**VBA for Excel 快速入门**

By Acylation

1. **概述**

VBA，全称Visual Basic for Applications，由Visual Basic发展而来，是一种面向对象的编程语言，能扩展Office软件的功能。特点是编程可视化、依赖于父应用而不能独立存在。

1. **准备工作**
   1. **创建一个启用宏的工作簿**

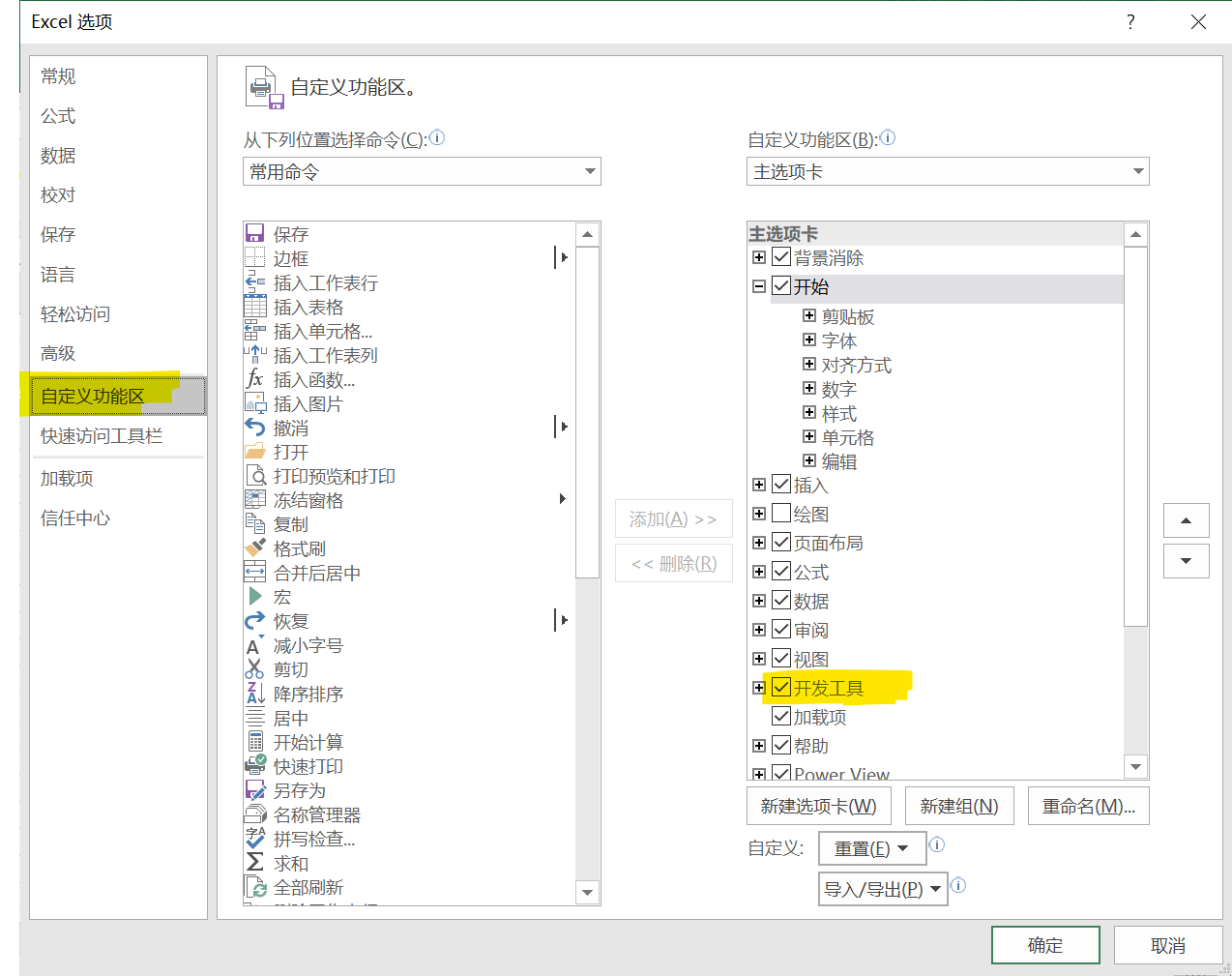
打开Excel，新建一个空白工作簿

保存，选择保存为\*.xlsm 格式

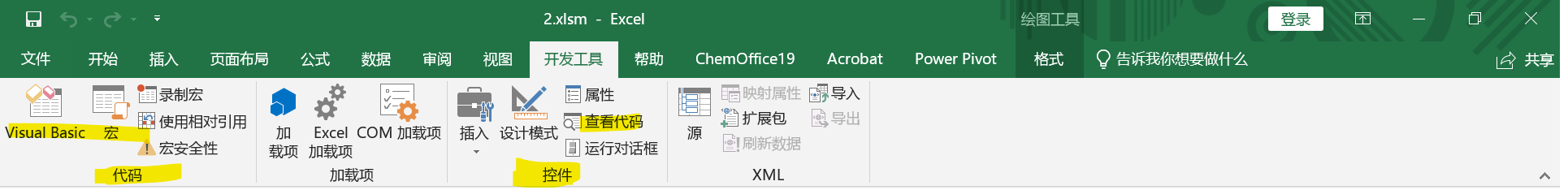


* 1. **设置选项卡**

进入“文件”-“选项”，选择“自定义功能区”，勾选右侧菜单中的“开发工具”选项



菜单栏设置后的效果如下，注意“代码”组和“控件”组，我们之后经常要用到



1. **开始编程**
   1. **示例1：Hello World！**

接下来我们来写第一个VBA脚本！，一些概念一开始会不懂，但先照着写哈！

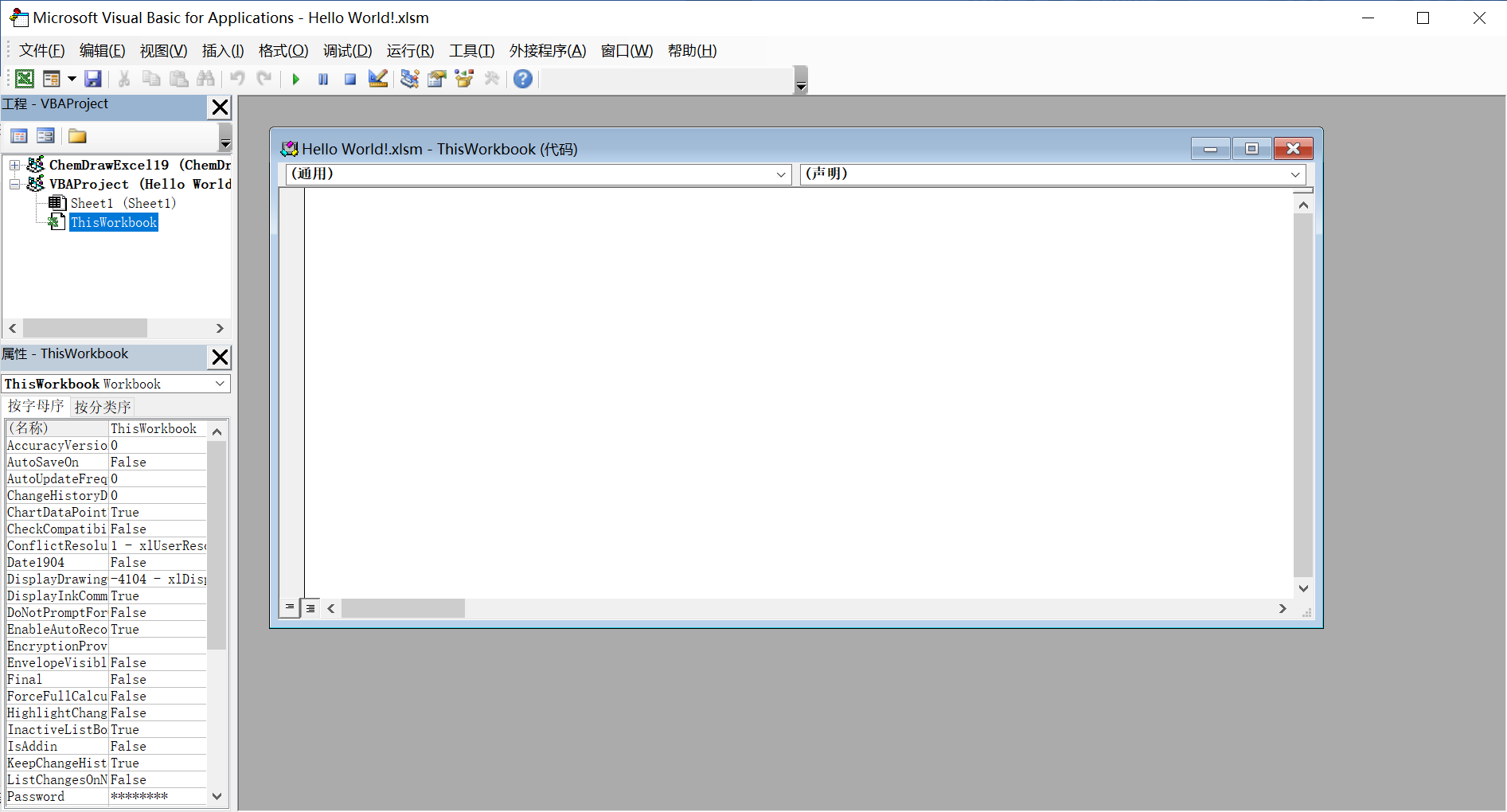
工作界面的介绍在第3节，编程中如果遇到界面相关的困难，可以先查看它~

打开\*.xlsm文件的时候，如果跳出“宏已被禁用”的提醒，请点击“启用内容”

[示例文件：1.xlsm](1.xlsm)

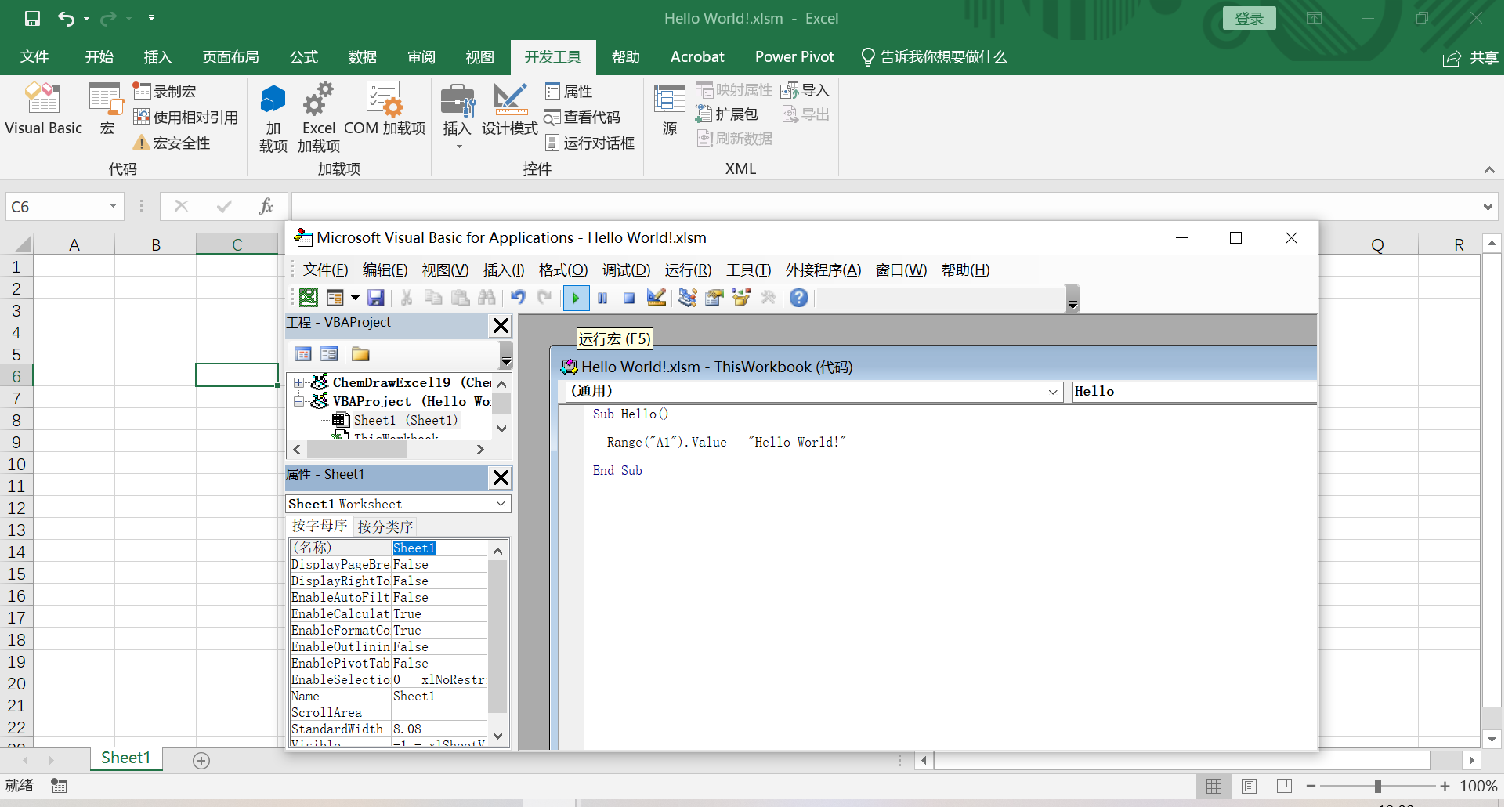
点击“代码”组中的“Visual Basic”按钮，进入VBA界面

双击左侧的“ThisWorkbook”进入代码界面,效果如下

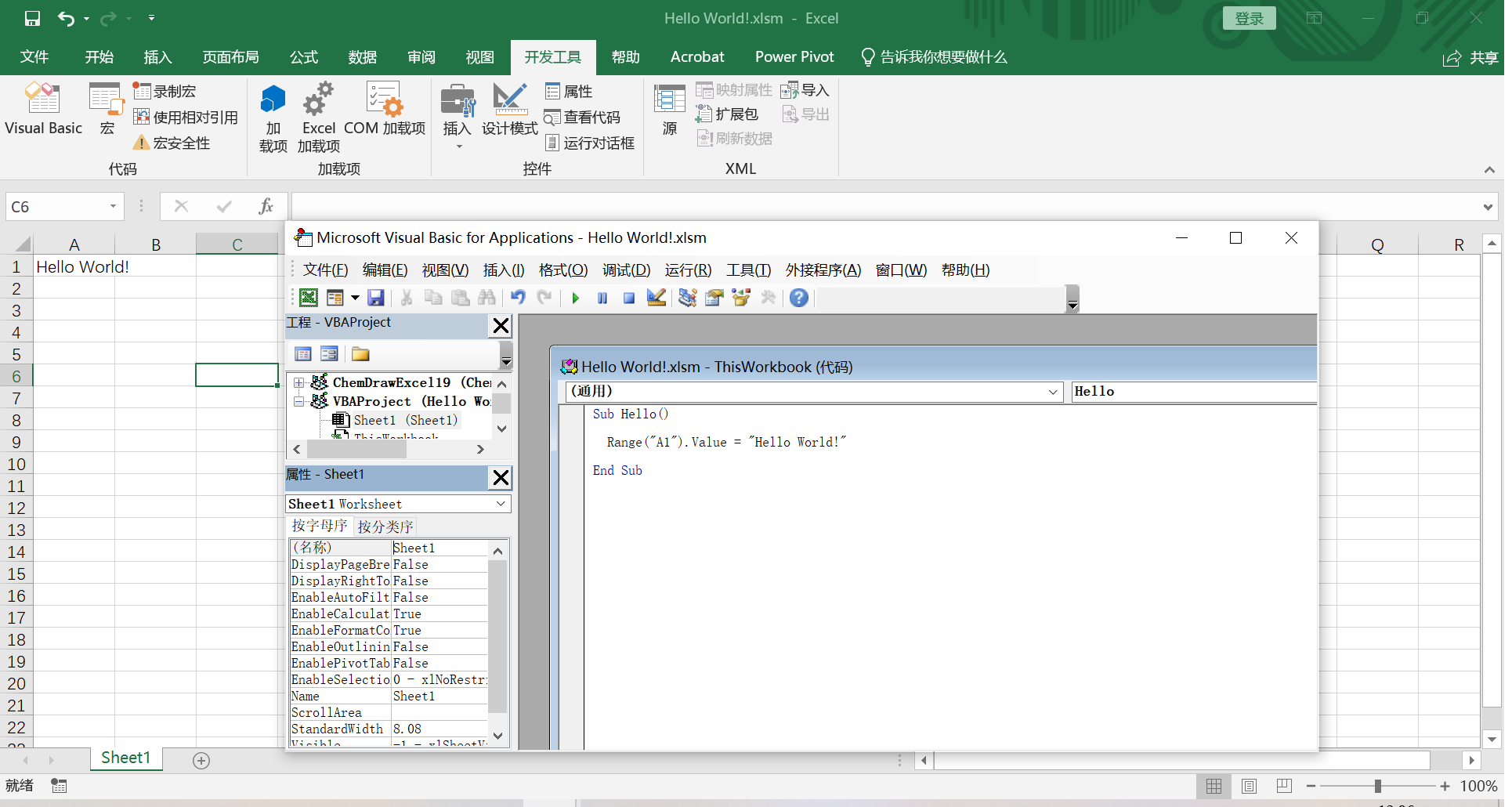


在代码窗口中输入以下代码，效果如下，注意标点应为半角英文

Range(“A1”).Value = “Hello World!”



点击运行按钮（看起来像播放键）或按F5键。若有窗口跳出，继续点 “运行”即可

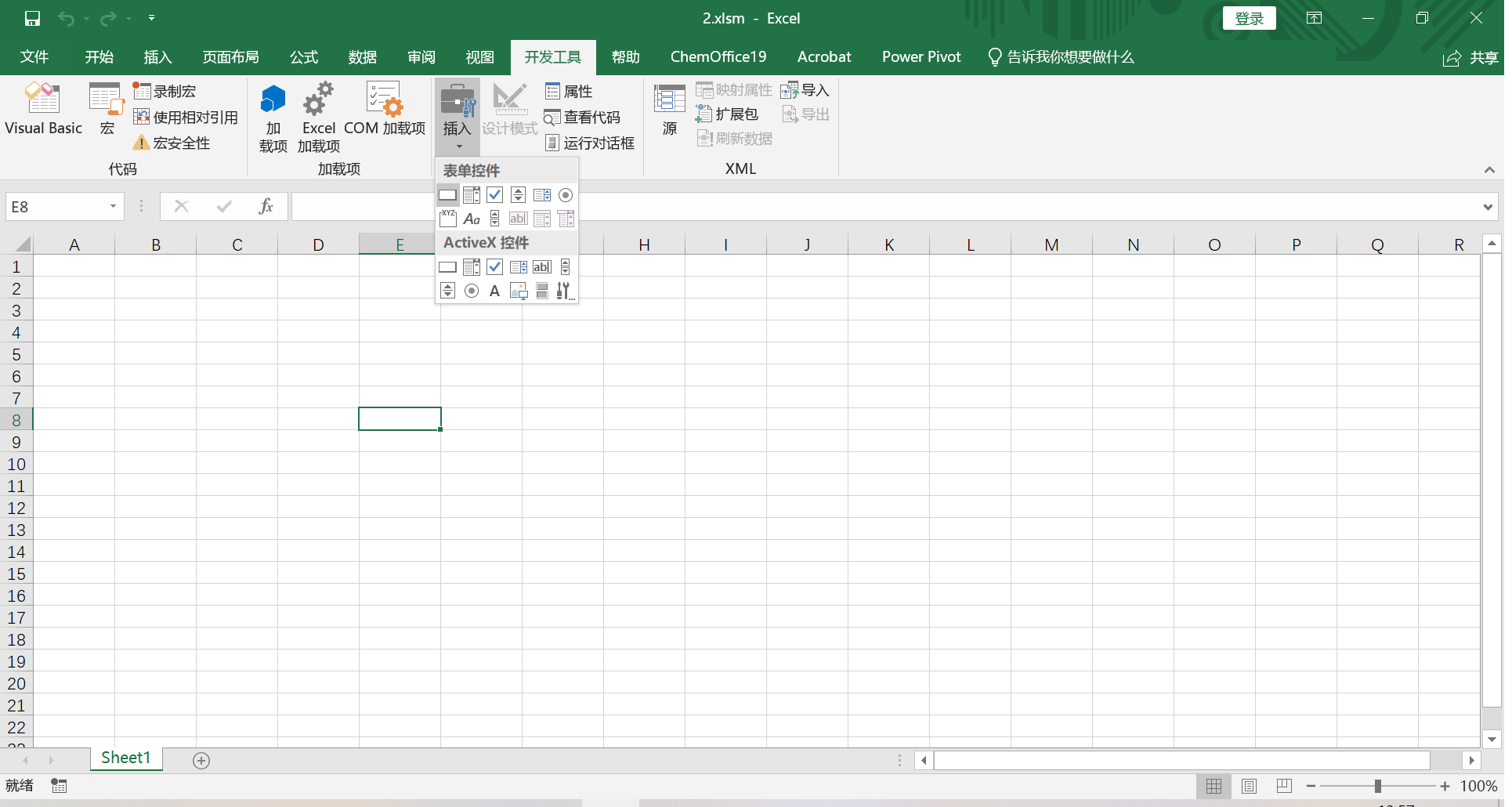


* 1. **示例2：加上按钮~**

[示例文件：2.xlsm](2.xlsm)

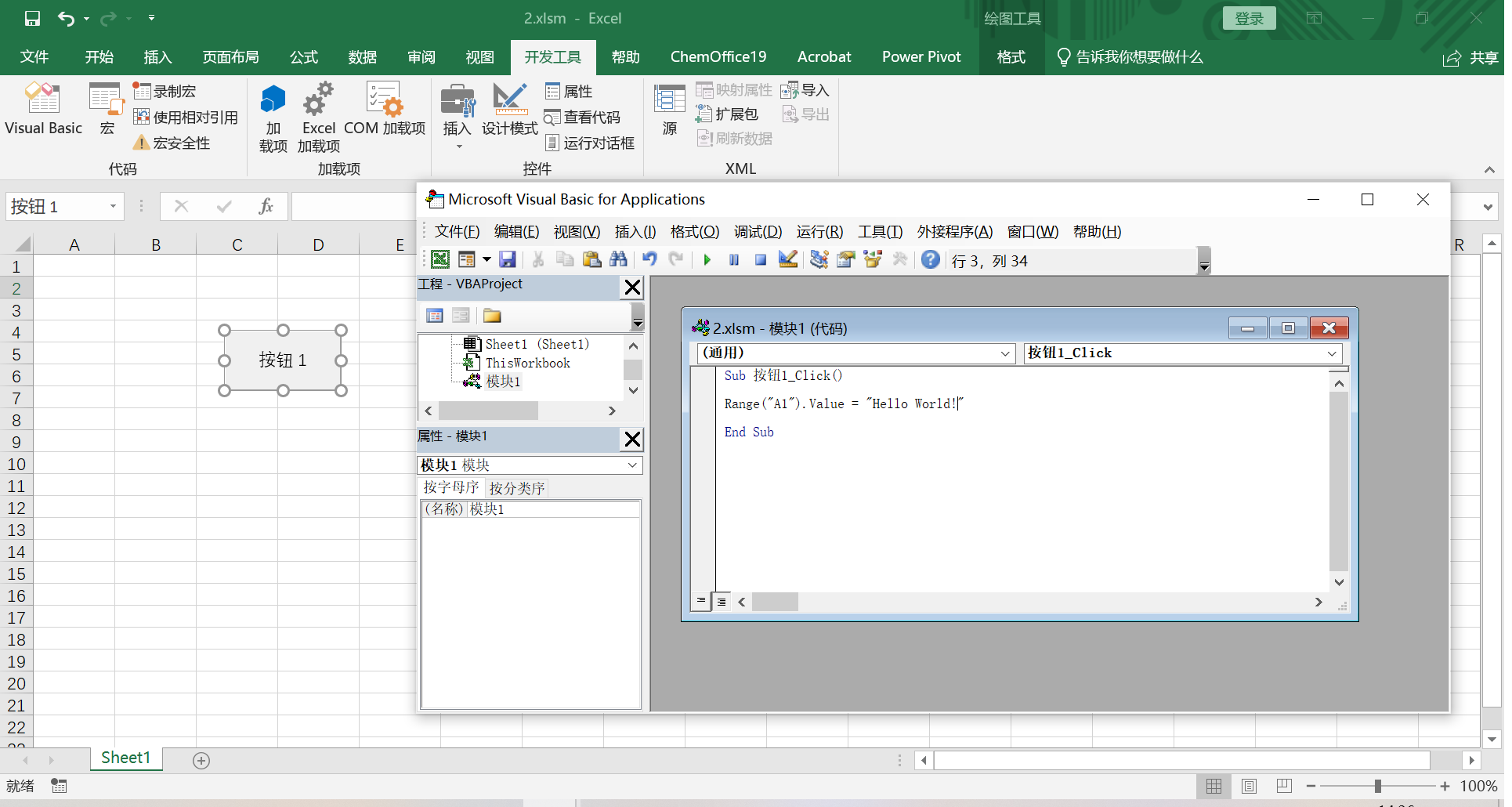
（这项任务可以沿用上一节中的那行代码）

如图操作，点击“开发工具”-“插入”-“表单控件”-“按钮（窗体控件）”（第一个）



随后光标变为十字，按下鼠标左键并拖动以绘制控件

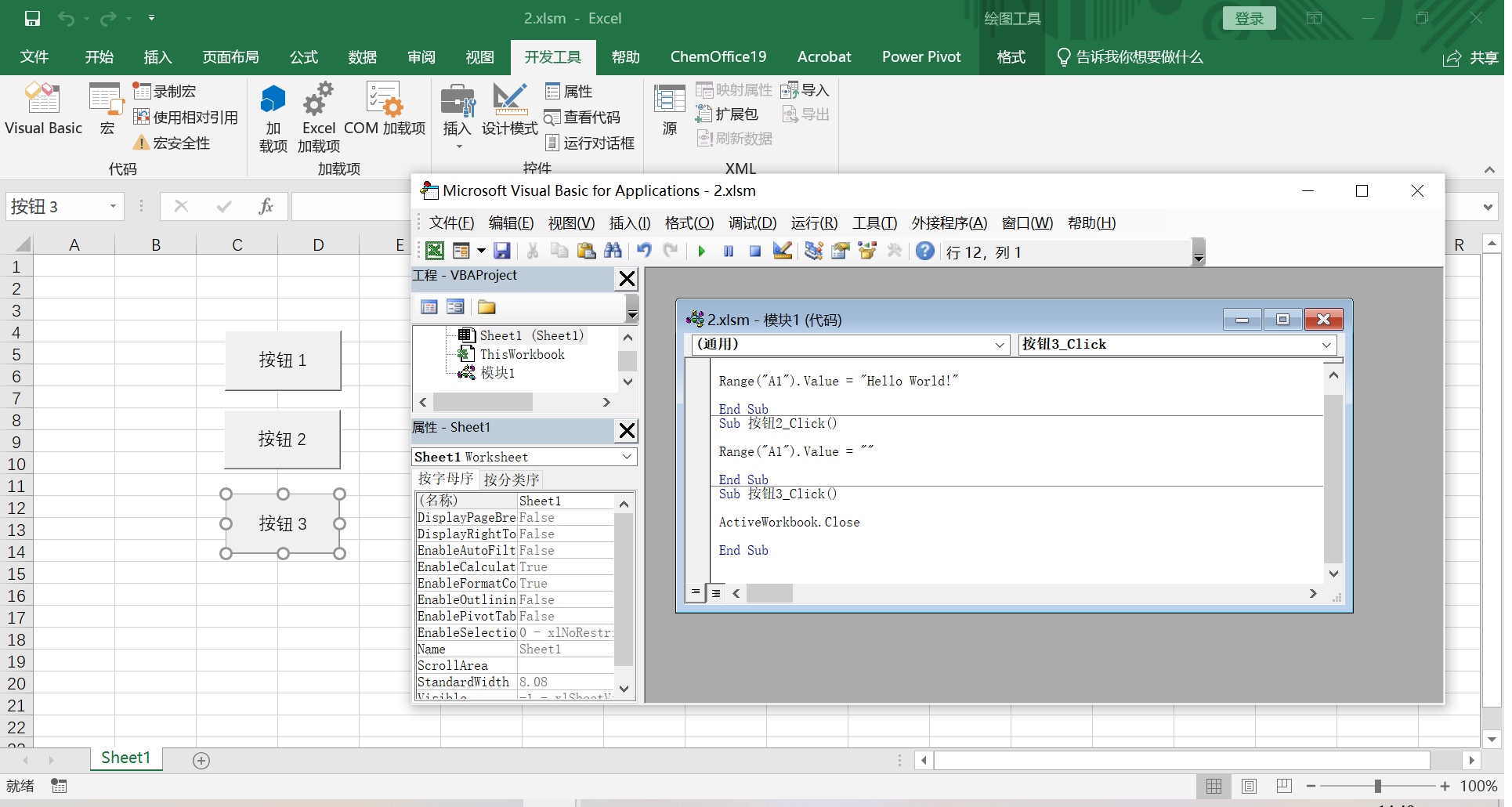
松开鼠标后会弹出一个窗口，点击右侧的“新建”，自动跳转代码窗口，写入以下代码



重复以上步骤，共绘制三个按钮，如下图所示输入各自的代码

在代码窗口中，按执行键或者F5键将运行光标所在位置的代码，试着执行一下

正常状态下，左键单击按钮将运行按钮对应的脚本，试着执行一下



接下来让我们对按钮进行一些编辑

代码窗口可以先关闭，稍后按下“控件”组中的“查看代码”按钮可以重新调出

编辑按钮方法说明：

右键单击按钮，按钮周围出现小白圈（如上图按钮三），说明按钮进入了可编辑状态

在可编辑状态下，左键单击按钮上的文字，可以对按钮上的文字进行修改

在可编辑状态下，按下左键可以在界面上拖动按钮，改变按钮的位置

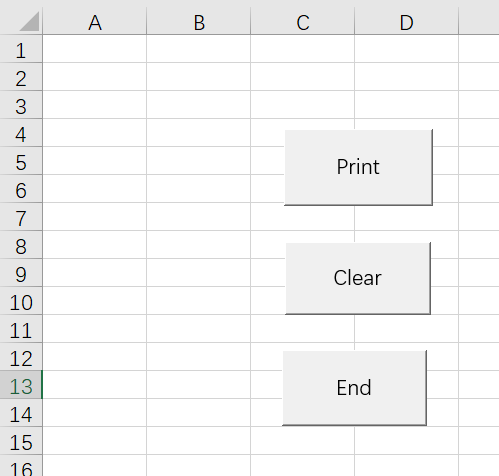
在可编辑状态下，左键拖拉小白圈进行按钮大小的调整

右键单击调出菜单，也可编辑文字，在“设置控件格式”中还可以执行更多操作

任务：

修改按钮文本，如右下图所示

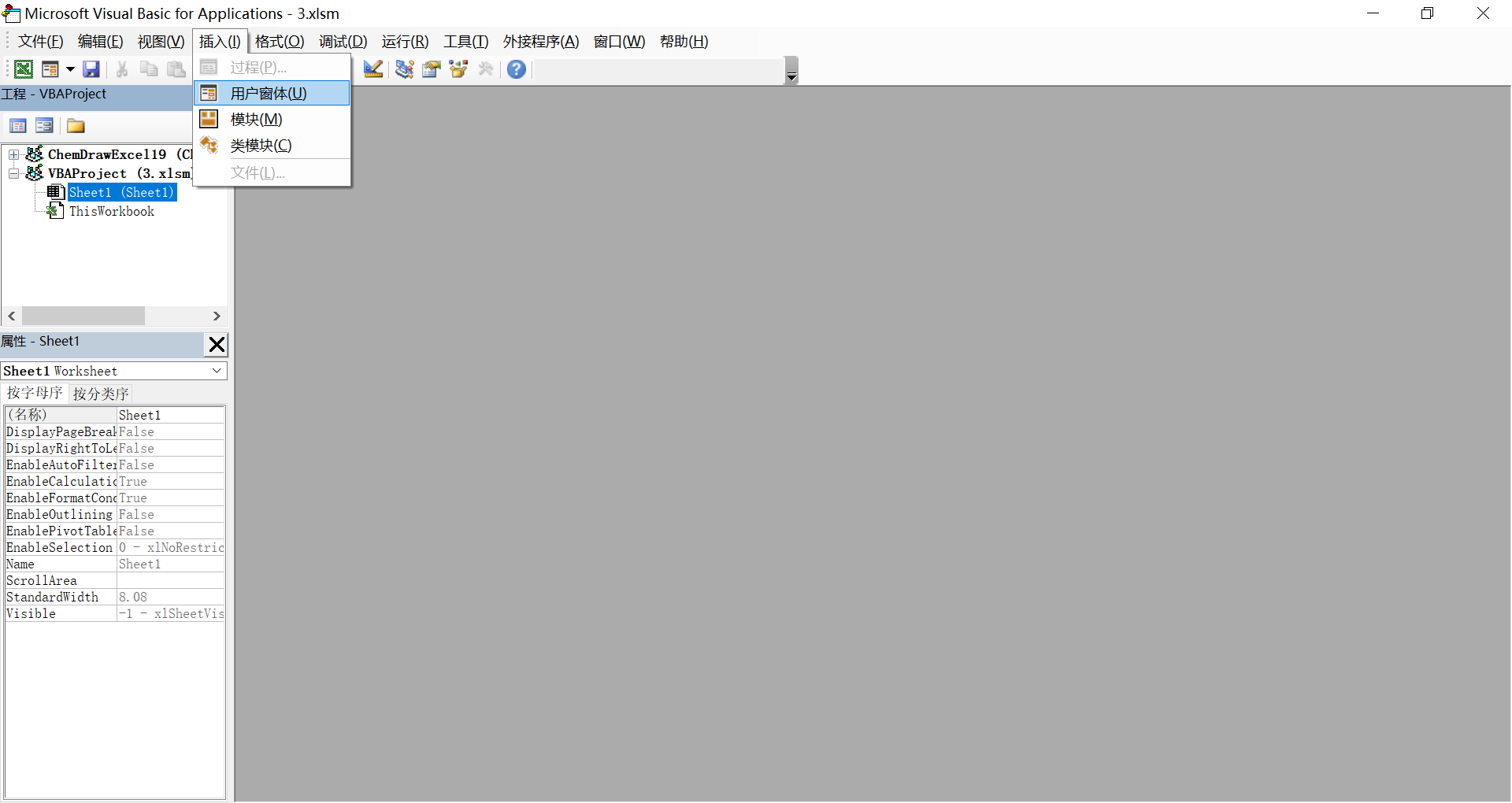
试着统一按钮大小并对齐，并尝试修改字体、字号和文字颜色

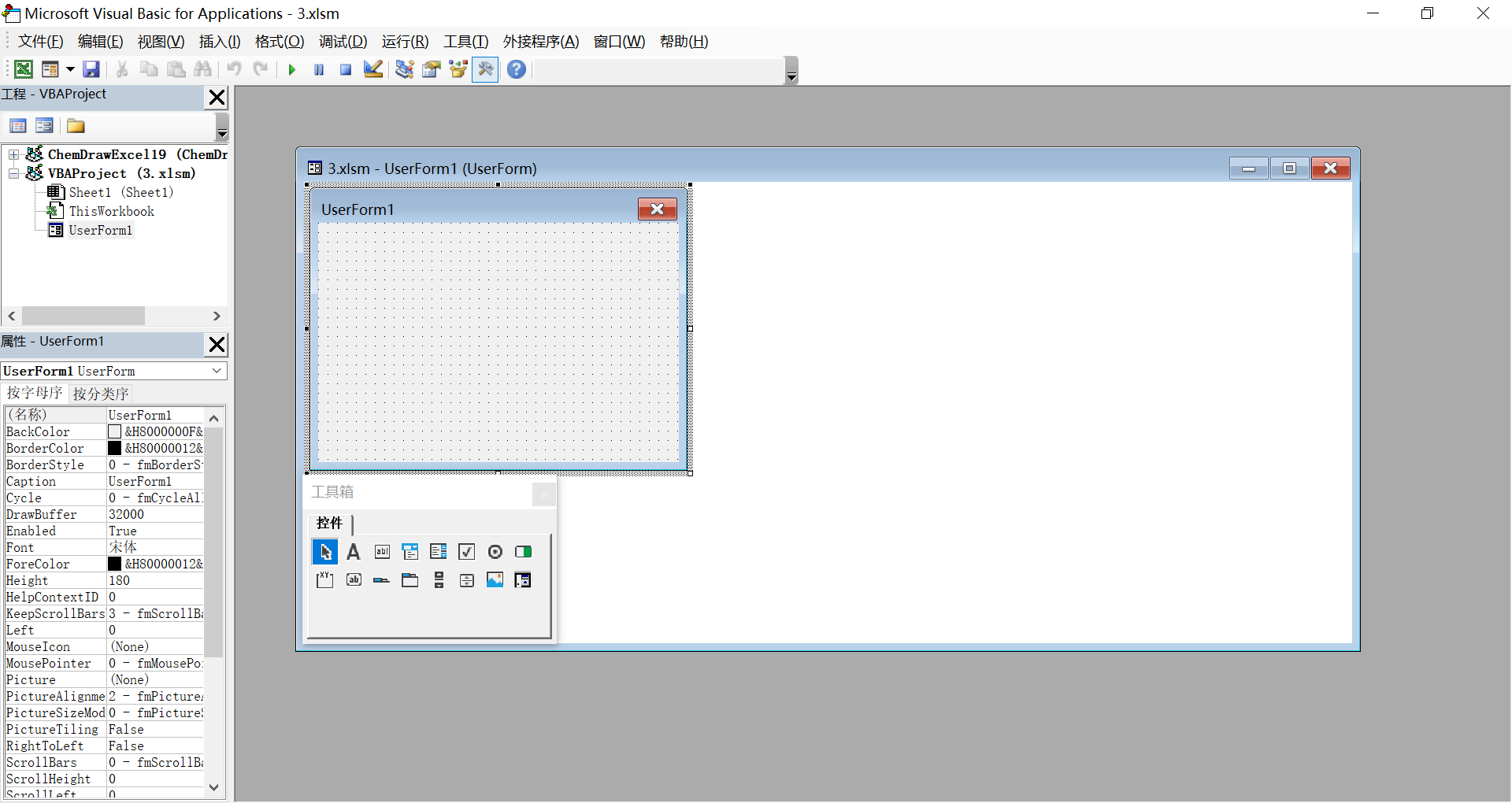
* 1. **写一个窗体！**

[示例文件：3.xlsm](3.xlsm)

点击“代码”组中的“Visual Basic”按钮，进入VBA界面，如图插入一个用户窗体

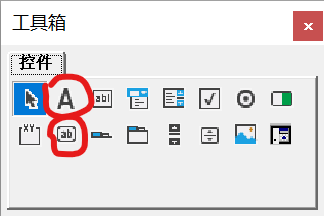


插入后效果如下

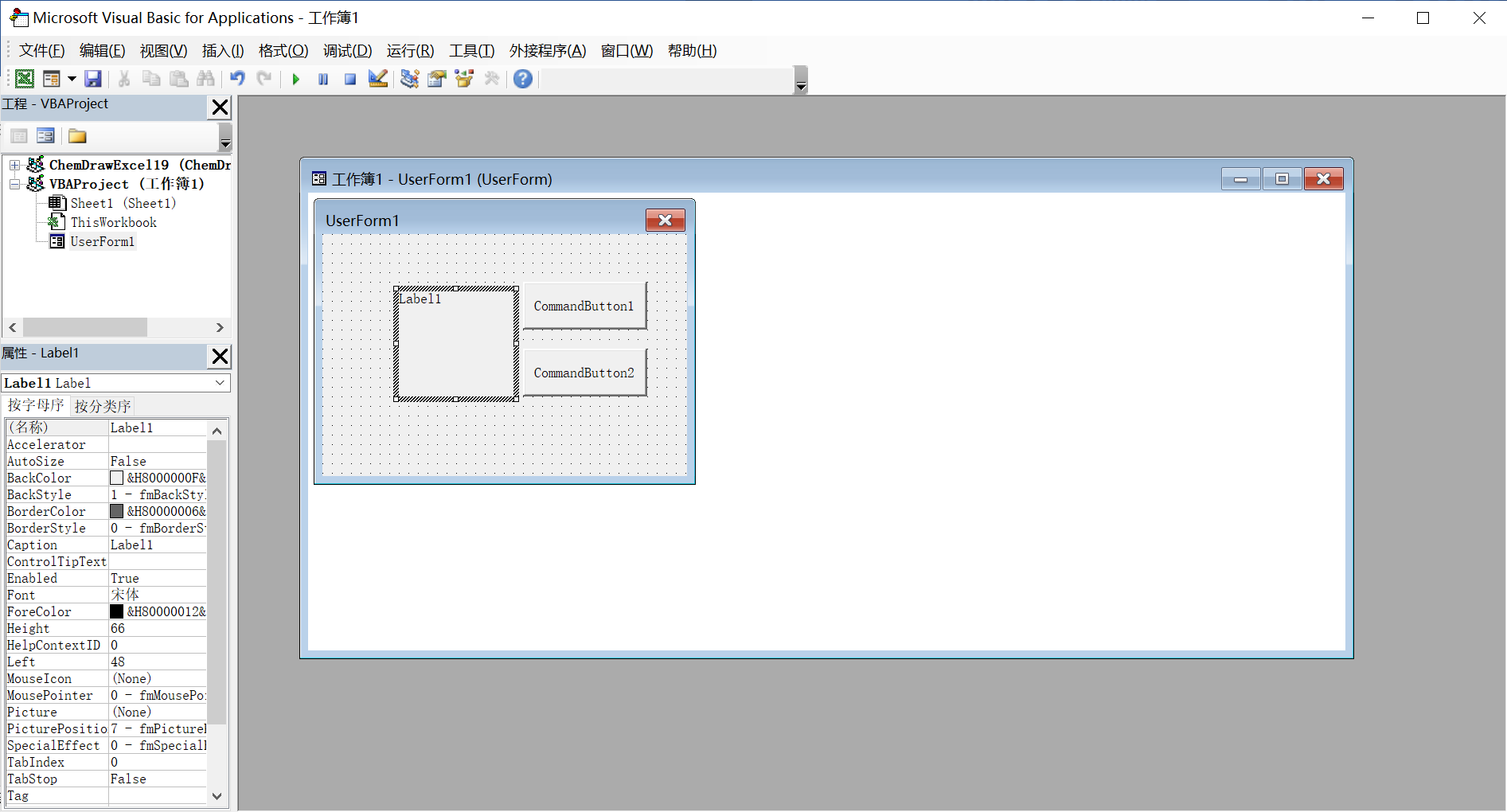


接下来我们放大看一下“工具箱”，将光标拖动到按钮上，会出现控件的名字

本程序将用到“标签”和“命令按钮”两个控件，分别是第一、二行的第二个



需要绘制一个标签和两个命令按钮，绘制方法与上一节相同，效果如下图所示



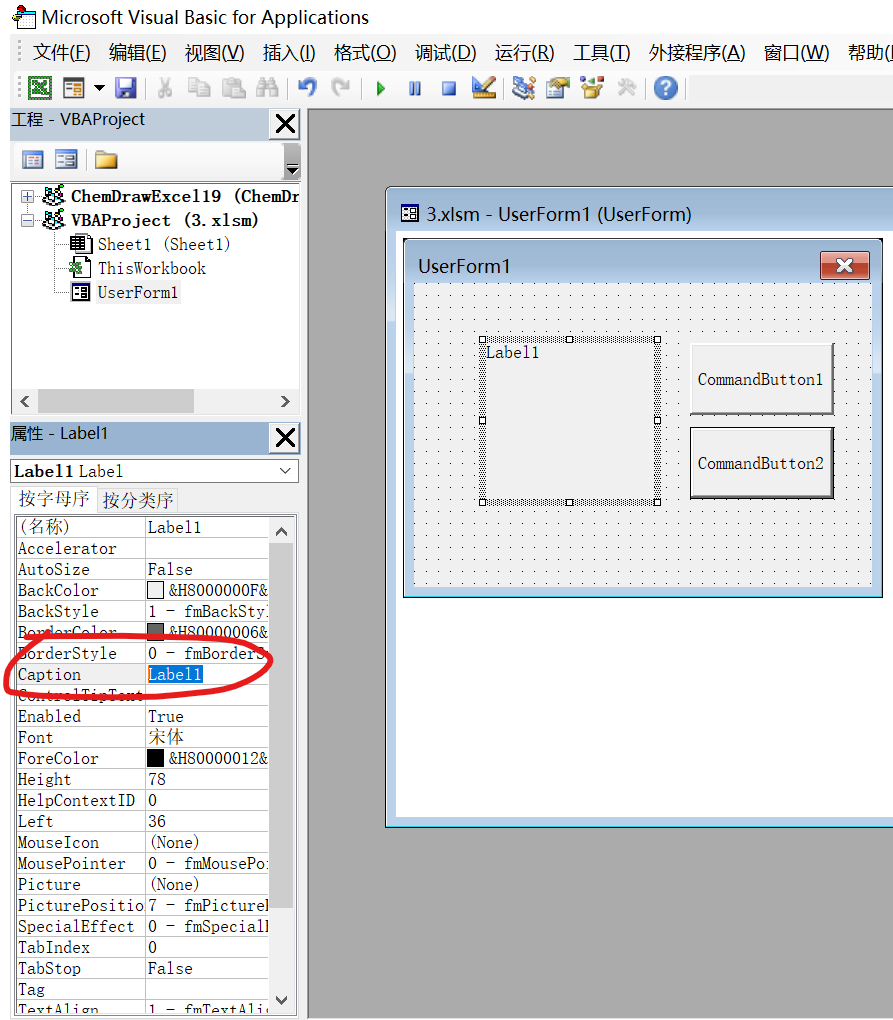
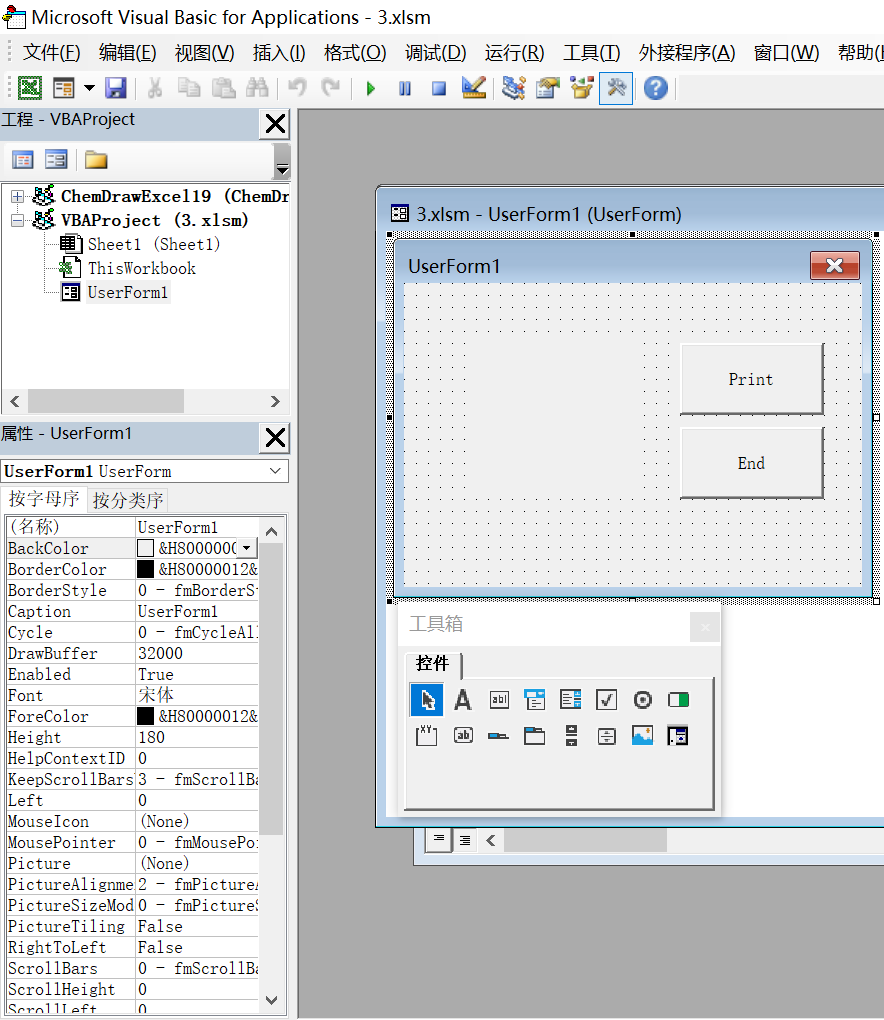
与之前在表格界面中稍有不同，在VBA编辑器中，左键单击控件并不会运行脚本

因此左键单击选中控件后，就可以直接拖动和缩放，单击文字区域还可以修改文本

修改文本也可以在左侧属性列表里找到“Caption”属性，进行手动更改，如下图左

双击控件将进入代码窗口，并自动填充默认事件。若误触发了关闭即可

任务：请修改界面，使效果如下图右所示

下面双击左侧“ThisWorkbook”，进入代码窗口，下拉代码窗口顶部左侧的菜单，选择“Workbook”会自动生成两行代码，在它们之间输入以下代码

UserForm1.Show

关闭代码窗口或双击“UserForm1”回到窗口，双击命令按钮1，输入下面的代码

Label1.Caption = "Hello World!"

关闭代码窗口或双击“UserForm1”回到窗口，双击命令按钮2，输入下面的代码

End

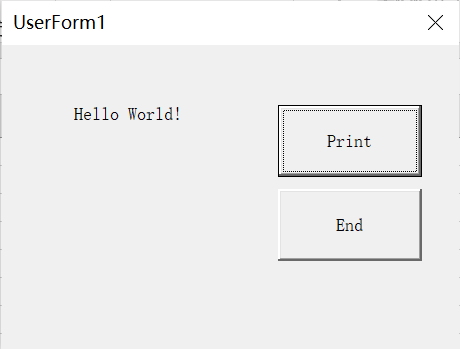
下面运行一下这个脚本

在VBA编辑器中点击运行，再操作控件执行脚本

保存并关闭文件，重新打开，再运行看看

这和第一个例子有什么不同？是如何实现的？

运行效果如下

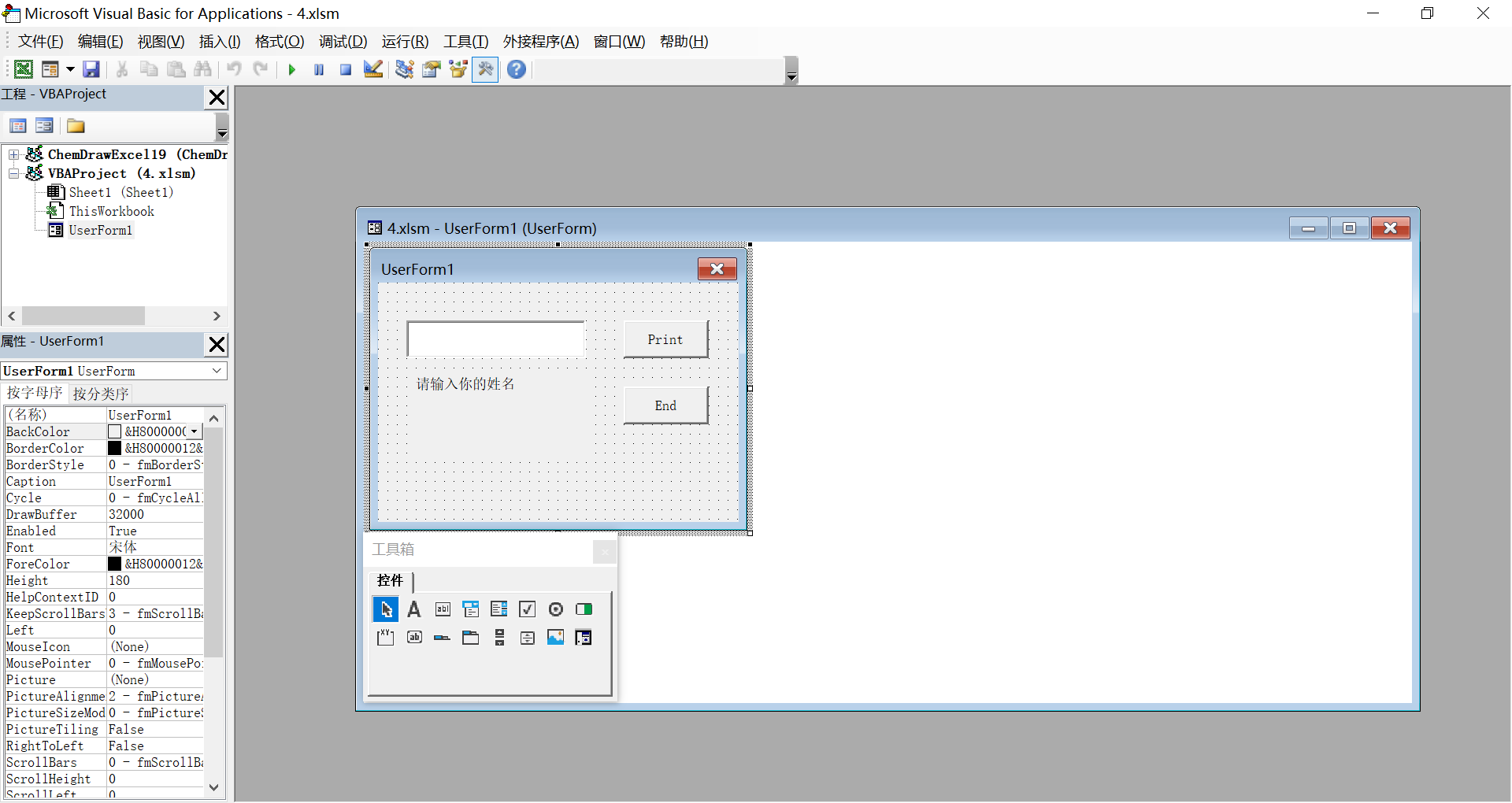


* 1. **读取输入！**

[示例文件：4.xlsm](4.xlsm)

新建一个.xlsm文件，进入VBA编辑器，绘制如下窗体

所用到的新控件是文本框（TextBox），在第一行第三个



“ThisWorkbook”代码

UserForm1.Show

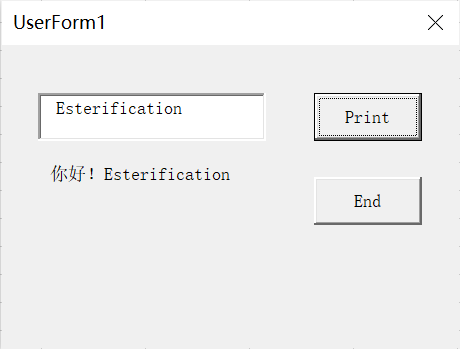
命令按钮1

Label1.Caption = "你好！" + TextBox1.Text

命令按钮2代码

End

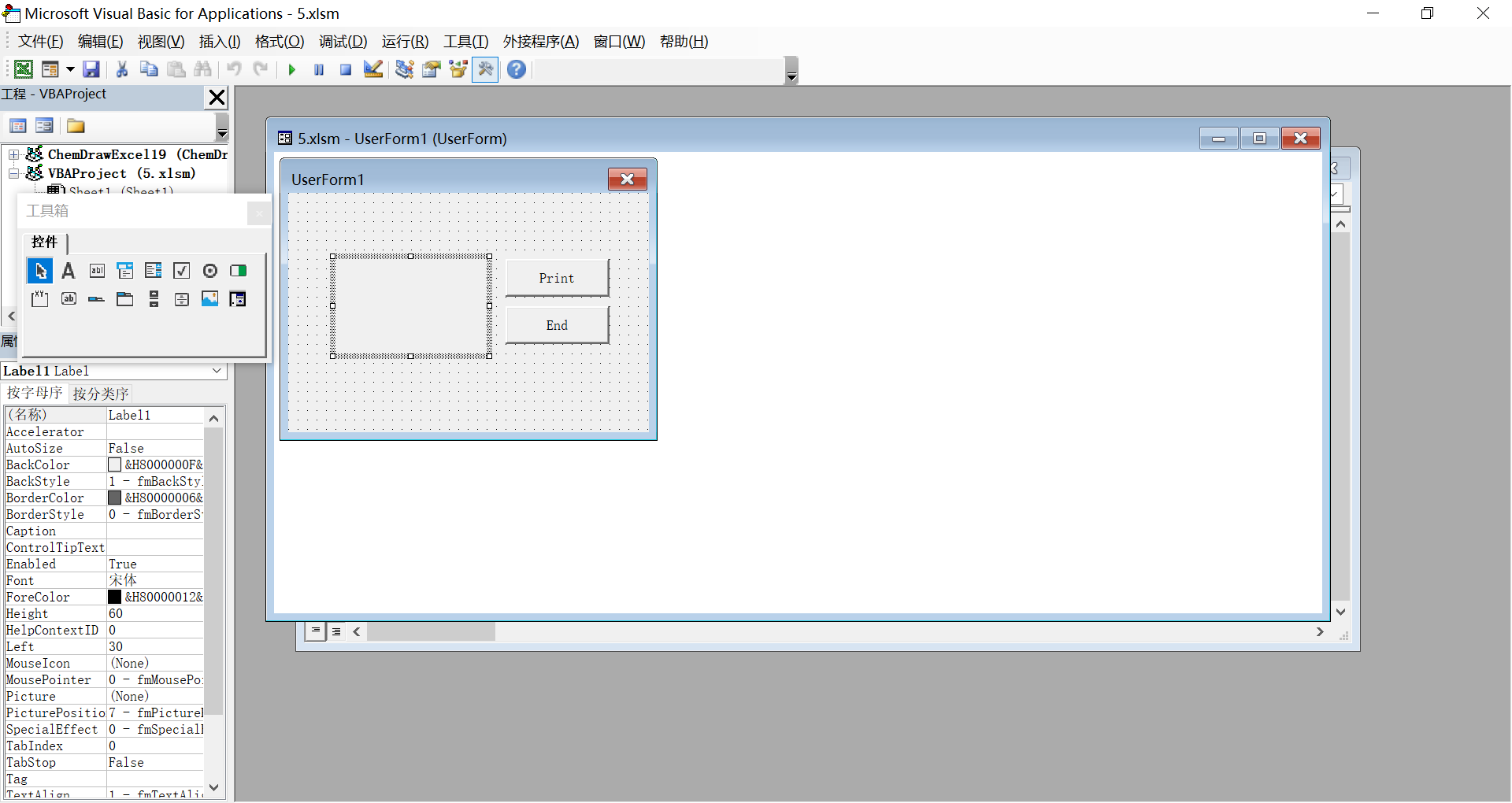
运行，在文本框中输入一些内容，再按Print键，结果如下



* 1. **继续读取！**

[示例文件：5.xlsm](5.xlsm)

新建一个.xlsm文件，先在A1单元格输入“Hello World！”，再绘制如下窗体

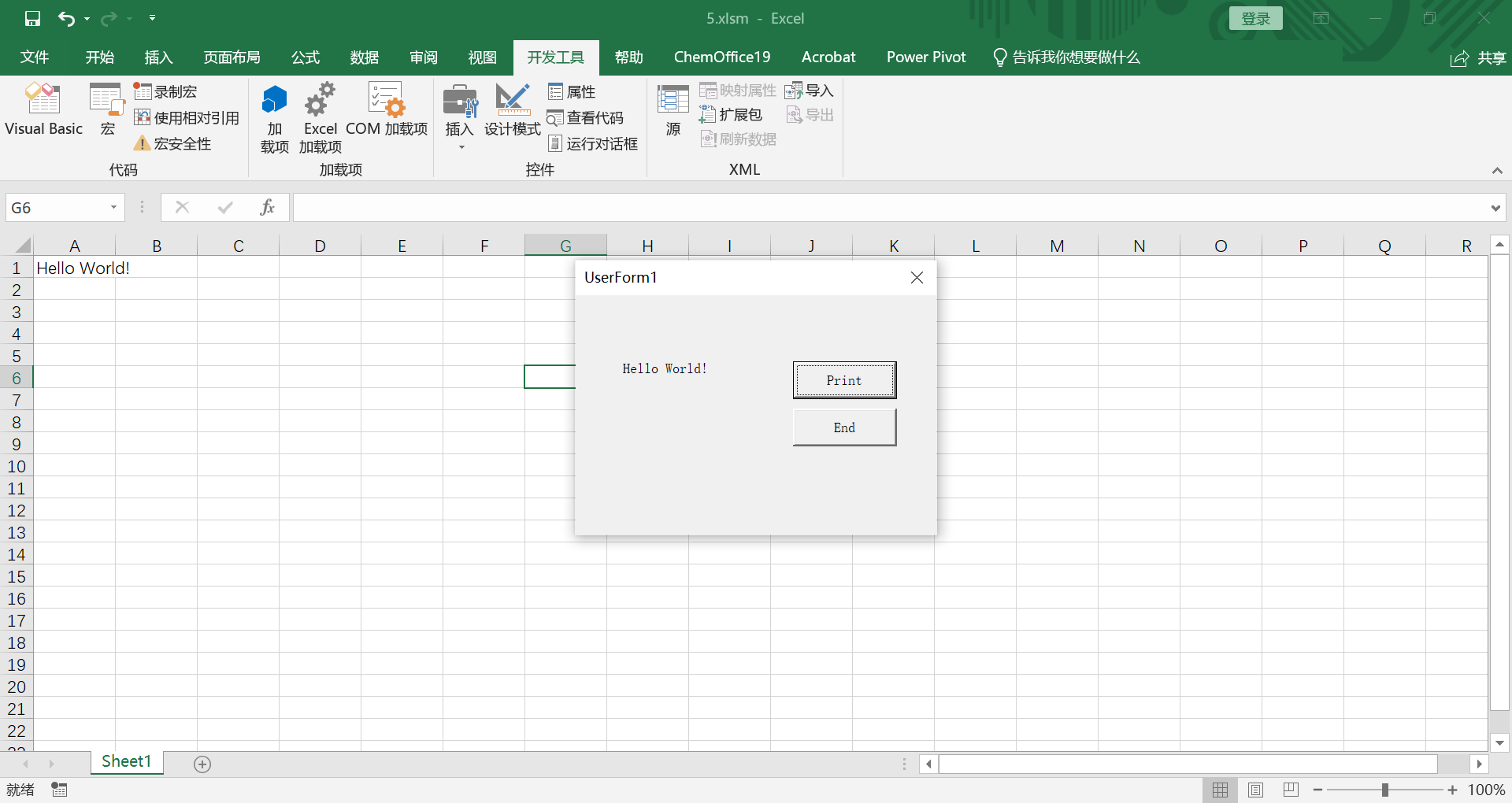


“ThisWorkbook”和“End”代码同上

“Print”代码如下

Label1.Caption = Range("A1").Value

运行结果如下



1. **VBE简介**
   1. **什么是VBE？**

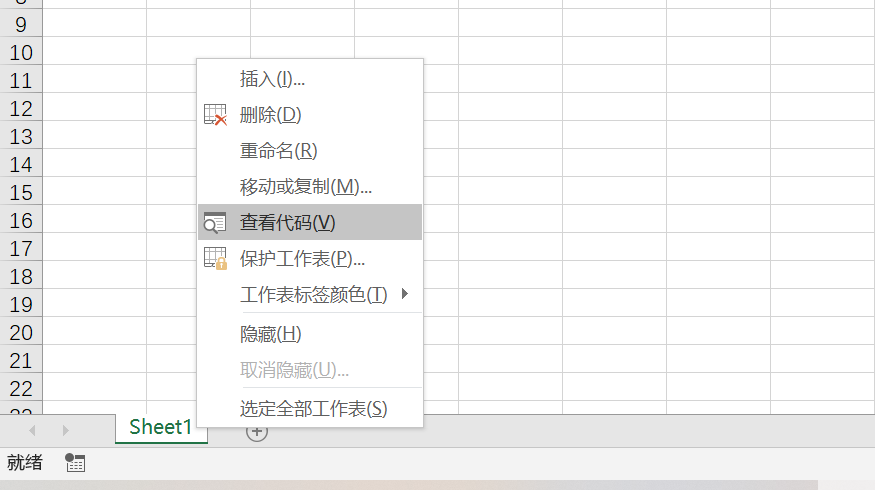
上文中，工作界面被称为“VBA编辑器”。实际上，这个界面是一个集成开发环境（IDE），有自己的简称“VBE”。尽管不像别的IDE（例如Visual Studio）那么好用，但使用VBE可以直接调试Excel，操纵Excel提供的对象也非常方便。

集成开发环境（IDE）是用于提供程序开发环境的应用程序，一般包括代码编辑器、编译器、调试器和图形用户界面等工具，集成了代码编写功能、分析功能、编译功能、调试功能等，可以实现一体化的软件开发。

题外话，除了IDE，另外一种用于编程的工具是文本编辑器（不同于文档编辑器/文字处理器，如Word），如Windows自带的“记事本”、Visual Studio Code等。一些文本编辑器除了提供最基础的文本编辑功能以外，还针对编写程序设置了代码高亮、自动缩进等。更专业的文本编辑器还有Vim、Unix和Emacs，这里不再展开

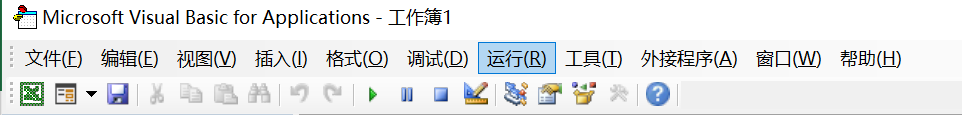
* 1. **如何进入VBE？**

1. 在“开发工具”选项卡中点击“Visual Studio”按钮
2. 使用Alt+ F11键盘命令（在VBE中使用Alt+ F11将回到Excel界面）
3. 右键单击工作表标签，选中“查看代码”



* 1. **主界面概述**

从上到下，分别有标题栏、菜单栏、工具栏



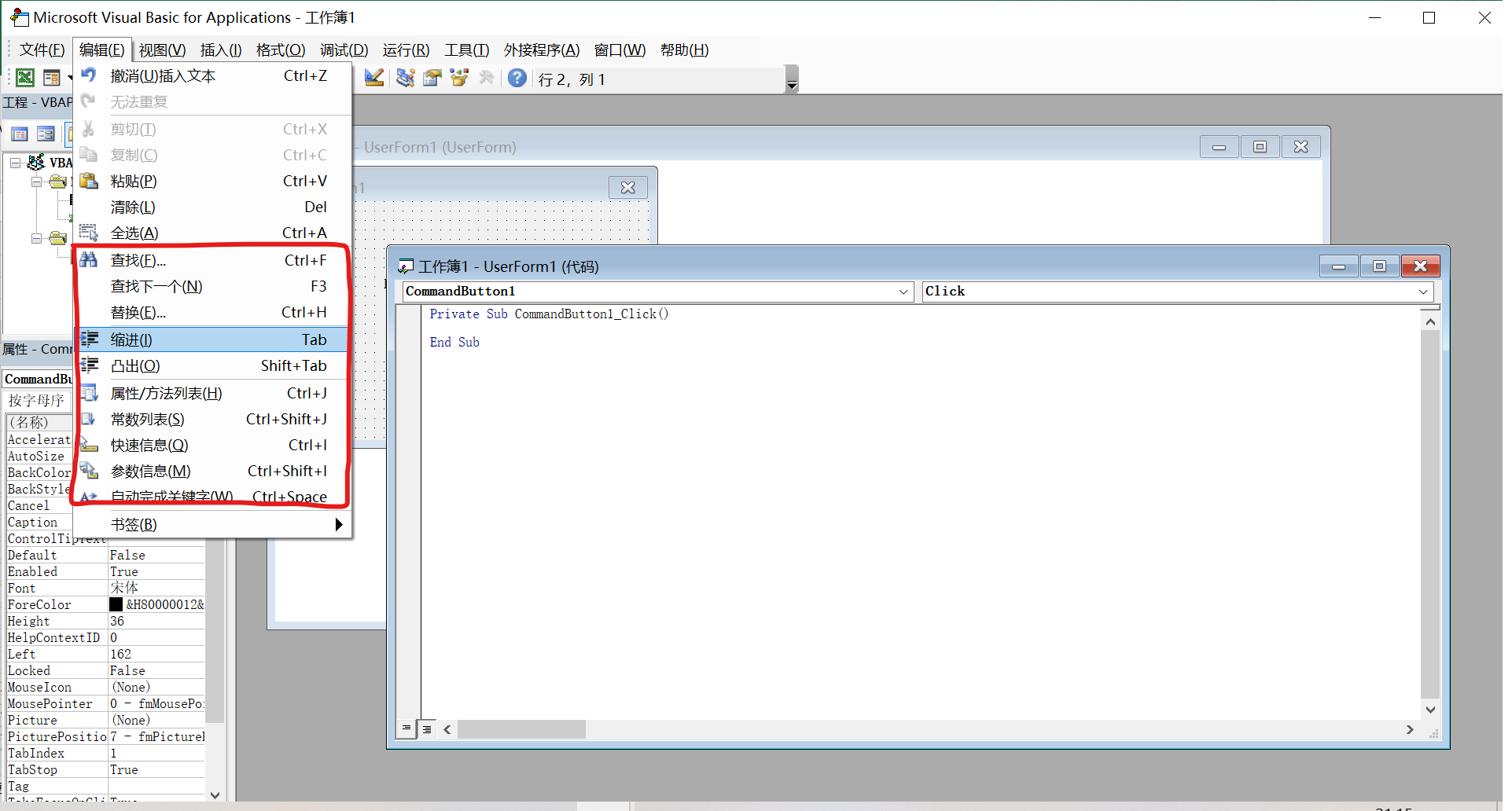
左侧有刚刚已经看到过的工程资源管理器和属性窗口

中间是用户界面、代码窗口、工具箱等，拖动或缩放用户界面将修改运行时的效果

* 1. **菜单栏**

接下来我们将逐一查看菜单栏的选项卡

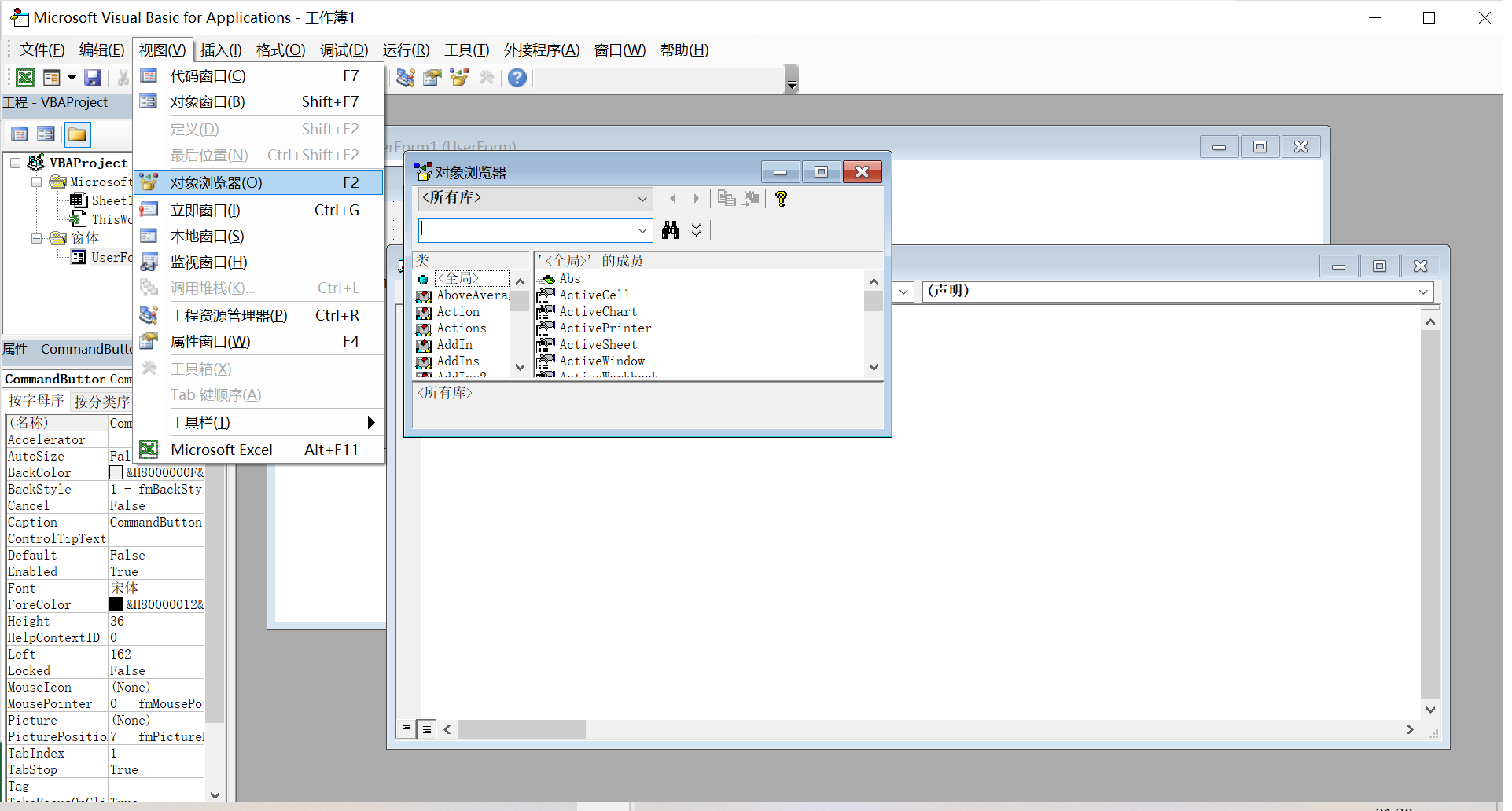
“文件”和“编辑”没啥好说的，基本操作，“编辑”中有一部分选项只在代码窗口下可用



“视图”选项卡下，注意“对象浏览器”。下一个章节会有相关内容，按F2键调出

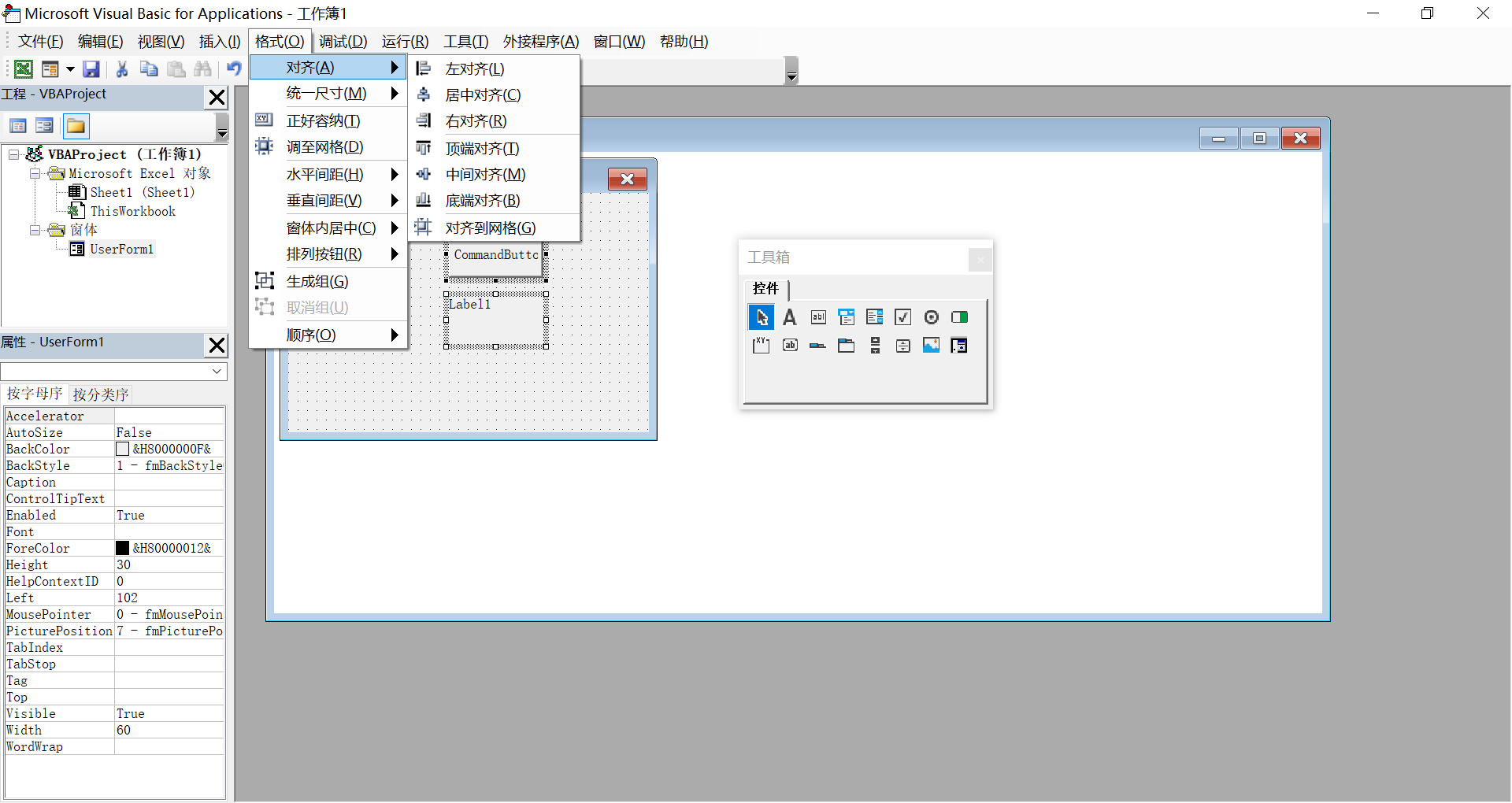
另外还可以唤出“工程资源管理器”和“属性窗口”

“工具箱”和“Tab键顺序”在“用户窗体”下可用



“插入”一栏，在初级阶段应该只需要用到“用户窗体”，不再展开

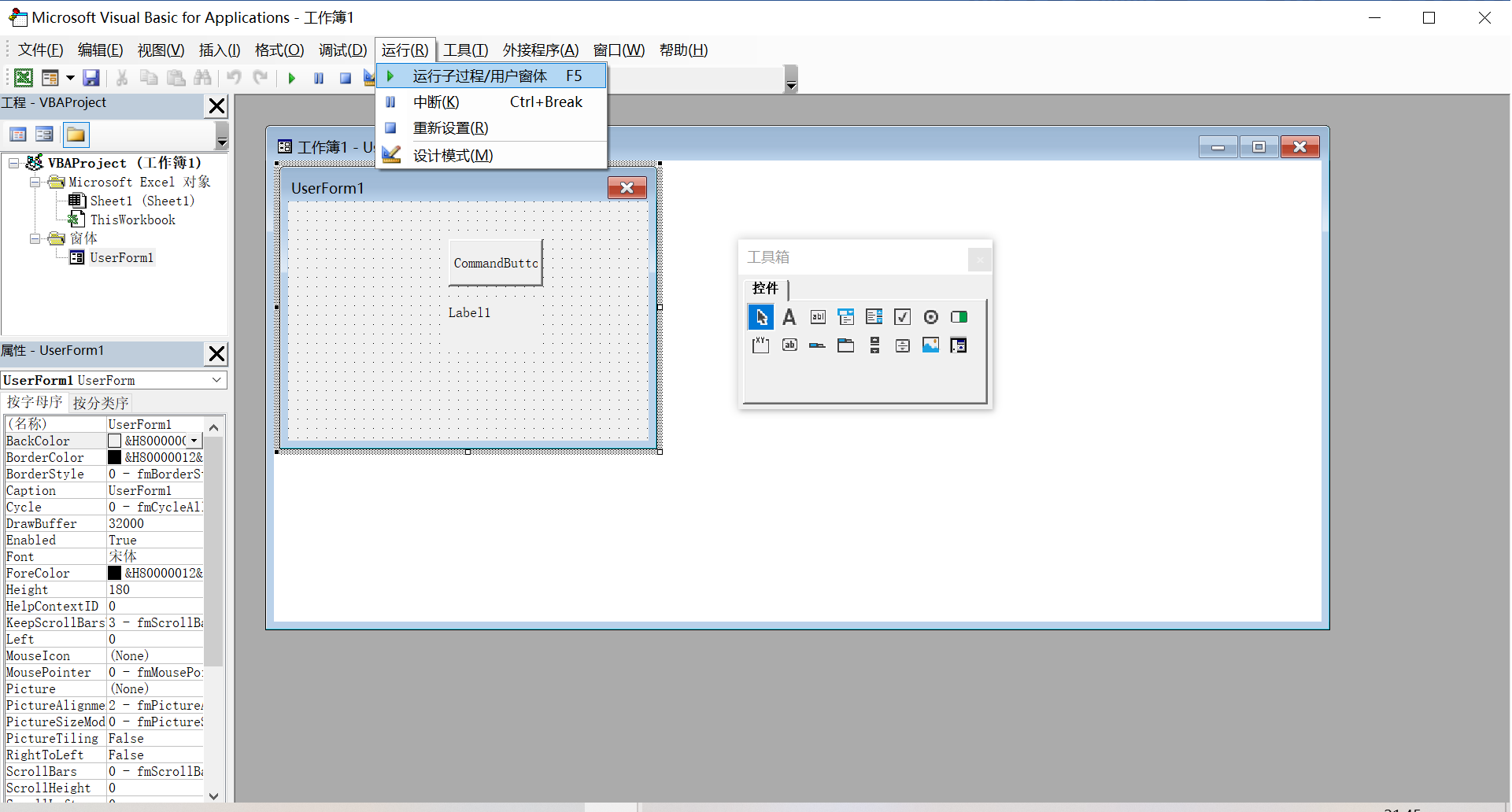
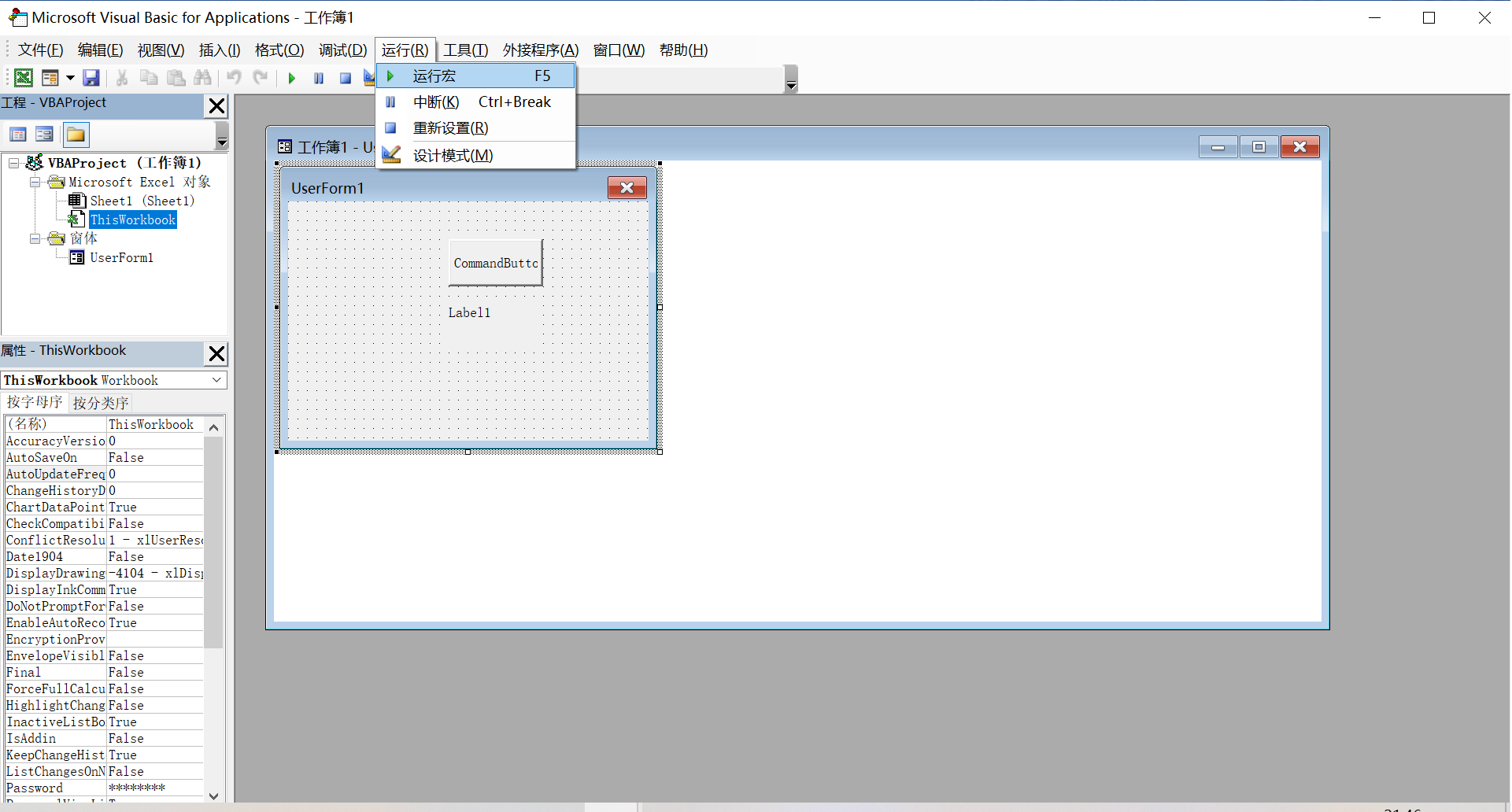
“格式”选项卡下的功能主要服务于窗体的绘制。注意！有的选项需要选择两项或以上的控件才能执行，选中控件的方法与Windows文件管理器选择图标类似。即，按住“Shift”或“Ctrl”并单击需要选中的控件，或者拖动光标框住需要选中的控件。具体功能就留给各位自行尝试叭~



“运行”选项卡的功能与工具栏的小图标功能重叠，基本用不上，但请注意以下区别

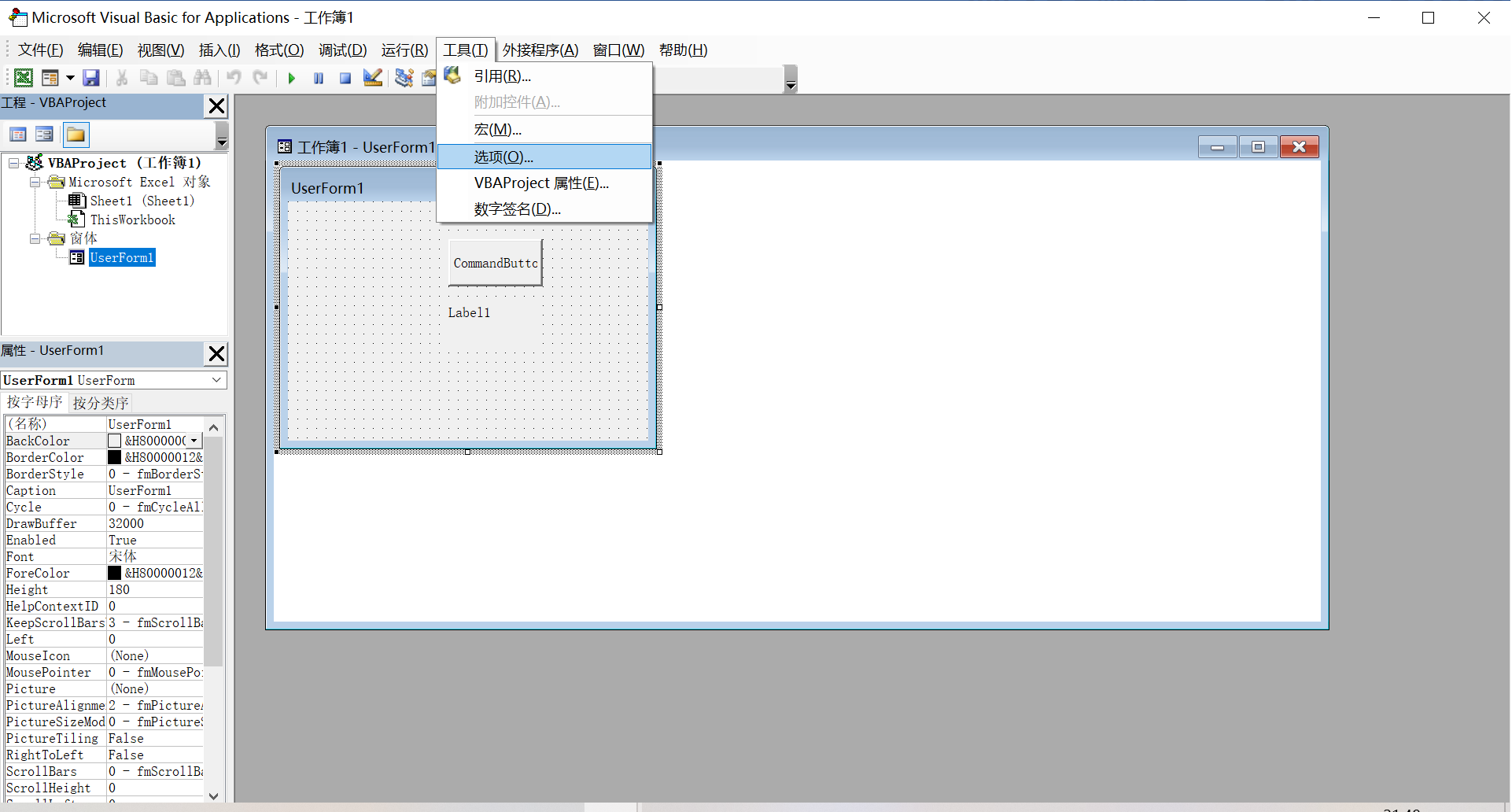
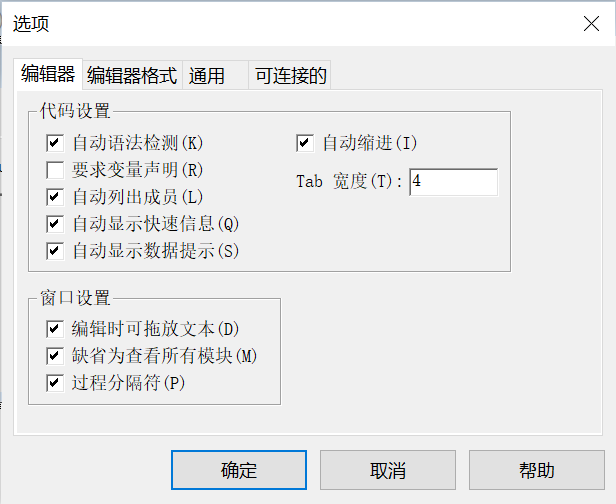
光标点击过用户窗体的情况下，运行的是“子过程/用户窗体”

光标点在左侧用户资源管理器的任意选项上时，运行的是“宏”，一般请避免

“工具”选项卡下，进入“引用”和“选项”两项，可以进行一些高级个人偏好设置，我们用默认的就可以了。如果想养成良好的编程习惯，可以勾选“要求变量声明”选项

点选“宏”之后，可以进入宏的编辑器，之后也会讲到相关内容，现在先跳过

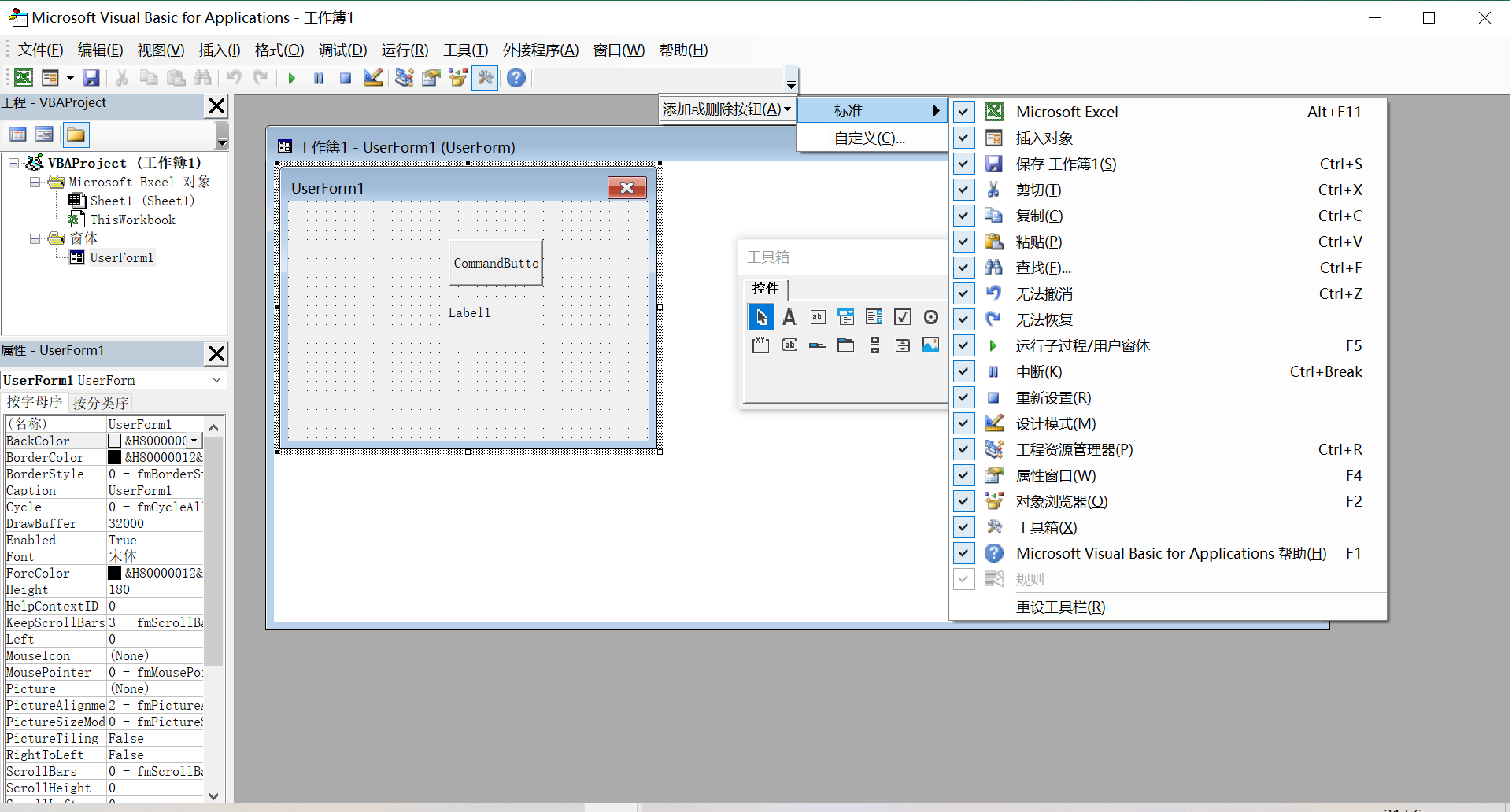
 

后面几个选项卡和“调试”选项卡就不讲了Orz

* 1. **工具栏**

将鼠标放置在对应按钮的上方就可以看到功能的名字，都是我们刚刚看到过的，有不少都是常用重要功能，自行了解即可。

点击工具栏最右端的小箭头可以调出下拉菜单，可以修改工具栏上的按钮

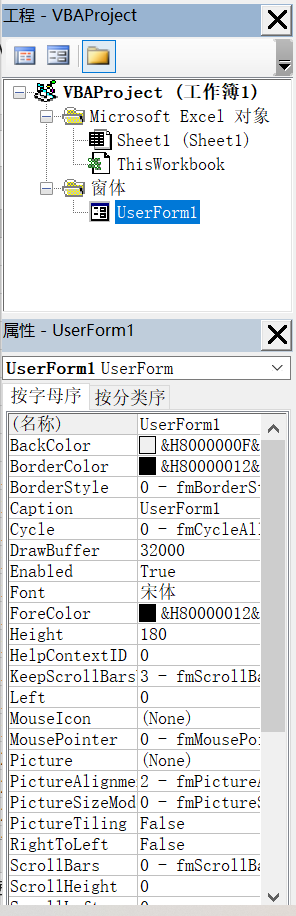
****

小提醒，对于很常用很常用的操作，如复制、粘贴、剪切、撤销、全选、保存等，还是尽量记住键盘命令，能有效提高工作效率，而且这些指令几乎是全平台通用的!

* 1. **工程资源管理器**

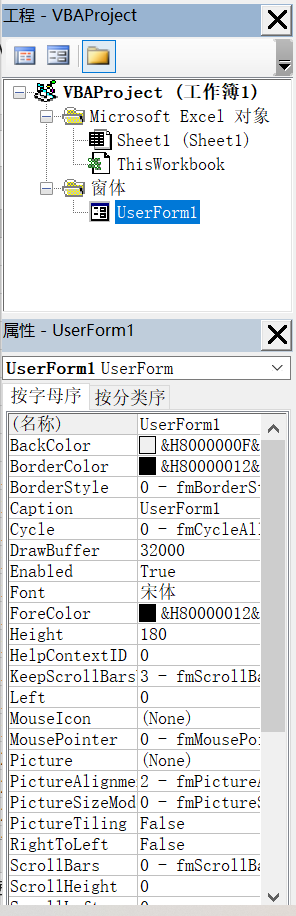
工程资源管理器可以理解为一个导航窗格，单击在对象之间切换，双击进入代码窗体；上方的三个小按钮分别是查看对象、查看代码和切换文件夹，点点试试！

在“视图”选项卡点选“工程资源管理器”可以调出，也可以点击工具栏小图标唤出或使用键盘命令Ctrl+ R唤出。另外窗体大小、位置都可以用拖动修改。



* 1. **属性窗口**

属性窗口中列出了目前光标所指对象的属性，属性的具体内容稍后再讲。在这个窗口里，可以方便地编辑某些属性，如名称、字体（Font）、说明文字（Caption）等。点击属性名右侧的属性内容，即进入可编辑模式，键入或选择以完成编辑。



1. **一些基本概念**
   1. **对象（Object）**

含义与字面相同。即，我们要操纵的“物件”，如CommandButton（命令按钮）等控件、UserForm（用户窗体）、Application（整个Excel应用）、WorkBook（一个工作簿）、WorkSheet（一个工作表）、Range（选区）、Cells（单元格）、Selection（选中的区域）、Window（窗口对象）等等

对象的调用格式类似地址或系统路径的书写，需要从大到小按序书写，如

Application.Wookbooks(“FileName”).WorkSheets(“SheetName”).Range(“A2:D3”)

有时可以简化，如果当前工作簿、工作表均为激活状态，直接写Range(“A2:D3”) 即可，甚至可以简写为[A1:D10]

* 1. **属性**

对象都具有属性,在属性窗口中可以查看和修改，也可以用代码改变属性的值。请注意！不同的属性，其值的数据类型（字符串型、数值型、布尔型等）可能是不同的。

对象的Name（名称）属性是该对象的唯一标志，是调用对象及其他属性所必须的

常用的控件属性还有Caption、Font、Enable、Visible等，还有TextBox的Text属性

常用的Excel属性有Selection的Row和Column 等

属性的调用格式是[对象名].[属性名]，如 Label1.Caption、Text1.Text 等

* 1. **方法**

方法是指对对象进行的操作，如：

Range("A1").Select UserForm1.Show

第一个例子中，Range("A1") 是对象，Select 是对其执行的“选中“操作，Select 就是一个方法；第二个例子中，UserForm1 是对象，Show是一个方法

有的方法需要提供一些额外信息，称为参数，如

ActiveDocument.SaveAs ("New Document Name.docx")

其中，SaveAs 是一个方法，后面括号中的内容就是该方法的参数。

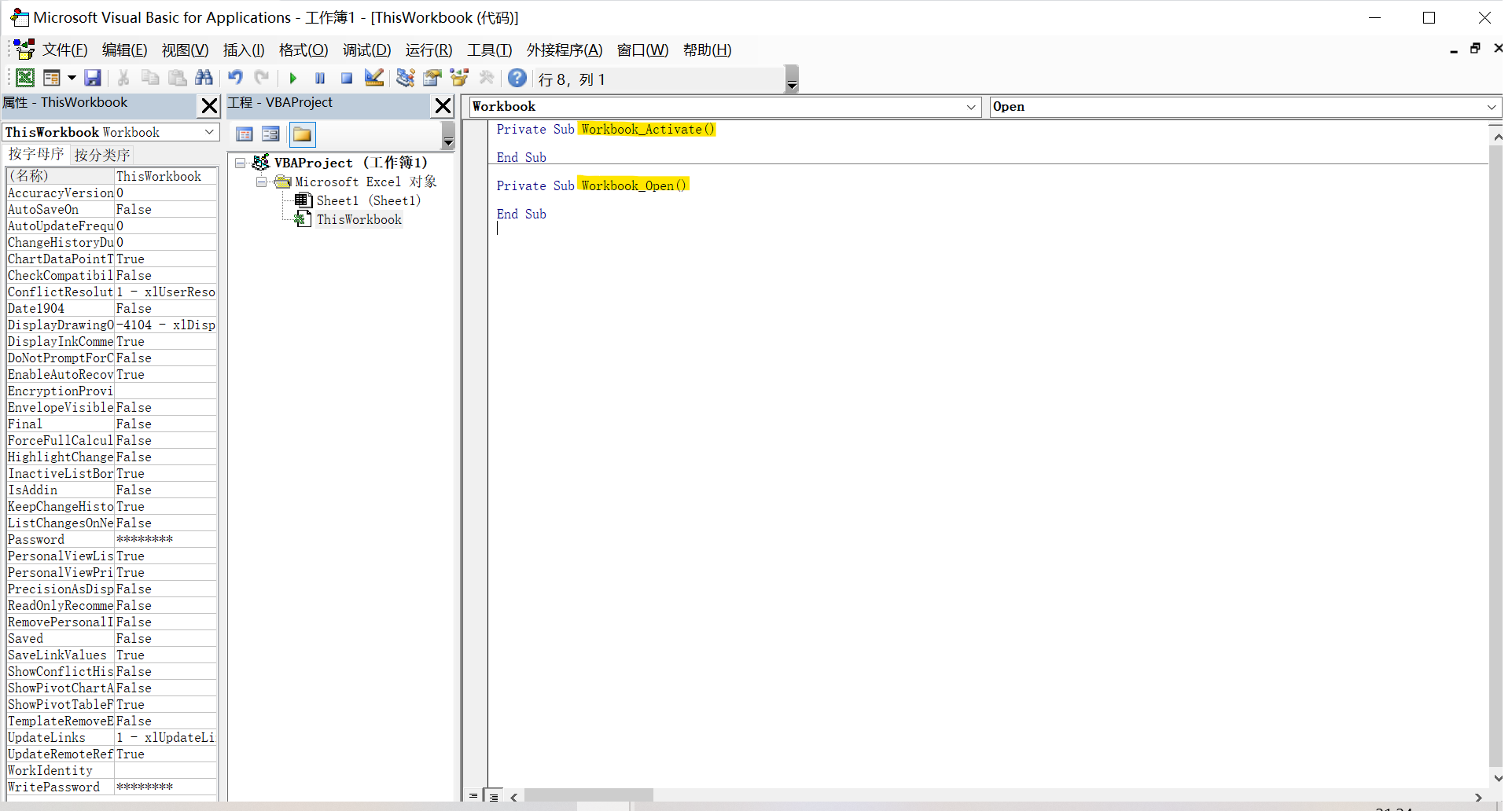
方法的调用格式类似于属性，是[对象名].[方法](参数)

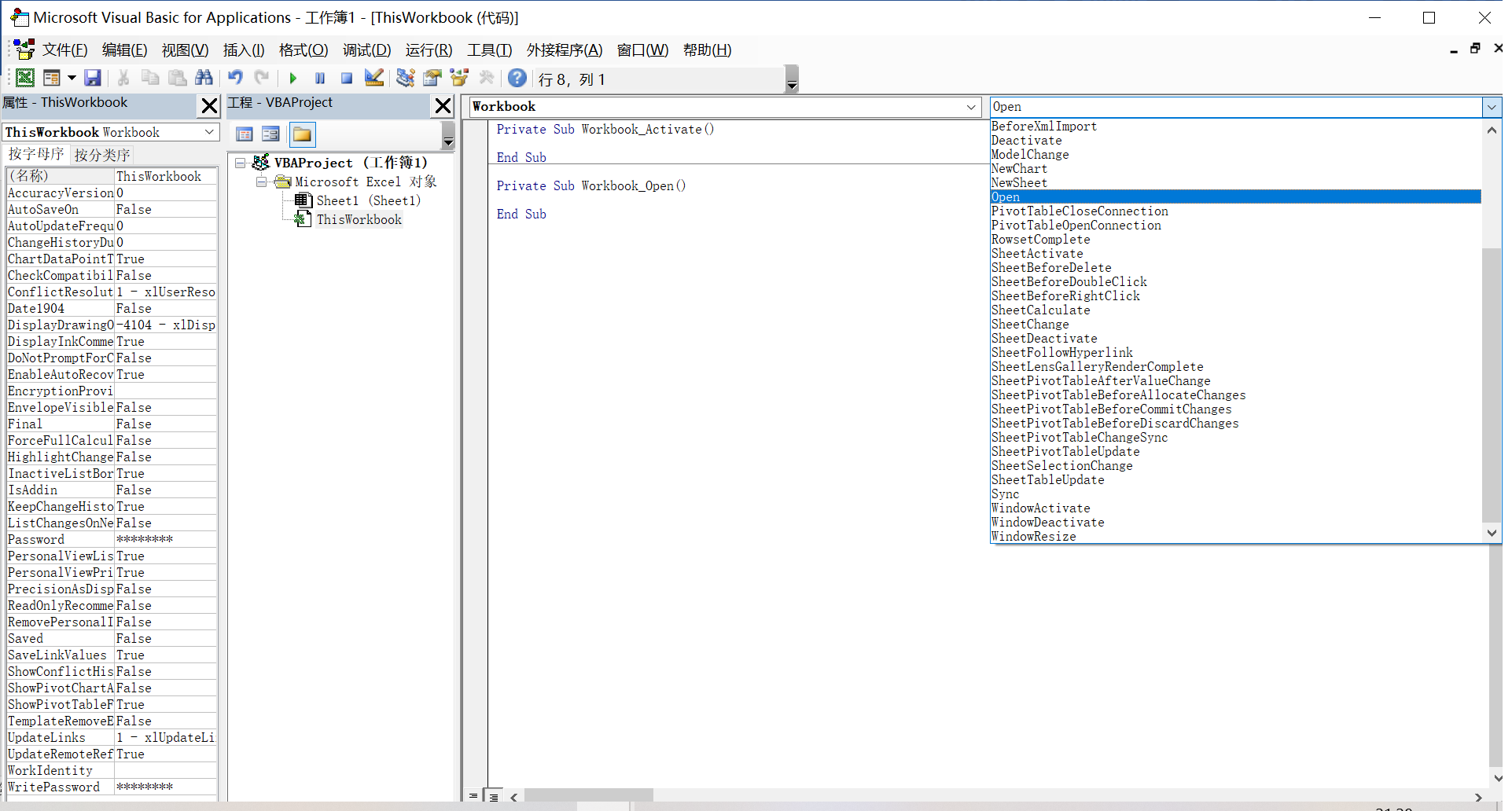
* 1. **事件**

字面意思，由系统或用户触发，一般用来在某条件下执行预先写好的命令

如前面看到过的，WorkBook\_Open 事件，就是指在工作簿打开时执行其中的代码

调用方式是下拉菜单。选中对象进入其代码窗口后，左侧选择对象，右侧选择事件，效果如下



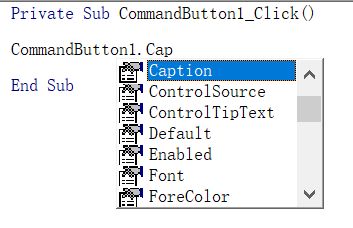


“Sub”是一个“过程”，在下面的章节会提到，这里强行接受即可

* 1. **小结**

对象、属性、方法、事件过于繁杂，这里提到的只有很少的一部分。除此之外，我们还略过了类、模块等复杂的概念。此节的目的在于建立对代码的感性认识，并将代码与控件、窗体、表格等对象之间联系起来。第一次阅读完本节，能够用这些概念解释第二节的示例代码即可。VBA中的对象、属性、方法、事件不需要也不可能全部记住，需要的时候知道在哪查就可以了，也就是说，可以现学。

VBE提供了“自动列出成员”功能，可以帮助快速书写代码，各位在使用Visual Studio的过程中应该也体验过了。该功能默认开启，请在“工具”-“选项”中检查。



1. **VBA语法基础**

因为VBA是一种面向对象的编程语言，学习过程中可以和同为面向对象编程语言的C++联系起来。大部分概念我们在计算机基础课上已经学过，因此，本节将不再对概念进行展开解释，主要列出VBA实现。

* 1. **常量和变量**

常量声明语句

Const 常量名 = 数值或表达式

Const Pi = 3.1415926

变量声明语句

Dim 变量名 [As 数据类型]（方括号代表可缺省，下同）

Dim i As Integer，x As Single

常量和变量的命名原则：

1）开头必须为英文或中文字符

2）名称长度不超过255字符（一个汉字占两个字符）

3）不能与VBA保留字冲突

4）量之间名字不冲突

在Sub 和End Sub 之间声明的变量仅作用于这个语句块

在Sub 外声明的变量作用于整个程序

* 1. **数据类型**
     1. **概述**

字符串（String）、整形（Integer）、长整型（Long）、单精度浮点型（Single）、双精度浮点型（Double）、布尔型（Boolean）、日期型（Date）

货币型（Currency）:储存定点数，小数点左侧15位、右侧4位

字节型（Byte）:字面意思，长度为一字节，储存0~255的整数

* + 1. **强制转换**

VBA提供了很多类型转换函数，摘录几个

CStr()，将数值型转化成字符串型

CInt()，对分数四舍五入，转化成整数

CBool()，转换为布尔型，0对应False，其他对应True；True转出时对应-1

还有一些有意思的转换函数

Asc(String)，给出字符串首字母对应的ASCII码

Chr(Charcode)，给出ASCII码对应的字符

Int(Number)，向下取整

Fix(Number)，向绝对值小的方向取整

Str(Number)，转化为字符串，正数前面加一个空格，负数前面加负号

Val(String)，提取字符串中的所有数字，并取合适的数据类型

下面给出一个使用示例

X = CInt(TextBox1.Text)

* + 1. **获知数据类型**

使用VarType函数，返回值与数据类型的对应关系请上网查阅

Label1.Caption = VarType([A1])

* 1. **运算符号、函数**
     1. **数值运算**

+；-；\*；/；\（整除）；%（求余）；^（幂运算）

Abs()；Int()；Sqr()；Exp()；Log()；Sin()；Cos()；Tan()

=（算术运算符和赋值运算符是同一个！注意！）

<>；<；>；<=；>=；

* + 1. **逻辑运算**

Not；And；OR

* + 1. **字符串操作**

字符串除了声明以外，在代码书写中用引号表识。引号内的引号是正常字符

“+”号或“&”号实现字符串的连接

去除空白：Trim(String)、LTrim(String)、RTrim(String)；L、R是左右的前缀

获取字符串长度：Len(String)

截取：Left(String,x)、Right(String,x)、Mid(String,Start,x)；x代表截取位数

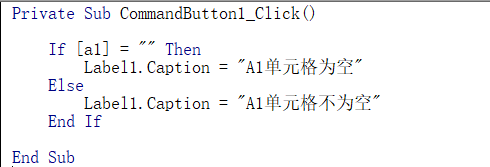
* 1. **逻辑结构的VBA实现**
     1. **基础语句简介**

赋值语句：咱刚刚一直在使着呢

注释语句：VBA的注释号用英文引号

结束语句：End

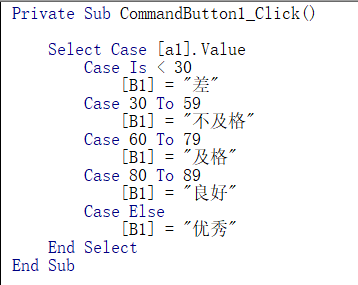
* + 1. **分支结构的VBA实现**



这个结构被称作“If-Then-Else”结构，还可以有“If-Then”结构、“If-Then-Elseif-（好多好多个Elseif，Elseif 后面要跟逻辑表达式和Then）-Else”结构

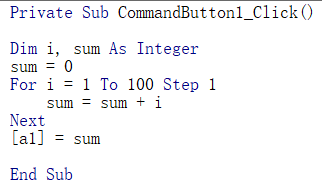
注意不要忘了写End If ！

Select Case语句，注意End Select 和Case 条件的两种表达



* + 1. **循环结构的VBA实现**

For-Next语句



其他语句还有

For-Each 语句，需要用到数组的知识，不确定循环次数的时候比较好用

Do-While语句

Do-Until语句

* + 1. **转向语句Goto**

Goto 一般与If 搭配使用，If(逻辑表达式) Then Goto (标记)，不写End If

标记位置的格式是(标记：)

一般不建议使用

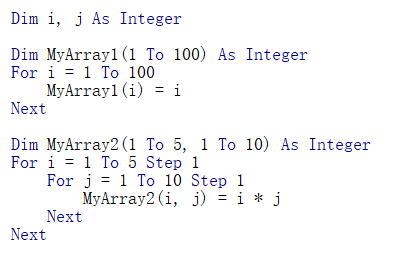
* 1. **数据结构的VBA实现**

一维和多维数组的声明格式、调用方法、赋值方法

还可以用Public命令声明全局变量和数组

For 结构中，Step 缺省时其值默认为1，方便计数

运行过程中可以用ReDim命令重设数组大小



* 1. **用户交互**

在之前的很多案例里，我们通过读取单元格内容、文本框控件来实现内容输入，利用写入单元格和修改标签的Caption属性输出内容，是否还有别的方法呢？

InputBox与MsgBox

[示例文件：6.xlsm](6.xlsm)



另外还可以调用Application对象的Dialog属性。具体值对应的对话框查F1或F2

Application.Dialog(103).Show

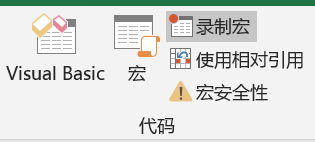
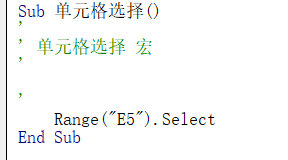
1. **Excel宏**
   1. **概述**

我们为何使用VBA？实际上，几乎所有Excel操作都有其对应的代码表示。当我们面对大量重复工作时，使用VBA能大大加快工作效率。另外，VBA能提供灵活的和用户进行交互的方式。那么如何找到我们想要的操作的代码？

宏（Macro），是一种批处理规则。进一步了解此概念请百度。

* 1. **录制宏**

在“开发工具”栏，可以找到“录制宏”按钮，点击之后会弹出弹窗，可以编辑宏的名称、作用范围等。设置好后，点击“确定”，开始录制。直到点击“结束录制”为止，每一步操作都将被记录，并生成代码，点击“宏”按钮，跳出弹窗，点击“编辑”以查看录制好的代码。

** **

做一个小解释，这里我们又出现了Sub 和End Sub，搜索“Sub 过程”，了解更多~

在Excel界面，点击“宏”按钮调用录制好的宏；在代码层次，可以用Call函数调用宏

1. **进一步学习**
   1. **阅读官方参考文档**

虽然感觉有点不说人话，但官方文档提供了充足的例子，是个很好的学习资源

总览

<https://docs.microsoft.com/zh-cn/office/vba/api/overview/>

入门必读：

https://docs.microsoft.com/zh-cn/office/vba/library-reference/concepts/getting-started-with-vba-in-office

后端请读：

<https://docs.microsoft.com/zh-cn/office/vba/language/reference/user-interface-help/visual-basic-conceptual-topics>

前端请读：

https://docs.microsoft.com/zh-cn/office/vba/language/reference/user-interface-help/concepts-microsoft-forms

* 1. **录制宏以获得操作的代码实现**

将想要代码化的功能进行录制，获得并修改对应操作的代码。

为了弄懂代码如何工作，可以尝试修改其中的一些内容，再看看运行结果的变化

最终将这段代码与逻辑结构结合起来，合理安排进程序之中，实现目标功能

* 1. **分解问题**

将想要解决的问题抽象化并分成更小的逻辑模块，然后百度

如，“将两个特定行之间的第一列的单元格内容填充为1”，可以分解为“查找并获取特定行的行号”和“批量修改单元格内容”两个更小的问题。小的问题容易查找到解决办法。

1. **其他Excel使用技巧**

Ctrl+G ，对录制宏很有帮助，选取含有“某一类”内容的单元格，可以自己试试

Ctrl+E ，好像是个智能操作？搜搜看看？再动手做做