参阅以下链接，了解有关主要 Excel 对象的详细信息

ChartObject 对象

Name 对象

Shapes 集合

SmartTag 对象

EXCELHOME.NET

2-1.jpg (150.82 KB)

Excel 对象模型在 Excel 的帮助文件里就有了，我们可以通过打开帮助来查看它。如果你找不到，请打开你的 OFFICE 的安装目录，找到\OFFICE11\2052 目录下的 VBAXL10.CHM。这个文件就是 Excel VBA 的 Help 文件了，双击打开它，你就可以看到我们上面的图了。

从对象模型，我们可以看到对象之间的包含关系，Excel 最顶端的对象为 Application 对象，即 Excel 程序本身，其他所有的对象都是在 Application 对象的下面。用鼠标单击其中的每一个对象，就可以跳转到对象的说明。

呵呵，知道什么是 VBA 了吧？就是对以上这些对象的操作。就像在厨房里做饭是对厨房里各种对象进行操作一样，在 Excel Vba 程序设计过程中，我们就是通过读取或设置对象的属性、调用对象的方法、编写事件过程来操作 Excel。

对象模型很重要，是的。但是 Excel 对象模型包括了大量的对象、属性和方法，就像书架上放着的汉语大词典，也许有些内容我们一辈子都不会用到它，所以就像我们不用去记住词典的内容一样，我们不必记住对象模型里的所有内容。我们要做的只是熟悉它的结构和组成，需要用到时，翻开字典，查到它所在的页码，然后快速地去打开它。

熟悉了对象模型，当你在学习 VBA 的过程中，如果再有什么问题，有字典可查了吧？千万不要忘记了这个随时跟在我们身边的老师。

不要嫌查字典的麻烦，二年级时你天天翻字典，到高中到大学了你还天天翻吗？从菜鸟到高手，就是一个从查 Help 到不需要查 Help 的过程，刚开始时解决一个问题也许需要几个小时，后来也许会快到只要几分钟。呵呵……这是别人告诉我的，我相信，现在我还在坚持翻字典，但我坚信会有不需要再翻它的那一天！也希望你信！

PS：如果你的 Excel 里没有 Help，找不到对象模型的文件，那我给你一个：

VBAXL10.part1.rar (480 KB)

VBAXL10.part1.rar (480 KB)

VBAXL10.part2.rar (480 KB)

VBAXL10.part2.rar (480 KB)

VBAXL10.part3.rar (480 KB)

VBAXL10.part3.rar (480 KB)

VBAXL10.part4.rar (480 KB)

VBAXL10.part4.rar (480 KB)

VBAXL10.part5.rar (333.47 KB)

[小试牛刀]制作个性化的欢迎界面

如果你辛辛苦苦设计了一个 Excel 应用程序，想不想很臭美的显摆一下？那我们就给它设计一下“欢迎界面”吧，让每一次

别人用它的时候都给他点信息，告诉他，这是爷我做的，顺便也打打广告什么的。

什么是“欢迎界面”？经常作用电脑的我们应该不陌生吧？打开 Excel、Word、PS 等等软件时，在加载的那几秒钟里，都显示一个欢迎界面，给我们介绍了一些基本的信息。

这一贴我们就来给我们的 Excel 做一个个性化的欢迎界面，需要实现的目的是:

（1）打开 Excel 文件时，隐藏 Excel 应用程序的界面，同时弹出一个欢迎界面，界面上有你想向别人显摆的内容，有什么，随你了;

（2）欢迎界面显示 5 秒钟后，退出欢迎界面，进入 Excel 文件;

（3）界面上要动态显示几秒钟后进入工作薄。

例子见附件:

欢迎界面.rar (40.37 KB)

欢迎界面.rar (40.37 KB)

这是我们第一次来做例子，在制作的过程中，也许你需要弄清楚以下一些问题，如果不知道的，去查 Help 了:

1、想一想，需要发生什么事件才能实现在打开 Excel 的时候隐藏程序的界面而显示欢迎界面?

2、要隐藏或显示一个对象，需要修改它的 Visible 属性，

Excel 应用程序是什么对象？还记得吗？

3、显示一个窗体要用什么方法？关闭一个窗体要用怎么做？

4、如果我们用一个标签来显示“几秒种后进入文件”，那么每隔一秒种我们需要修改一次该标签的什么属性？

5、怎样安排一个程序在将来的某个时间运行？这需要用到 OnTime 方法。

如果有一个程序名叫“mysub”，我们需要这个程序在 5 秒钟后运行，则代码为：

复制内容到剪贴板

代码：

```
Application.OnTime Now + TimeValue("00:00:05"), "mysub"
```

详情可以查对象模型，里面有介绍。

6、怎么在程序里调用另一个过程？

语句格式为：Call 程序名

如要在一个程序里运行名为“mysub”的过程，只需要输入：Call mysub 即可。

其实直接输入程序名也可以，但我不喜欢这样做，总感觉不便于阅读！

7、怎样禁用户通过关闭窗体右上角的关闭按钮关闭“欢迎界面”？这需要用到窗体（UserFrome 对象）的 QueryClose 事

件了。请查看 Help。

好了，能告诉你的就这么多了，希望你能做出来！我例子的密码会在下一贴的时候公布！为了能好好显摆一下，希望你的欢迎界面不要像我的这样简单，多给它美化美化，不但可以加深自己对窗体的认识，还可以增加显摆的条件。

希望各位把自己的成果展现出来，供大家学习！

处理单元格数据

VBA 程序设计的最终目的是什么？更多的当然是为了处理数据。所以，这一贴，将来讲一讲用 VBA 处理单元格区域。当然，因为我知道的本身不多，所以，并不能把所有的东西都告诉你，我只是举一些简单的例子，供大家学习作参考用。

建议大家在读贴的过程中，动动手，亲自试一试，相信这样的亲身经历会给你的学习带来一定的帮助。

一、复制单元格区域

如果你感到无从着手，别忘了“录制宏”的功能。虽然录制宏生成的代码不是最有效的，但它生成的代码可以为我们提供很多的信息。

好了，闲话不说。自行录一段复制 A1 单元格到 B1 单元格的宏。打开 VBE 编辑器，得到如下代码:

复制内容到剪贴板

代码:

```
Sub Macro1()
```

```
'
' Macro1 Macro
' 宏由 ggsmart 录制，时间: 2009-8-23
'


'
    Range("A1").Select
    Selection.Copy
    Range("B1").Select
    ActiveSheet.Paste
End Sub
```

除去注释的部分，实际上只有句代码。不用我细讲，大家已经知道它们的作用分别是: 第一句选中 A1 单元格，第二句复制选中的单元格，第三句选中 B1 单元格，第四句粘贴。

但是，就像我们用 VBA 在单元格输入数据不用先选中它一样，我们同样可以在不选中单元格的情况下就执行复制粘贴命令。所以，以上的代码实际上我们可以简化为:

复制内容到剪贴板

代码:

```
Sub Macro1()
    Range("A1").Copy Range("B1")
End Sub
```

以上的代码只是发生在活动工作表内，如果要在其他工作薄或工作薄中实现复制粘贴的命令，只需要改变引用即可。如我们要将 Book1.xls 的第一个工作表中的 A1:A10 复制到 Book2.xls 的第一个工作表中的 B1:B10 单元格，则代码为:

复制内容到剪贴板

代码:

```
Sub mycopy()
  Workbooks("book1").Sheets(1).[a1:a10].Copy _
  Workbooks("book2").Sheets(1).[b1:b10]
End Sub
```

引用:

提示: 一行的代码过长，可以用空格加下划线进行换行。

复制的区域过大时，对于目标区域，我们可以只指定一个单元格，这时它代表目标区域的左上角单元格。如上面的代码，我们可以写成这样:

复制内容到剪贴板

代码:

```
Sub mycopy()
  Workbooks("book1").Sheets(1).[a1:a10].Copy _
  Workbooks("book2").Sheets(1).[b1]
End Sub
```

如果你需要只粘贴数值，或其他什么，你可以利用录制选择性粘贴的宏来进行修改。同样的道理，我就不再多说。

二、剪切单元格

剪切单元格，或者说移动单元格和复制单元格类似，有了上面的经验，我们不难得到它的语句。其实它和复制单元格的命令大同小异，唯一不同的是使用 Cut 方法代替了 Copy 方法。

下面的代码是将 A1:A10 单元格的数移动到 B1:B10 单元格:

复制内容到剪贴板

代码:

```
Sub mycut()
 [a1:a10].Cut [b1]
End Sub
```

三、删除单元格

删除一个单元格的命令也很简单,调用 Range 对象的 Delete 方法即可。同我们用鼠标操作一样，同样有四个选项，下面以删除 A1 单元格为例:

复制内容到剪贴板

代码:

```
[a1].Delete Shift:=xlUp ′  下方单元格上移
[a1].Delete Shift:=xlToLeft ′   右侧单元格左移
[a1].EntireRow.Delete ′  整行删除
[a1].EntireColumn.Delete ′  整列删除
```

只写成[a1].Delete时，默认为“下方单元格上移”。

Range对象的End属性

Range对象有一个End属性，也许你在阅读别人的例子里经常会看到如“[A65536].End(xlUp).Row”这样的语句，这就是End属性的利用。

End属性应用于Range对象，同样也返回一个Range对象。该对象代表包含源区域的区域结尾处的单元格。如果你还是不明白，那请你在工作表里试一试，分别按Ctrl+上、下、左、右方向键，看看得到的是什么？End属性返回的单元格就相当于在源单元格按住Ctrl键+上(或下、左、右)方向键所得到的单元格。它的语句格式简单写为:

源区域.End(移动方向)

移动方向可以是:

xlToLeft：向左移动，相当于在源区域按Ctrl+左方向键。

xlToRight：向右移动，相当于在源区域按Ctrl+右方向键。

xlUp：向上移动，相当于在源区域按Ctrl+上方向键。

xlDown：向下移动，相当于在源区域按Ctrl+下方向键。

所以在程序设计的过程中，很多人喜欢用“[A65536].End(xlUp).Row”语句来返回A列的最后一个非空单元格的行号。

如果当前活动工作表的A列填写的是学生的姓名，现在我们想在表中A列最后一个学生的后面增加一个叫“张青”的学生，

则代码为:

复制内容到剪贴板

代码:

```
ActiveSheet.Cells([A65536].End(xlUp).Row + 1, 1).Value = "张青"
```

语句 [A65536].End(xlUp).Row + 1 中，行号加 1 的目的是在最后一个非空单元格的下一个单元格里输入数据。需要注意的是：如果你的 A 列全为空的话，“[A65536].End(xlUp).Row”这个语句返回的是 1。

我们来做一个简易的学生信息录入的例子:

（1）点工作表中的“录入资料”按钮，弹出“录入”窗体;

（2）录入信息后，点击“确定”按钮即完成记录追加;

（3）不考虑录入内容是否规范。

详细请看附件， 信息录入.rar (11.09 KB)

信息录入.rar (11.09 KB)

附件没有设密码，希望大家尝试过后再看代码!

汇总多工作表花名册

上一贴中，“花名册分类”的题目主要是检验大家对循环语句、判断语句及一些单元格数据处理的方法。因为目标工作表和源工作表的格式都是一样的，所以处理起来也相对比较简单，无非就是给数据搬了个家。所以可以用我们前面贴子里说的 Copy 的方法，当然，也可以直接等于源区域的 Value 属性，即：目标

区域.value=源区域.value。比如我们想把 Sheet1 里的第一行数据复制到 Sheet2 的第一行，可以用以下两句代码中的任意一句:

复制内容到剪贴板

代码:

```
Sheets("sheet1").Rows(1).Copy Sheets("sheet2").Rows(1)
```

代码:

```
Sheets("sheet2").Rows(1).Value=
Sheets("sheet1").Rows(1).Value
```

这一点，在“宝贝”的作业里已经用到了。

上次的作业是对名册进行分类，进行原样的搬家。这一次的作业还是搬家，和上次不一样的是，这次是将名册汇总到一个工作表，而且只汇总相应的内容，不再是那种原封不动的搬家了。当然，原理还是一样，如果你会了前一题，这一题对你来说，就显得很小儿科了。

设计一个程序，能否实现目的很重要，实现目的的同时，高效率的程序也是我们追求的。但是我们现在才开始，大家不必太过于考虑效率的问题，管他什么代码，只要能实现目的即可。在附件里，我加了计算程序执行时间的代码，你不用去管它，之所以加上这段代码，是因为在下面的帖子里我可能会用到。

复制内容到剪贴板

代码:

```
Sub mysub()
```

```
    Dim start As Double
    start = Timer

      '在这里添加你的代码

    MsgBox "程序共执行了" & Timer - start & "秒!"
End Sub
```

作业见附件， 学生花名册汇总.rar (21.53 KB)

学生花名册汇总.rar (21.53 KB)

下载次数: 218

2009-8-27 12:53

此贴到此为止，咱们不见不散，楼下见!

新建工作表

在“花名册分类”的作业里，moon2778 加入了为不同地区建立工作表代码，让“沧海一滴”和“宝贝有酒窝”两个朋友都跟贴问了新建工作表的相关问题。有需要就有市场，呵呵.....那我再在这里简单讲一讲我对新建工作表的认识，希望对大家学

习有帮助。

新建一个工作表（Worksheet)，要调用工作表(Worksheet对象）的 Add 方法（可以建制一个新建工作表的宏来查看相关的代码）。

如果我们想在当前工作表中新建一个工作表，代码可以写为:

复制内容到剪贴板

代码:

Worksheets.Add ′新建的工作表的位置在当前活动工作表的前面

此时，新建的工作表 Excel 默认把它放在活动工作表的前面，如果我们需要改变它的位置也是可以的。例如:

复制内容到剪贴板

代码:

Worksheets.Add before:=Worksheets("sheet2") ′在工作表 sheet2 之前添加一个工作表

Worksheets.Add before:=Worksheets(1) ′在最前面添加一个工作表

Worksheets.Add after:=Worksheets("sheet2") ′在工作表 sheet2 之后添加一个工作表"

```
                                        Worksheets.Add
after:=Worksheets(Worksheets.Count) '在最后面添加一个工作表
```

试了以上的代码，我们会发现，新添加的工作表都是采用“老大、老二、老三.....”的命名方式，如果你需要为新添加的工作表改一个名字，也是可以的，只需要再添加一句代码即可:

复制内容到剪贴板

代码:

```
    Worksheets.Add before:=Worksheets(1) '在最前面添加一个工作表
    ActiveSheet.Name = "abc" '将新建的工作表更名为“abc”
```

如果想更简单点，这句代码还可以合二为一:

复制内容到剪贴板

代码:

```
     Worksheets.Add(before:=Worksheets(1)).Name = "abc" '在最前面添加一个名称为 “abc” 的工作表
```

有时，也许需要一步就添加多个工作表，那可用下面的代码:

复制内容到剪贴板

代码:

```
Worksheets.Add Count:=3 '添加三个工作
```

但是一次性添加多个工作表时，好像就无法同时对其重新命名了。

关于 247 楼朋友的提问

引用:

原帖由 沧海一滴 于 2009-8-27 19:13 发表

楼主你好!

我可能没有表达完整，现再重新解释如下:

1、“花名册”中有固定的工作表三个或四个，而“花名册”只是其中之一工作表;

2、用“moon2778”作业为例子，当没有建立子工作表情况下，建立时，如何在固定工作表后重新建立分配后的工作表。

moon2778 的例子里，在新建工作表之前，先执行了一个删除工作表的操作，把工作薄中所有不是当前工作表的所有工作表全部删除。如果你的工作薄里原有子工作表的话，建议你用我在 213 楼

（http://club.excelhome.net/viewthread.php?tid=470603&page=22#pid3110867）的回复里的方法。我们可以在新建工作表的时候，先在工作表集合里看一下是否有我们需要添加的工作

表，如果没有，则执行添加操作。这样，就不用去删除所有的工作表了。

还是拿“花名册汇总”例子的工作表来说吧，如果我们需要自己新建相应的工作表，并命名，可用以下代码:

复制内容到剪贴板

代码:

```
Sub shtadd()
 On Error Resume Next ′出现错误继续执行下一句
   Dim xrow As Integer
      xrow = Worksheets("外在本就读花名册").[b65536].End(xlUp).Row
   For i = 3 To xrow
     If Worksheets("外在本就读花名册").Cells(i, 8).Value = "清镇" Then
          If Worksheets(Worksheets("外在本就读花名册").Cells(i, 9).Value) Is Nothing Then ′  ′判断是否存在该地区的子工作表
          Worksheets.Add(after:=Worksheets(Worksheets.Count)).Name = Worksheets("外在本就读花名册").Cells(i, 9).Value ′新建表并命名
       End If
   nd If
```

```
 Next
If Worksheets("清镇市外") Is Nothing Then          '判断是否存在“清镇市外”工作表
    Worksheets.Add(after:=Worksheets(Worksheets.Count)).Name = "清镇市外"
End If
End Sub
```

以上代码是按 moon2278 的思路更改，但只是添加表格并给表格命名，并没有对表格进行设置，所以如果需要对表格进行格式设置（比如添加表头），可再自行添加相应的代码。

表头都是一样的，大家有没有想过用复制来代替新建？复制后，将第三行以下的值全部删除，再对工作表重新命名，也是可以的！

PS：我休息两天，周一见！

引用：

原帖由 宝贝有酒窝 于 2009-8-30 15:51 发表

我的作业

原本是想在利用字段调用子表合并到汇总表中的但是目前的技术实现不了

另外绿色底纹被我覆盖了……

底纹我只是给来作标记的，盖了没事。如果想不覆盖底纹，可以

使用选择性粘贴或直接等于 Value，都可以的。

对于你的代码，能实现需要得到的结果，但可以再完善一下。比如，其实程序中并不需要定义那么多的变量，也不用使用两层循环。这样只会让程序进行许多多余的计算，增加程序执行的时间。

楼上的 jiminyanyan 朋友已经给你修改了一下，可以参考一下，但他在修改时，可能不小心把“为第三列(性别)添加数据的代码”删到了。我按照你的思路，把代码整理了一下（几乎和楼上的一样），供参考。

虽然我说了，只需要完成目的即可，但你你也可以看一下程序执行的时间，对比一下两次代码的区别，同样的思路，甚至是同样的语句，整理过的效率可是你原来的效率的七倍以上哦。

(回复)宝贝有酒窝的学生花名册汇总.rar (23.92 KB)

VBA 数组（一）

一、什么是数组

对于数组，在前面 7 楼的贴子里我们已经简单介绍了一些简单的概念，如果你忘记了，请回过头去花两分种读一读。

读完了也许你还是不清楚，没关系，请接着往下看。

新学期开始了，我上七（1）班数学，学生太多了，我老是

记不住学生的名字，还好班主任为了便于管理，为班上的学生都编了学号，所以上课时点名我习惯性只点学生学号，我说，5 号同学请上黑板演示一下你的解题过程，10 号同学下课后请替老师把大家的作业收起来……下课后，在办公室，和大家聊天，我更不叫学生名字而直接统称为“七（1）班”。

待我学习了 VBA 的数组后，我忽然明白了，这种简化的称呼方式原来就是“数组”。

这里，我们是把一群学生聚在了一起组成了“七（1）班”（数组名）这个集合，班级里的学生我们通过“学号”（索引号）对其进行区分。上课时，我问：“5 号同学是谁？”，一个同学站起来答：“我叫 moon2778”，于是，我知道了，“七（1）班”这个数组里的第 5 个元素就是“moon2778”。用数组来表示就是“七（1）班(5)”，也就是数组名加索引号。好了，当别人果问你：“‘七（1）班(5)’的同学是谁？”时，你一定要知道，是“moon2778”。

那么，什么是数组，你知道了吧？它有什么特点，我们可以试着简单小结一下：

1、数组共享一个名字，即数组名；

2、数组是变量的集合，有多个元素；

3、数组中的元素按顺序保存在数组中，元素通过索引数字加以区分；

4、数组是变量。

二、定义数组

声名数组同声明其他变量的方式一样,唯一不同的是数组有大小，包含多个单个变量。

初一学生报名入学后开始分班级，领导说，七（1）班分 50 个学生（确定了班级学生的个数最多只能有 50 个学生，再多一个都不行，因为没有座位了。），知道有 50 个学生后，老师拿到学生花名册就开始给学生编学号，建学籍。1 号是谁？2 号是谁？……50 号是谁？

这就是一个定义数组,并给数组赋值的过程,用语句来表示,可以这样:

```
Dim 七（1）班(1 To 50)
七（1）班(1) = "张三"
七（1）班(2) = "李四"
七（1）班(3) = "王王"
.....
七（1）班(50) = "赵六"
```

这里，我们定义了一个 50 个元素的一维数组，并分别给数组的元素赋值。当然，这里用的只是一个模拟的语句。如果我们需要定义一个数组 arr，存放 1 到 100 的整数集合，则代码为:

代码:

```
Sub ggsmart()
```

```
    Dim arr(1 To 100) As Integer ′声明数组名称及数据类型
    For i = 1 To 100 ′循环，为数组的元素赋值
      arr(i) = i
    Next
End Sub
```

其实“Dim arr(1 To 100) As Integer”我们也可以写成“Dim arr(99) As Integer”，这样，定义的数组同样是 100 个元素的一维数组，因为如果不指定起始索引号的话，VBA 默认从 0 开始。

当然，你也是可以通过 OPTION BASE 来修改数组默认的起始序号，在模块的第一句写上“OPTION BASE 1”这样，默认的数组起始序号就是 1 了。

学习累了，恰好第二节课刚下，课间操时间，顺便可以到操场练练。可是，有些同学不认真啊，这不，老师在广播里叫上了：“七年级，（1）班的第五个同学请认真一点，七年级，（2）班的第三个同学注意队形……”

天了，我惊奇地发现，原来我们生活中有这么多“数组”，可是，你知道上面的这两个同学共享了一个什么名字吗？

是的，“七年级”就是它们共有的名字。

新生分班时，校长就说了，550 名新生，每班 50 人，分成 11 个班，到各个班级里，老师都给学生分了学号（1 到 50 号），11 个班级就有 11 个 1 号，也有 11 个 50 号。晕，一个班的学生

名字我都记不住，如果 11 个班的学生全站在一起，我该怎么称呼？

我高兴了，还好，我会数组。找到“moon2278”同学，我会说“七年级，1 班，5 号同学”，哈，难不了我。如果“宝贝有酒窝”同学是初一（9）班的 25 号，没关系，我也会说“七年级，9 班，25 号同学”。

哇，好简单，前面加个班级的序号区分就行了。

这里，“七年级”实际上是一个由 11 个一维数组（11 个班）组成的二维数组，它的元素的个数是 11*50（11 个班，每个班级 50 名学生）个。习惯性，大家会说它是一个 11*50 的二维数组。

如果要写成代码，那就是：

```
Dim 七年级(1 To 11, 1 To 50)
七年级(1, 1) = "张三"
七年级(1, 2) = "李四"
'.....
七年级(2, 1) = "钱林"
七年级(2, 50) = "林平"
'.....
七年级(11, 50) = "周六"
```

我们说上面的数组是一个 11*50 的二维数组，放到表格里就相当于是一个 11 行 50 列的数组。

如果我们想定义一个二维数组，第一列存放 1 到 100 的奇数，

第二列存放 1 到 100 的偶数，代码为:

代码:

```
Sub ggsmart()
 Dim i As Integer, j As Integer
 Dim arr(1 To 50, 1 To 2) As Integer '定义一个 50*2 的二维数组（50 行，2 列）
 j = 1     '数组 1 to 50 的索引号
   For i = 1 To 100 Step 2
     arr(j, 1) = i '奇数赋值
     arr(j, 2) = i + 1 '偶数赋值
     j = j + 1 '索引号加 1
Next
[a1:b65536].Clear '清除原有数据
[a1].Resize(50, 2) = arr '将数组赋值给 A1 单元格向下 50 行，向右 2 列的区域
End Sub
```

1 到 100 的奇偶数.rar (6.63 KB)

1 到 100 的奇偶数.rar (6.63 KB)

把代码复制到模块里，按 F5 可以看到程序运行的效果。

上面，我们讲的是一维数组和二维数组，你也可以按同样的思路来理解三维、四维数组，甚至五维、六维……或者更多。

无非就是将称呼 moon2278 同学的方式改为“XX 中学、七年级、1 班、5 号同学”或“XX 市、XX 中学、七年级、1 班、5 号同学”而已，如果你喜欢，你还可以将这个称呼扩展，随你了，这里不再举例。

三、动态数组

声明一个数组，必须指定数组的大小，即它是几维数组，有多少个元素。

但有些时候，我们在定义数组的时候，并不能确定数组的大小，这时，我们在首次定义数组的时候括号内为空（当然先指定一个大小也是可以的，但纯属是多余），在程序过程中用 Redim 语句去重新指定数组的大小，这就是动态数组。

比如一个工作表的 A 列存储了学生姓名，现在我们需要把把有姓“张”的学生存储在数组 arr 中，预先我们并不知道 A 列姓张的学生有三十个还是五十个，所以，我们在定义时代码可以这样:

代码:

```
Sub ggsmart()
 Dim i%, xrow%, j%, xcount%
 Dim arr() As String
 xrow = [a65536].End(3).Row '最后一个非空单元格行号
 j = 1     '数组索引号
```

```
    xcount=
Application.WorksheetFunction.CountIf([a1:a65536], "张*") '统计有多少姓张的学生
    ReDim arr(1 To xcount) '重新定义数组大小，元素共有xcount个
    For i = 1 To xrow
      If Left(Cells(i, 1).Value, 1) = "张" Then
        arr(j) = Cells(i, 1).Value '给数组元素赋值
        j = j + 1 '索引号加1
      End If
    Next i
    [b1:b65536].Clear '清除原有数据
[b1].Resize(xcount,1)=Application.WorksheetFunction.Transpose(arr) '将数组输入单元格区域
    End Sub
```

姓“张”的学生.rar (7.16 KB)

姓“张”的学生.rar (7.16 KB)

四、其他创建数组的几种方式

其实在上面，我们已经接触了如何创建数组了。无非就是像给学生排座位一样，你坐第一排，我坐第二排，他坐第三排，然后你们大家统称为第一组。

好简单，这里，我再介绍几种上面没讲到，但常见的方法:

1、使用 Array 创建数组

看以下代码:

代码:

```
Sub ggsmart()
 Dim arr()
 arr = Array(1, 2, 3, 4, 5, 6, 7, 8, 9, 10) '将 1 到 10 十个自然数赋给数组 arr
 [a1].Resize(1, 10) = arr '将数组的值输入单元格
End Sub
```

这里使用 Array 函数将 1 到 10 的自然数赋给数组 arr 这里，我就不上传附件了，大家将代码复制到模块里查看效果。

引用:

提醒一点: 使用 Array 创建的数组索引号默认从 0 开始，当然，如果已经用“OPTION BASE 1”语句指定了的另当别论。

需要提醒的是: 如果赋的值是文本，那需要用英文引号引起来，如:

代码:

```
arr = Array("张祝", "徐国军", "陈茜", "王欢", "许松", "张小海", "陈飞", "张明元", "徐国东", "张款")
```

2、使用 Split 创建数组

Split 的作用是把一个文本，根据指定的分隔符，建立一个数组。感觉就有点像我们基础操作中的“分列”。不同的是“分

列”操作是将一个单元格的文本按指定的分隔符分开，存放在同行的多个单元格里，而 Split 的作用是将一个文本（也可以是存放在单元格里的）按指定的分隔符分开，存放在一个数组变量里。

如果你使用过“分列”，再来看这个，就简单多了。

还是上面的那串姓名“张祝，徐国军，陈茜，王欢，许松，张小海，陈飞，张明元，徐国东，张款”，只是这时，这些姓名是一个完整的字符串，此时，如果我们想把这些姓号按逗号（，）分开，存放在数组 arr 里，则代码为：

代码：

arr = Split("张祝，徐国军，陈茜，王欢，许松，张小海，陈飞，张明元，徐国东，张款"，"，")

你可以用上面的方式将生成的数组输入到单元格查看效果。当然，也可以在立即窗口里查看，这样就更方便了。

特别强调一点：用 Split 把文本转换成数组，索引号总是从 0 开始。不管你是否使用了 Option Base 语句，这点和 Array 不同。

3、通过 Range 创建数组

（1）循环给单元格赋值

比如想把 A1:A100 单元格的值给数组 arr，则代码为：

代码：

```
Sub a()
 Dim arr(1 To 100)
```

```
    For i = 1 To 100
      arr(i) = Cells(i, 1)
    Next
End Sub
```

(2)直接等于单元格区域

这个很简单，直接数组名等于单元格区域就可以了。

比如想把 A1:A100 单元格的值给数组 arr，代码还可以简单写为:

代码:

```
arr = [a1:a100]
```

好了，很简单，我就不再多说，大家自己去练习吧。

两种方法哪种的效率高显而易见吧？呵呵……有兴趣，可以自己去比较一下。

五、UBount 和 LBound 函数

我们定义数组时，我们可以指定数组的起始索引号不为 0，而为 1 或其他的任意数。比如 Dim arr(5 to 6)定义的这个数组起始索引号为 5,共有两个元素。

那么，对于一个已经定义了的数组，我们想知道它的最大索引号（上界）和最小索引号（下界）是多少，怎么办呢？别慌，用 UBound 和 LBound 两个函数就可以了。

如，要想知道数组 arr 的上界是多少，则：UBound(arr)

要想知道数组 arr 的下界是多少，则：LBound(arr)

想知道数组有多少个元素，则：UBound(arr)-LBound(arr)+1

如:

代码:

```
Sub ggsmart()
 Dim arr(10 To 50)
 MsgBox "数组的上标是: " & UBound(arr) & Chr(13) _
      & "数组的下标是: " & LBound(arr) & Chr(13) _
      &"数组的元素个数是: " &UBound(arr) -LBound(arr) + 1
End Sub
```

如果是一个二维数组，想知道它的上界是多少，就得指定维数，看一看下面的例子:

代码:

```
Sub ggmsart()
 Dim arr(1 To 10, 1 To 100)
 MsgBox "第一维的上界是:" & UBound(arr, 1) & Chr(13) & _
 "第二维的上界是:" & UBound(arr, 2)
```

End Sub

六、Join 函数

Join 的作用和 Split 的作用相反。Split 是将字符按指定字符转为数组，Join 是将数组以指定字符分开，连成一个字符串。如:

代码:

```
Sub ggsmart()
 Dim arr(1 To 10), i As Integer, txt As String
 For i = 1 To 10 '给数组赋值
   arr(i) = i
 Next i
 txt = Join(arr, ",") '将 arr 数组的元素连成字符串，用逗号作分隔符
 MsgBox txt
End Sub
```

Join 的第二参数，即分隔符可以不指定，在不指定时，默认为以空格作分隔符。

七、将数组输入单元格区域

1、使用循环输入

如将存储了 1 到 60000 的数的数组写入 A 列，则:

代码:

```
Sub test1()
```

```
'创建数组，并赋值
Dim arr(1 To 60000), i As Long
For i = 1 To 60000
  arr(i) = i
Next i
'将数组的值写入单元格（A 列）
[a1:a65536].Clear '清除原有数据
Dim irow As Long
Dim startime As Double
startime = Timer
For irow = 1 To 60000
  Cells(irow, 1) = arr(irow)
Next irow
MsgBox "数组写入共用了" & Timer - startime & "秒！"
End Sub
```

以上程序是使用循环，将数组里的元素逐个写入单元格，为了对比不同的输入方式的效率，我加入了计算时间的代码，有了前面的基础，相信读上面的代码对大家来说，不会太困难。

2、批量写入

实际上用一条语句就可以将数的内容写入单元格区域中，即：单元格区域=数组名

还是上面的例子，我们可改为：

代码:

```
Sub test2()
'创建数组，并赋值
Dim arr(1 To 60000), i As Long
For i = 1 To 60000
  arr(i) = i
Next i
'将数组的值写入单元格（A 列）
[a1:a65536].Clear '清除原有数据
Dim startime As Double
startime = Timer
[a1:a60000]=Application.WorksheetFunction.Transpose(arr)
MsgBox "数组写入共用了" & Timer - startime & "秒！"
End Sub
```

这里，我们用“[a1:a60000]=Application.WorksheetFunction.Transpose(arr)”代替了上面代码 For 循环，在我的电脑上，程序的效率就提高了 25 倍以上。

数组写入单元格.rar (8.04 KB)

数组写入单元格.rar (8.04 KB)

所以在大批量的数据需要写入时，我建议尽量不要用循环的方式写入。

这种批量写入的方式在上面的例子里其实我已经用到了，相

信大家不会陌生。

这里，我再补充两点我个人在学习过程中的心得:

（1）一维数组写入单元格区域时，这个单元格必须是水平方向的，也就是一个一行多列的单元格区域，如果想写入垂直区域，必须使用工作表的 Transpose 函数转换一下;

（2）和工作表的多单元格数组公式一样，数组写入单元格时，单元格的区域必须和数组的大小一致。

(回复)宝贝有酒窝的学生花名册汇总.rar (23.92 KB)

for_next-1 到 1000 自然数的和.rar (6.14 KB)

IF 语句.rar (6.7 KB)

11.jpg (35.5 KB)